NUS
National University of Singapore

# UNDERGRADUATE RESEARCH OPPORTUNITIES PROGRAMME (UROP)

Using Deep Learning For Medication Recognition

YiFei Tang

Department of Innovation and Design, Faculty of Engineering
National University of Singapore

## Abstract

In 1999, the Institute of Medicine (IOM) found that up to 98000 hospital deaths a year were caused by medication error. [1] It is therefore of utmost importance that a system is implemented to verify the accuracy of prescriptions.

This paper introduces a system called PillSafe, used in recognizing capsules of medication using software in a realistic hospital setting, accounting for various modes of lighting. The colour properties of the pill is extracted, then classified using unsupervised machine learning. A user can input a pill in both loose form and blister packaging into a database of pills. IN order to classify the pill, the software will first use machine learning to extract HSV features. It will compare the features to the database of pills and select the correct pill. Users can interact with the application using a very convenient Graphical User Interface (GUI).

## Preface and Acknowledgments

The author would like to express gratitude towards Dr. Tang Kok Zuea who provided essential feedback regarding the scope and range of the project. Resources provided were highly useful and Dr. Tang’s key insight helped guide the course of the project.

Thanks towards fellow research assistant Nayan Arora are also due. He provided a good deal of support throughout the project as the author tackled difficult topics of machine learning.

## Table of Contents:

# 1. Introduction

A 2012 report by the University of Nottingham found that 4.9% of all prescriptions items were errors. [1] While most errors have little to no harm, some can be fatal. As stated in the 1999 paper “To Err is Human”, up to 98000 deaths a year can be attributed to prescription error. [2]

The goal of the research project is to create a convenient and efficient pill classification system. Some of the notable questions that came up during the course of the project was regarding the nature of the pill. The size of the pill is generally very small and occupies little space. Any good classification algorithm must be able to isolate the pill from a larger image. Another question is the lighting of the test conditions. Due to different lighting conditions and cameras, the algorithm must be able to function in many environments.

# 2. Literature Review

In order to determine the scope of this project, research and industry detection systems were reviewed during the course of the project.

The predecessor of this project, MedR2, was the thesis project of National University of Singapore student Ong Bi Qin Kimberly. [3] MedR2 was developed as an extension of a previous student project, MedSafeRegistration, a system used to detect pills by their colour and shape. MedR2 was developed to address MedSafeRegistration's problem of not having a database nor live pill recognition (a pill had to be uploaded in order to be classified). MedR2 also had the advantage of being able to recognize pills in their blister packaging. As PillSafe uses many similar image processing techniques, the techniques described by MedR2 and MedSafeRegistration were consulted frequently during the early stages of the project.

Commercial solutions like MedSnap ID was also reviewed prior to the development of PillSafe [4] MedSnap ID is a Mobile app used to classify pills by their colour, size, shape and imprints. Similar to PillSafe,  computer vision is used to process the image and retrieve data from a database regarding the pills. The advantage of MedSnap ID is that it is user oriented so the user can classify their personal pills. The problem of this system is that it does not recognize pills inside their blister packaging.

Another commercial solution is SwissLock [5], an automated packaging system designed to limit human touch. Due to the fact that an average pill is handled 10 times before reaching the patient, reducing human interaction with the pill allows for a much smaller chance of failure. The Pharmacist manually loads the pill into the system and PillPick organizes pills and removes incorrect medications. This system is also limited however, with the initial step of requiring the Pharmacist to manually load the machine.

# 3. Design Of Study

The aim of this study was to research how deep learning can be employed to classify medication in pill forms. In order to determine the scope of the project, constraints and objectives were set. Constraints are inviolable requirements that the project must follow while objectives were nice to have. In addition the writer assessed the limitations of the project due to the fact that the project must be completed in a limited time frame (2 months).

**A. Constraints**

1. The accuracy of classification must be at least as effective as the MedR2 (90% for one test set)
2. The research must investigate the efficacy of convolutional neural networks
3. The detection system must be able to recognize pills in both loose forms and blister packaging

**B. Objectives**

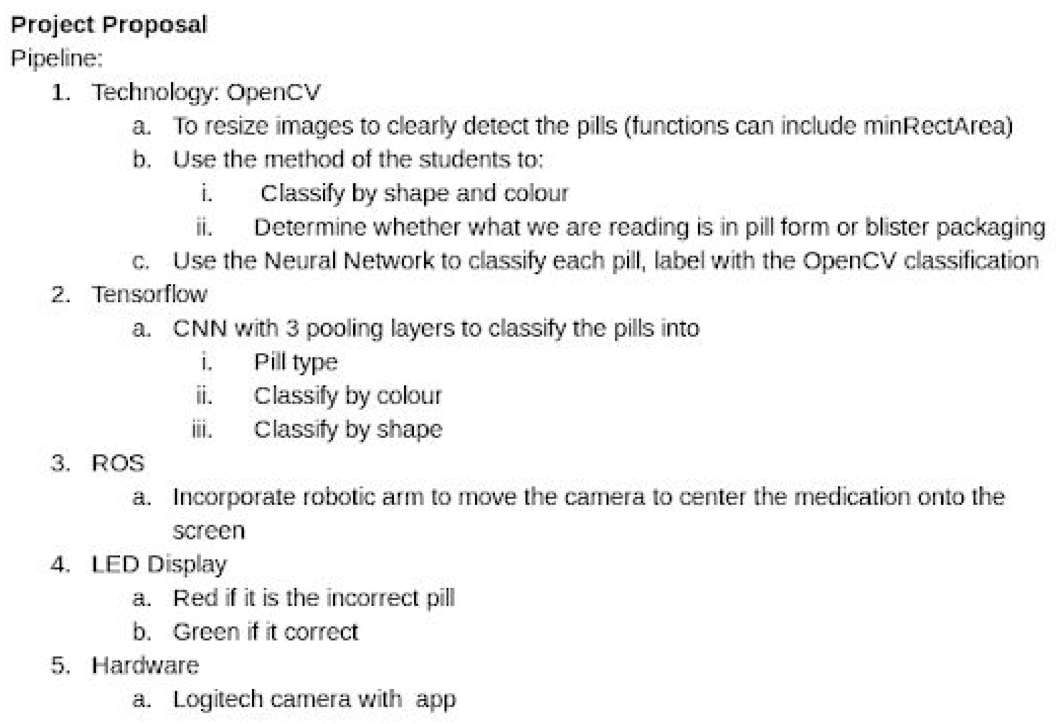
1. The detection system should be able to detect the correct pill regardless of the lighting or camera type use.
2. The detection system should integrate a functioning database to add and delete pills.

3. The detection system should interaction with a usable and responsive Graphical User Interface (GUI).
4. Classification should occur in less than one second in order for hospital workers to recognize a drug in an on time manner.

**C. Limitations:**

1. The system can only classify medication in pill form. It is not expanded to classify liquid medication such as cough medicine
2. Weather in blister or loose packaging, the pill must be visible to the camera. The software is not designed to detect pills hidden under a bottle.
3. It is assumed that the medication is measured at the proper amount

The initial proposal for the project was designed bearing in mind these project requirements.



**Project Proposal**

Pipeline:

1. Technology: OpenCV
    a. To resize images to clearly detect the pills (functions can include minRectArea)
    b. Use the method of the students to:
        i. Classify by shape and colour
        ii. Determine whether what we are reading is in pill form or blister packaging
    c. Use the Neural Network to classify each pill, label with the OpenCV classification
2. Tensorflow
    a. CNN with 3 pooling layers to classify the pills into
        i. Pill type
        ii. Classify by colour
        iii. Classify by shape
3. ROS
    a. Incorporate robotic arm to move the camera to center the medication onto the screen
4. LED Display
    a. Red if it is the incorrect pill
    b. Green if it correct
5. Hardware
    a. Logitech camera with app

*Figure 1: Initial Project Proposal*

The initial hardware setup of the project was a semi-controlled environment. Unlike MedR2 which had very controlled lighting in the setup, the hardware in this case was exposed to the natural lighting of the room throughout the day to emulate a realistic hospital environment. A webcam was fixed directly above the platform where the pill was located. LED lights were fixed on top of the contraption (the best type of light to minimize glare).

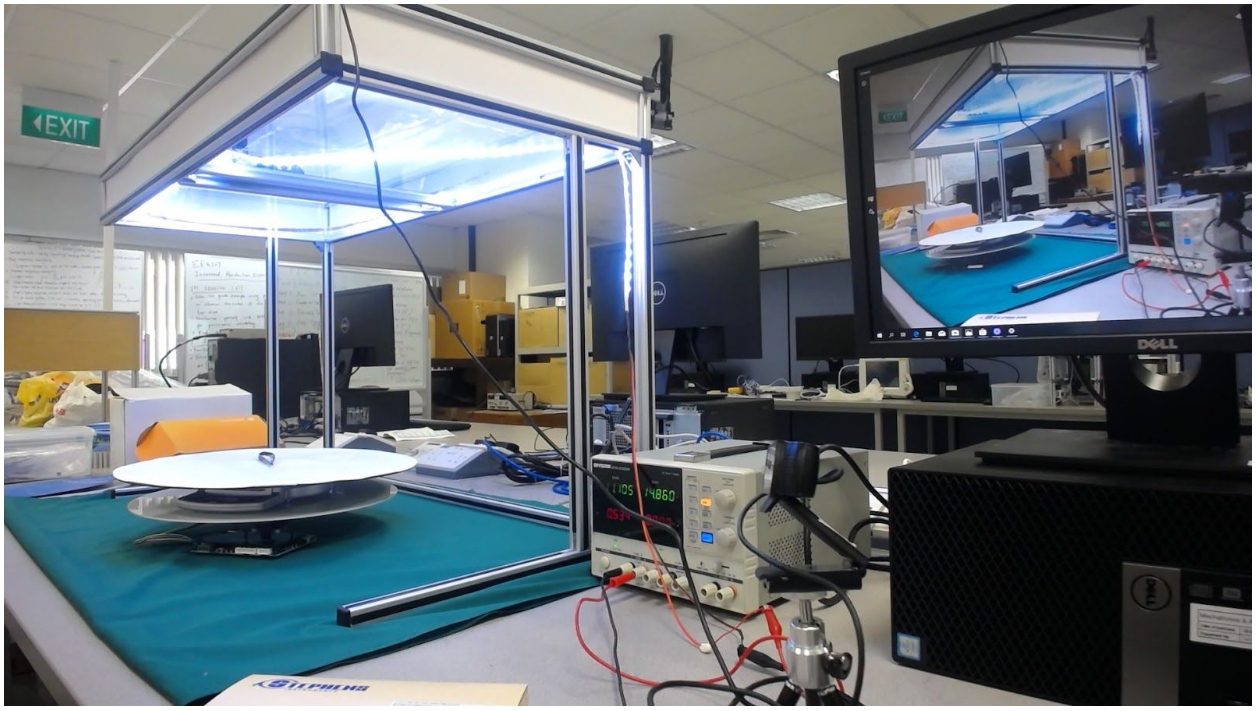


Figure 2: Hardware setup with the webcam, lights, power source, pill and platform.

## 4. Presenting the Findings

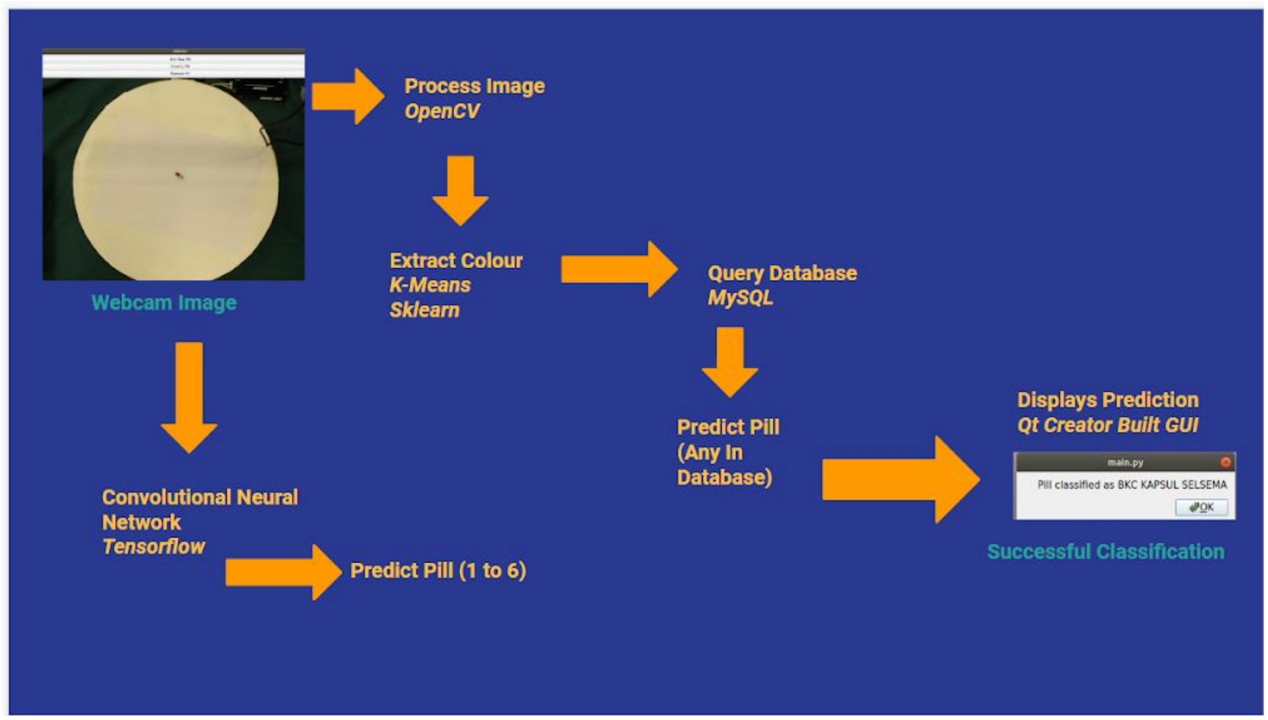


*Figure 3: Overall Software Architecture of the Project*

## Part A: Results From Convolutional Neural Networks

The first task for the project was building a convolutional neural network to distinguish between the 6 pills as provided.

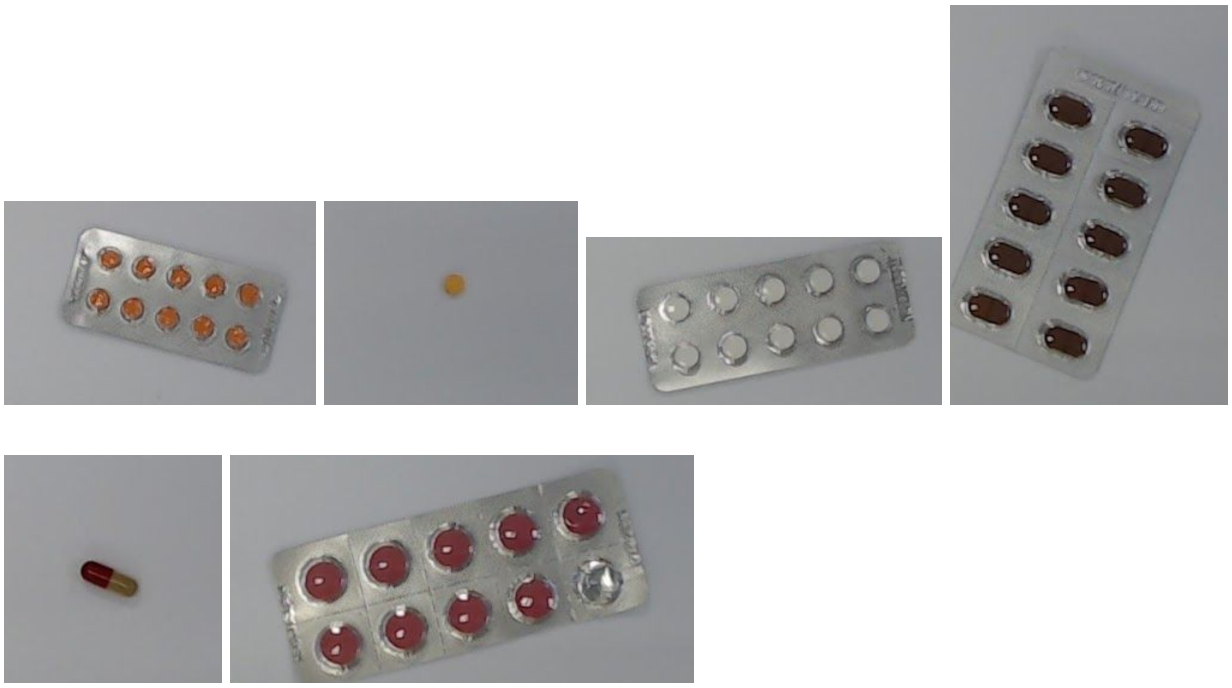
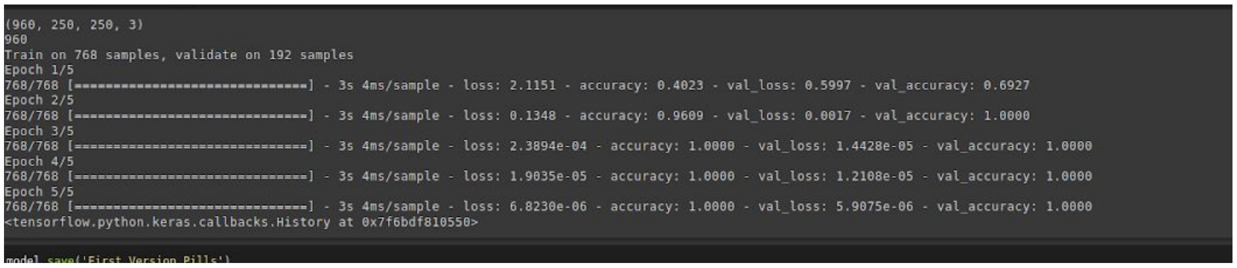


*Figure 4: Pills 1 to 6 Left to Right and Top to Bottom*

The given dataset from past experiments had 960 total images, which were cropped to focus on the pill and remove the noise.

The network was built with 3 convolutional layers with corresponding pooling layers. As the output was 6, the output layer was a softmax layer. The first model was tested with only a training set with a cross validation of 20%. The output provided an accuracy of

99.9%. The high accuracy rate was indicative of overfitting so the sample so it was decided to divide the data into a test set.

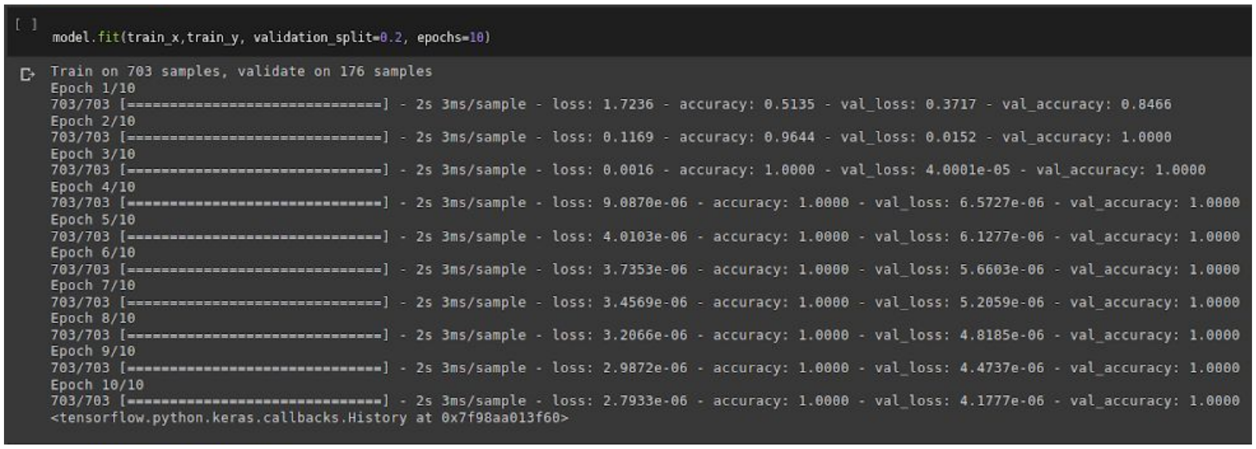


```
(960, 250, 250, 3)
960
Train on 768 samples, validate on 192 samples
Epoch 1/5
768/768 [==============================] - 3s 4ms/sample - loss: 2.1151 - accuracy: 0.4023 - val_loss: 0.5997 - val_accuracy: 0.6927
Epoch 2/5
768/768 [==============================] - 3s 4ms/sample - loss: 0.1348 - accuracy: 0.9609 - val_loss: 0.0017 - val_accuracy: 1.0000
Epoch 3/5
768/768 [==============================] - 3s 4ms/sample - loss: 2.3894e-04 - accuracy: 1.0000 - val_loss: 1.4428e-05 - val_accuracy: 1.0000
Epoch 4/5
768/768 [==============================] - 3s 4ms/sample - loss: 1.9035e-05 - accuracy: 1.0000 - val_loss: 1.2108e-05 - val_accuracy: 1.0000
Epoch 5/5
768/768 [==============================] - 3s 4ms/sample - loss: 6.8230e-06 - accuracy: 1.0000 - val_loss: 5.9075e-06 - val_accuracy: 1.0000
<tensorflow.python.keras.callbacks.History at 0x7f6bdf810550>

model.save('First Version Pills')
```

*Figure 5: Convolutional Neural Network Model 1 [6]*

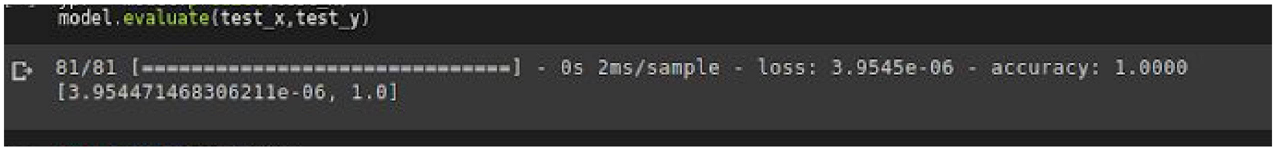
The second model divided the data in an 60/20/20 split between training, cross validation and test sets. The network was able to accurately predict the network.



```
model.fit(train_x,train_y, validation_split=0.2, epochs=10)

Train on 703 samples, validate on 176 samples
Epoch 1/10
703/703 [==============================] - 2s 3ms/sample - loss: 1.7236 - accuracy: 0.5135 - val_loss: 0.3717 - val_accuracy: 0.8466
Epoch 2/10
703/703 [==============================] - 2s 3ms/sample - loss: 0.1169 - accuracy: 0.9644 - val_loss: 0.0152 - val_accuracy: 1.0000
Epoch 3/10
703/703 [==============================] - 2s 3ms/sample - loss: 0.0016 - accuracy: 1.0000 - val_loss: 4.0001e-05 - val_accuracy: 1.0000
Epoch 4/10
703/703 [==============================] - 2s 3ms/sample - loss: 9.0870e-06 - accuracy: 1.0000 - val_loss: 6.5727e-06 - val_accuracy: 1.0000
Epoch 5/10
703/703 [==============================] - 2s 3ms/sample - loss: 4.0103e-06 - accuracy: 1.0000 - val_loss: 6.1277e-06 - val_accuracy: 1.0000
Epoch 6/10
703/703 [==============================] - 2s 3ms/sample - loss: 3.7353e-06 - accuracy: 1.0000 - val_loss: 5.6603e-06 - val_accuracy: 1.0000
Epoch 7/10
703/703 [==============================] - 2s 3ms/sample - loss: 3.4569e-06 - accuracy: 1.0000 - val_loss: 5.2059e-06 - val_accuracy: 1.0000
Epoch 8/10
703/703 [==============================] - 2s 3ms/sample - loss: 3.2066e-06 - accuracy: 1.0000 - val_loss: 4.8185e-06 - val_accuracy: 1.0000
Epoch 9/10
703/703 [==============================] - 2s 3ms/sample - loss: 2.9872e-06 - accuracy: 1.0000 - val_loss: 4.4737e-06 - val_accuracy: 1.0000
Epoch 10/10
703/703 [==============================] - 2s 3ms/sample - loss: 2.7933e-06 - accuracy: 1.0000 - val_loss: 4.1777e-06 - val_accuracy: 1.0000
<tensorflow.python.keras.callbacks.History at 0x7f98aa013f60>
```

*Figure 6: Convolutional Neural Network Model 2 Training [6]*

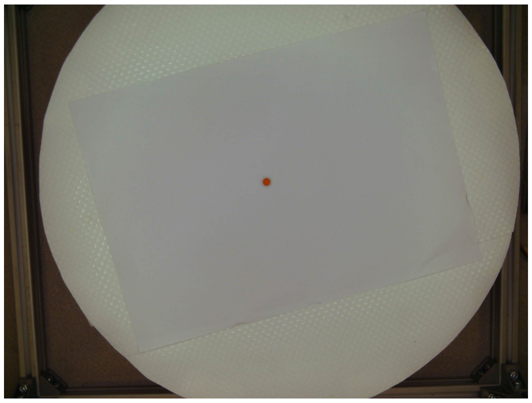


```
model.evaluate(test_x,test_y)

81/81 [==============================] - 0s 2ms/sample - loss: 3.9545e-06 - accuracy: 1.0000
[3.954471468306211e-06, 1.0]
```

*Figure 7: Convolutional Neural Network Test Set [6]*

On the surface, this appeared to be a highly efficient model. However, the speed at which the model was able to classify with 100% was indicative of overfitting. When the model was tested in a slightly different environment, it was unable to classify any pill correctly.



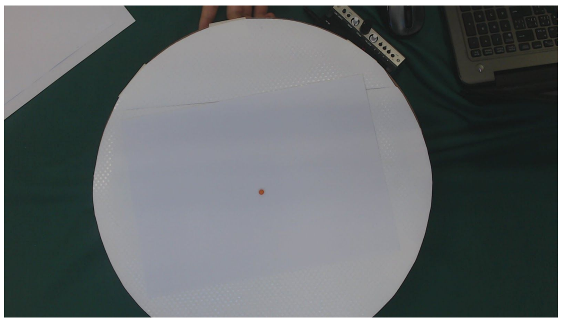


*Figure 8: Setup for original dataset (left), note the PCB camera and slightly different background vs. new setup (right) [6]*

This is likely due to 3 issues:

1. Different lighting from the two set ups
2. Size of pill occupied a very small space on the screen despite the cropping
3. The Convolutional Neural Network was built with flaws

Each of these assumptions were analyzed. Model 4[1] was built with an image processing function to isolate the pill on the screen before adding it to the neural network. This model was abandoned however, as overfitting was even more likely since images cropped exactly to the dimensions of their pills would created a dataset where every image looked identical (which made the system even more prone to over fit. This model was also unable to predict the newer images.

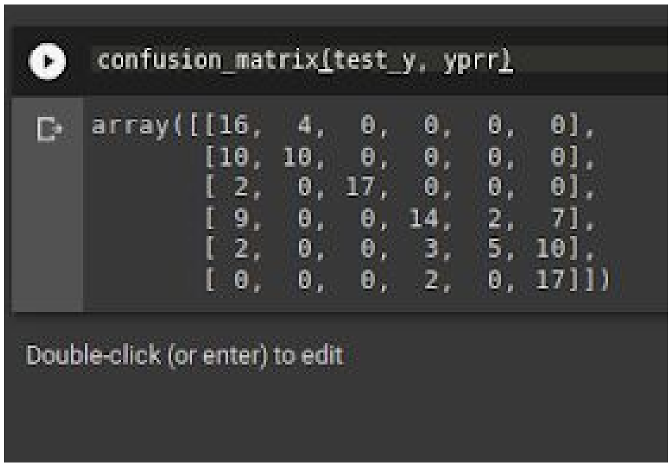
Finally Model 5 built a new dataset with 1920 pictures (double the amount from previously). Pictures were cropped to include all the pills before preprocessing. Initially, the test dataset had a 98.5% accuracy. However, testing the mechanism with new pictures taken in various types of lighting produced a 47% success rate in classification. The mechanism was tested on live images as well to lackluster results:

[1] Model 3 added some preprocessing along with incorporating object oriented programming to build the neural network in a simpler manner, resulting in little differences with Model 2.

| **Pill Number** | 1 - Dark Orange | 2 - Light Orange | 3 - White | 4 - Brown | 5 - Brown and Orange | 6 - Pink |
|---|---|---|---|---|---|---|
| **Blister with Light** | Not Work | Works | Works (rarely classifies as Light Orange) | Not work (classifies as Pink, Dark Orange) | Not Work | Partially Works, still misclassifies as orange |
| **Single with Light** | Not work | Works | Works | Not Work (misclassifies as white) | Not work (white) | Not work (white) |
| **Single without Light** | Not Work | Works | Works | Not Work (misclassifies as orange and white) | Not Work (white) | Not work (white) |
| **Blister without Light** | Not Work | Works | Works (rarely classifies as orange) | Not work (classifies as orange) | Not Work (white) | Does not work (orange/white) |

*Figure 9: Live classification under various lighting conditions. Tested blister and loose pills with light and without light*

In this table of results, the researcher defined “Works” as having an accuracy of 9 correct pills out of 10 pills. It is evident that the algorithm was biased towards classifying as white and brown. A new test dataset was created comprising of the above four categories.



```
confusion_matrix(test_y, yprr)
```

```
array([[16,  4,  0,  0,  0,  0],
       [10, 10,  0,  0,  0,  0],
       [ 2,  0, 17,  0,  0,  0],
       [ 9,  0,  0, 14,  2,  7],
       [ 2,  0,  0,  3,  5, 10],
       [ 0,  0,  0,  2,  0, 17]])
```

Double-click (or enter) to edit

*Figure 10: Confusion Matrix shows most of the pills are misclassified as Light Orange and Pink*

A deeper look was taken into the model itself. The following conditions were tested:

1. Modifying the dataset

Prior to the following changes, the dataset was modified to include images of each of the 4 categories in Figure 7. In itself, this decreased the accuracy of the validation set to around 80% by the 10th epoch. This proves that the previous data sets were very biased towards pills in that specific lighting.

2. Using Tensorboard to analyze the algorithm

Tensorboard is a tool effective for tracking the accuracy of a model throughout each epoch. By epoch 9 is is clear that both the validation accuracy and the loss have plateaued to a local optima. This indicates 9 epochs was optimal position to stop training.

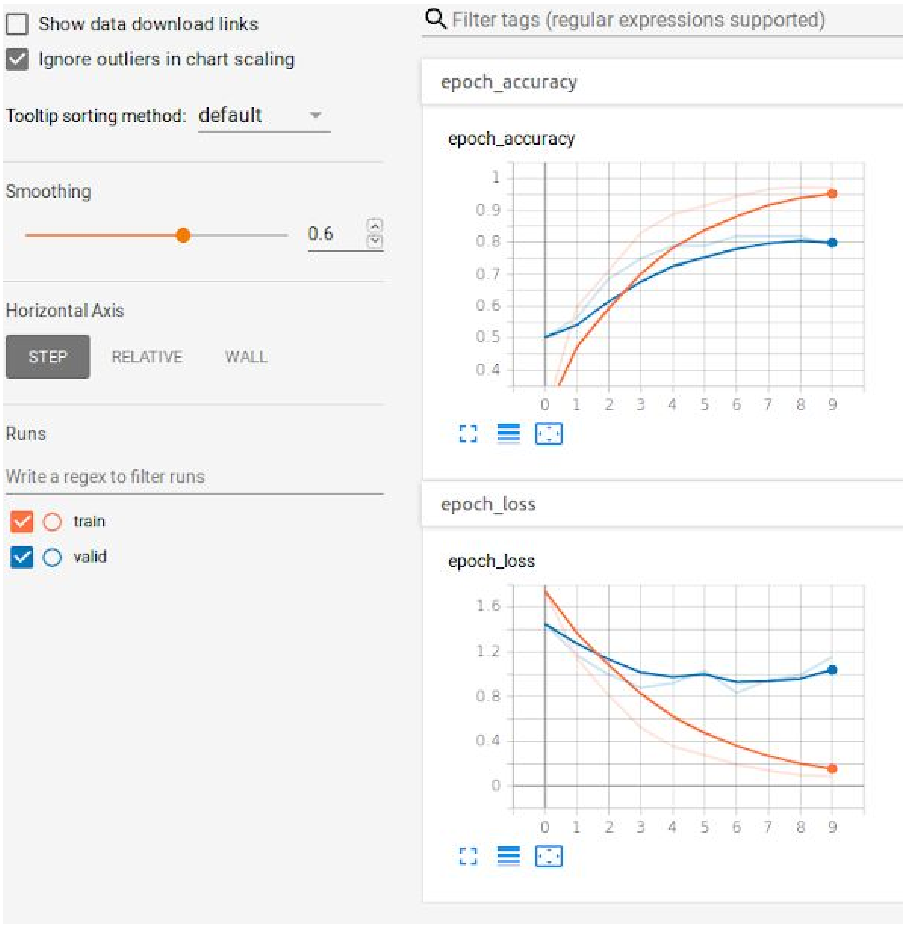

Show data download links
Ignore outliers in chart scaling [✓]
Tooltip sorting method: default
Smoothing 0.6
Horizontal Axis: [✓ STEP] RELATIVE WALL
Runs
Write a regex to filter runs
train [✓]
valid [✓]

*Figure 11: Modelling Accuracy After Ten Epochs [6]*

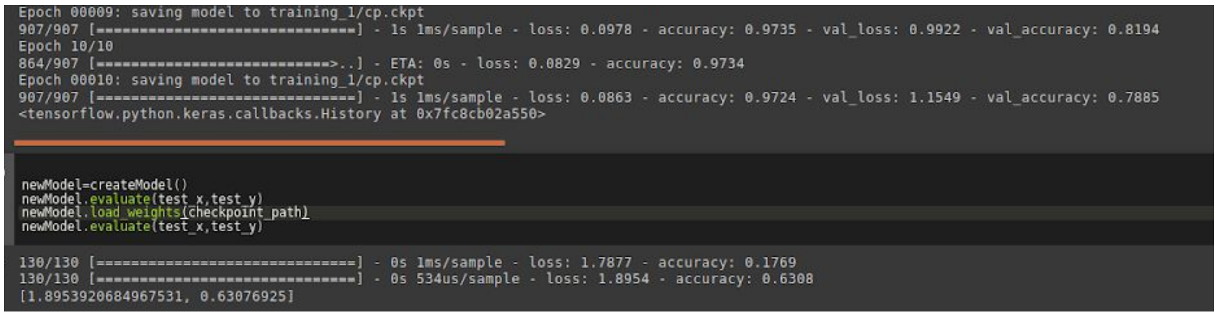
<u>3. Adding dropout layers</u>

In a neural network, dropout layers are very useful to prevent overfitting as they randomly “drop out” certain outputs in a neural network. The dropout layer was inserted before the first dense layer. It contributed to improving the neural network.

<u>4. Using Stochastic Gradient Descent instead of Batch Gradient Descent</u>

Batch gradient descent (involving passing data as batches) has been proven to have a lower accuracy than stochastic gradient descent. [7] This is due to the fact that stochastic gradient descent relies on calculating the cost for one sample at a time. Stochastic gradient descent takes longer to converge, however provides a better accuracy.

After testing each of these 4 changes, the accuracy of the neural network was increased to 64% on the pills photographed in diverse settings.



```
Epoch 00009: saving model to training_1/cp.ckpt
907/907 [==============================] - 1s 1ms/sample - loss: 0.0978 - accuracy: 0.9735 - val_loss: 0.9922 - val_accuracy: 0.8194
Epoch 10/10
864/907 [===========================>..] - ETA: 0s - loss: 0.0829 - accuracy: 0.9734
Epoch 00010: saving model to training_1/cp.ckpt
907/907 [==============================] - 1s 1ms/sample - loss: 0.0863 - accuracy: 0.9724 - val_loss: 1.1549 - val_accuracy: 0.7885
<tensorflow.python.keras.callbacks.History at 0x7fc8cb02a550>

newModel=createModel()
newModel.evaluate(test_x,test_y)
newModel.load_weights(checkpoint_path)
newModel.evaluate(test_x,test_y)

130/130 [==============================] - 0s 1ms/sample - loss: 1.7877 - accuracy: 0.1769
130/130 [==============================] - 0s 534us/sample - loss: 1.8954 - accuracy: 0.6308
[1.8953920684967531, 0.63076925]
```

*Figure 12: CNN Following The Changes [6]*

**Part B: Unsupervised Machine Learning Technique**

The second method utilized a combination of K-means clustering with sci-kit learn (an unsupervised machine learning technique) and image processing using OpenCV.

The following describes the image processing the pill would have to undergo in order to be added to the database or classified.

First the image was processed, searching for the pill in the wider background. This task was accomplished through the following process:

1. The image is converted to an HSV format and it is split into its H, S and V channels. This is to separate the pill from the surrounding lighting.
2. Thresholding the S channel with a binary threshold and Otsu binarization

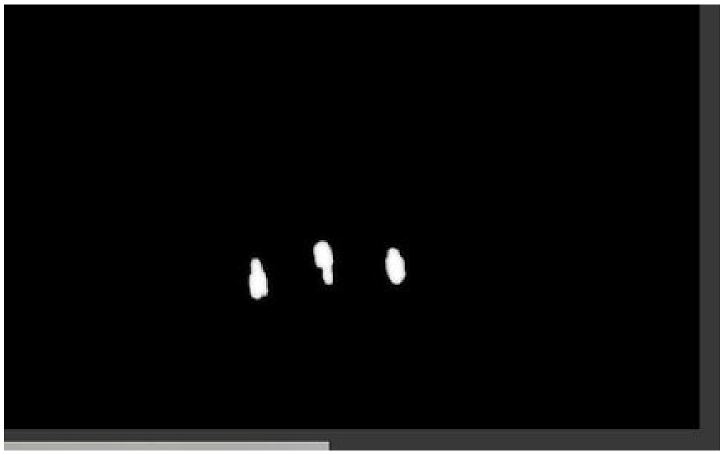


*Figure 13: Combining Otsu Binarization with a Binary Threshold*

3. A Gaussian Filter is applied to reduce the noise in the image.
4. A Bilateral Filter, also used to reduce noise is applied. Its primary purpose is to preserve edges while reducing noise to get a clear image.
5. Closing, a morphological filter is used to reduce holes in the image and “smooth the image”.
6. The contour(s) around the pill(s) are taken around the filtered image.

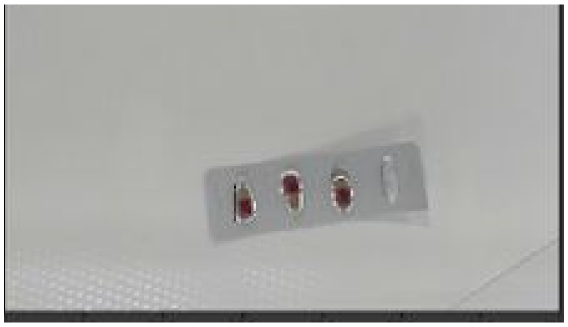


Figure 14: Contours drawn around the pills (for illustration purpose only)

7. Getting the MinAreaRect around the contour, rotating it so the image is square and cropping the original image to that dimension

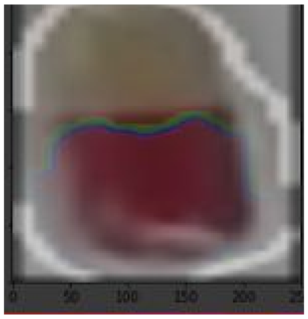
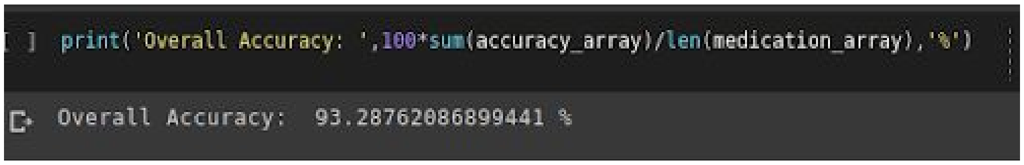


Figure 15: Image cropped and rotated

The RGB colours are then extracted from the pill using K-means extraction, and compared to the rest of the world. Initially, the image processing function was 93.98% accurate for the entire dataset.


```
[ ]  print('Overall Accuracy: ',100*sum(accuracy_array)/len(medication_array),'%')

     Overall Accuracy:  93.28762086899441 %
```
Figure 16: Overall Accuracy of Image Processing and Colour Extraction

However, the same problems were faced in terms of lighting. Testing on live images accounting for the various lighting conditions throughout the day greatly impacted the accuracy of the model. This is due to the fact that each channel, R, G and B has a value dependent on the lighting condition. To solve this problem, each image was analyzed in its HSV form. H (Hue) determined the colour of the image itself (Red, Blue, Green, Yellow etc.). Saturation reflected the amount of grey in the colour. Value was brightness of the colour (0 being a black image). The lighting conditions only affected the S and V channels. K-means clustering is designed to cluster at isolated regions on the images. If a pill was blue, red and had a grey background, K-means would manage to make clusters of 3 similar coloured regions. Each of these three colours would be analyzed and a Colour Classifying Value (CCV)  was set based on the analysis.

The algorithm first eliminated grey colours from the image. It was correctly deduced that the colour representing the grey background would have a low S value. The extracted HSV colour was determined to be grey if the saturation was below 25. The CCV  was set to an arbitrarily low value of 0 when grey was detected. This is the optimal way of querying the database.

Next the algorithm looked for the dark brown colours. Through analysis of the test set, it was discovered that the brown pill always had a V below 20. The CCV was set to an arbitrarily high value of 1000 when dark brown was detected.

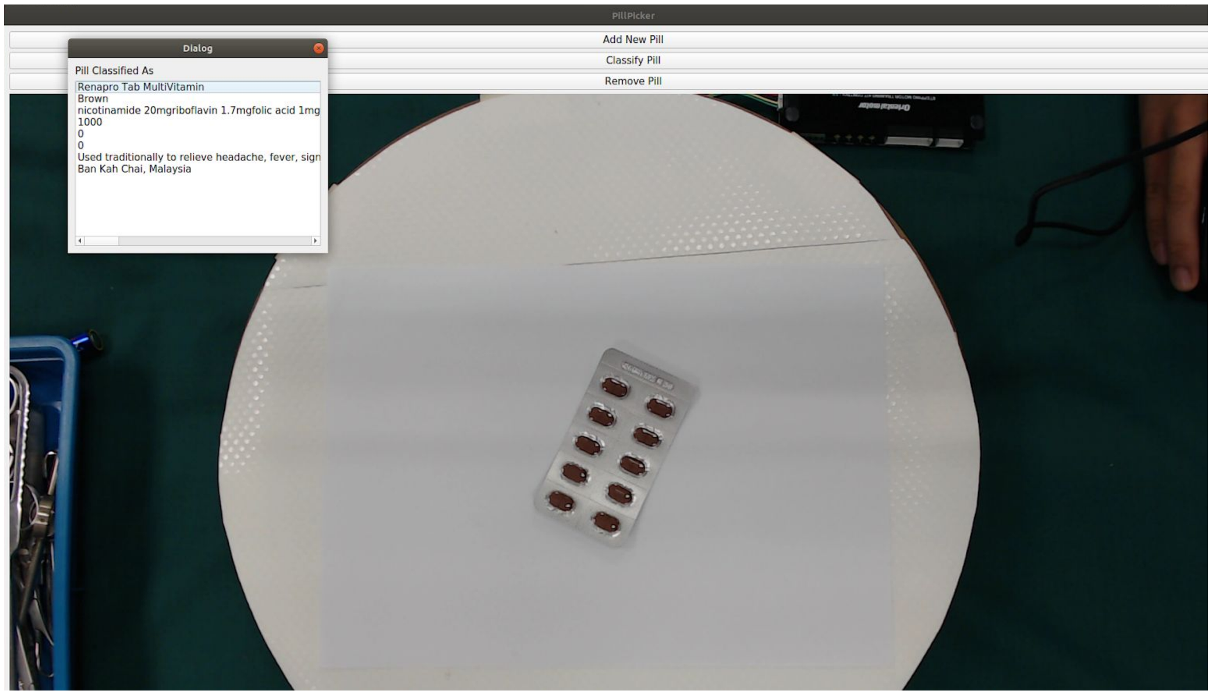
After skipping grey colours and classifying brown colours, the remaining colours were scanned for their hues. If the hue was within 10 of any of the stored hues in the database, the pill is identified to have that hue, and it is added to the CCV array. The CCV array is the key component to identifying the pill. When a pill is added, if the colour is not within the range of any of the other hues, the new hue is added to the database.

In the classification algorithm, after finding a list of 3 CCVs, the database was queried with a search for the pill with a matching CCV. Ultimately, the final product had a 94% accuracy with tested with live images of the pill.

**Part C: Graphical User Interface and Database**

The results  are displayed on the GUI created by PyQt. The GUI displays live video from the webcam of the pills being classified. The 3 main functions of the GUI is to:

1. Classify the pill



```
Pill Classified As
Renapro Tab MultiVitamin
Brown
nicotinamide 20mgriboflavin 1.7mgfolic acid 1mg
1000
0
0
Used traditionally to relieve headache, fever, sign
Ban Kah Chai, Malaysia
```

*Figure 17: The Renapro Tab Multivitamin is successfully classified in blister form*

When a pill is classified, its characteristics are extracted from a table in a MySQL database and displayed so the user can understand the exact nature behind the pill. Relevant features included in PillSafe are:

a. Name of the Pill

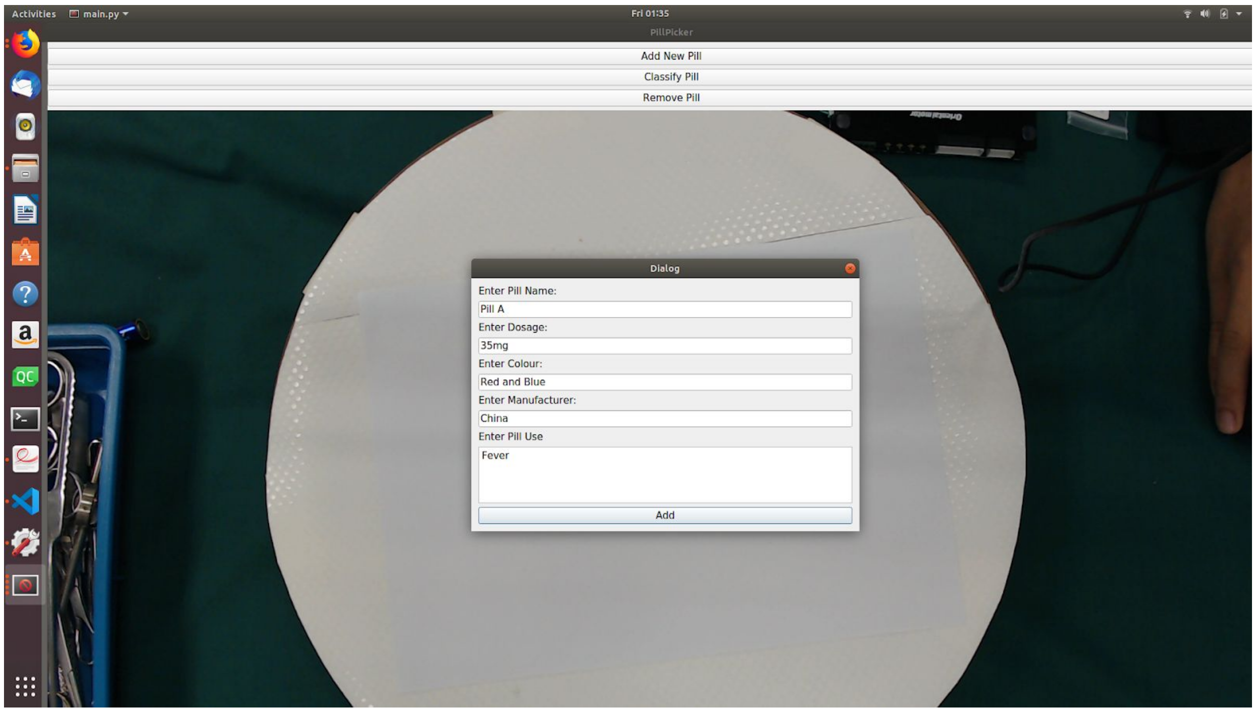
b. Dosage of the Pill

c. Colour of the Pill

d. Manufacturer of the Pill

e. Usage of the Pill

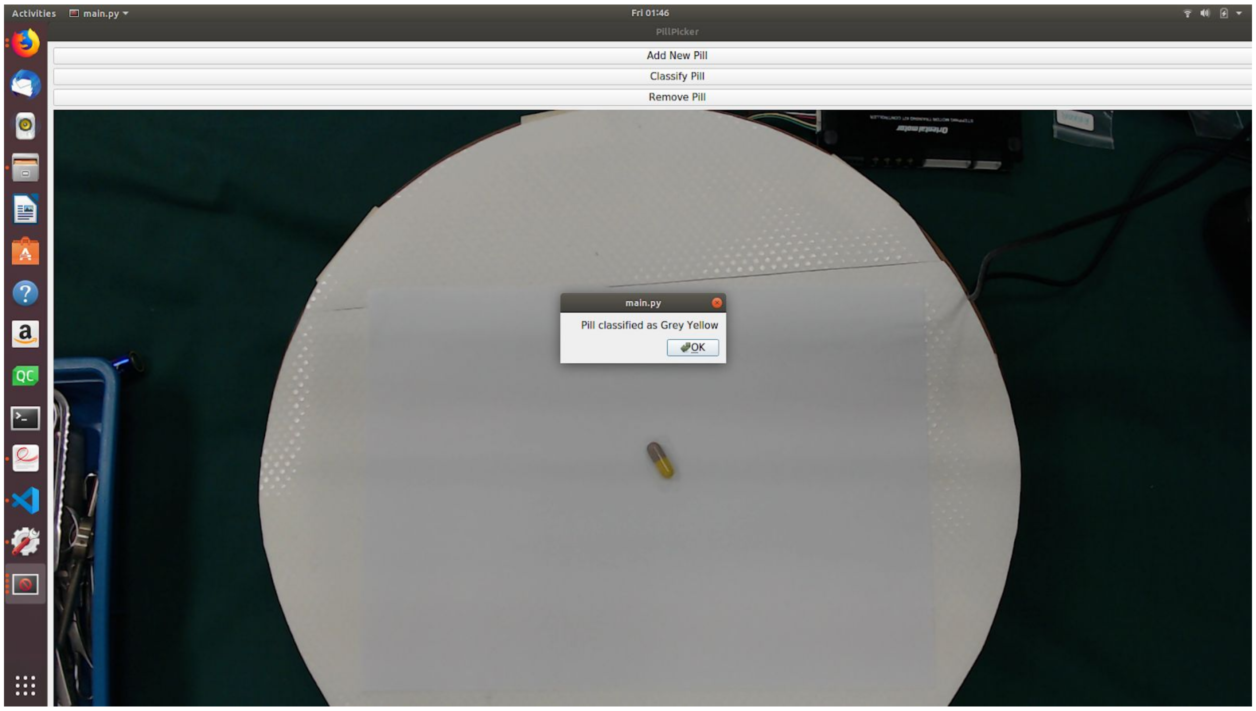
2. Add a pill



Enter Pill Name:
Pill A
Enter Dosage:
35mg
Enter Colour:
Red and Blue
Enter Manufacturer:
China
Enter Pill Use
Fever

*Figure 18: Dialog to add a pill to the database with options to fill out features*

When a pill is added to the database, the user will be prompted to fill out a form of the information behind the pill. This information will be added into the database, along with the CCV values extracted from the image.



Pill classified as Grey Yellow

*Figure 19: Correctly classifying a new pill (name arbitrarily set as "Grey Yellow")*

The final product differed from the initially proposal primarily in the aspects of hardware - less hardware was utilized as more emphasis was put on optimizing the classification algorithm.

# 5. Data analysis and discussion

With regards to the results of the convolutional neural network, it is apparent that using convolutional neural networks (CNNs) is an unreliable method of classification for medication pills. Although the algorithm can be improved upon with transfer learning and a larger, better dataset (machine learning datasets are advised to have at least 1000 samples for each category [8]), the amount of time needed outweighs the effectiveness of the approach. Not only does the network fail at classifying most of the pills due to the size, it also would take too much on the part of hospital technologists to train a dataset in a specific manner of lighting. The fact that a Convolutional Neural Network is limited to measure a given number of pills (in this study's case, 6 pills were studied) does not reflect the reality of modern medicine. Due to the fact that there are thousands of medications and medicine is constantly updated, there is no practical use to a CNN in this industry. Using image processing and K-means for feature extraction is the optimal method of classification due to the nature of the pill. It is fast (can extract features of a pill immediately), accurate, and works in various lightings.

Furthermore, while the accuracy of the neural network can theoretically be trained to be much better, the accuracy of 94% is sufficient for the detection purposes outlined in the scope of the project. As mentioned, PillSafe is designed to "double-check" a hospital worker's decision, and is not designed to work on its own.

# 6. Conclusion

The research conducted in this project found that colour extraction and image processing is the best way of classifying a pill. The system was able to function in a variety of lighting conditions using an average Logitech webcam. There are multiple routes in which this study can expand upon. Unfortunately, due to the limited time frame of this project, it was not within the realm of possibility. The GUI can be optimized to become more user friendly. Features that can be added include:

1. A search bar with autocomplete to look up a pill
2. Classifying multiple pills. This can be done by looking at all the contours in the photo and running the image processing algorithm on each one.
3. Improving the neural network to function in various lighting conditions, with larger datasets, more preprocessing and transfer learning.

However, the constraints and most of the objectives set during the design of the study have been fulfilled. The accuracy of classification has matched the accuracy of MedR2 with 94% of pills accurately classified. Neural networks were thoroughly examined, and the detection system must be able to recognize pills in both loose forms and blister packaging. The algorithm works in most lightings as demonstrated in the charts in figure (NUMBER). It integrates a  database that can be modified, and the GUI is responsive albeit simple. Ultimately, a functioning prototype has been built for pill detection. Due to necessity of a good pill verification system, there are very promising implications for the biomedical field and extensions that can be built upon this research.

# 7. References

[1] Donaldson M. S. (2008) "An Overview of To Err is Human: Re-emphasizing the Message of Patient Safety." In: Hughes RG, editor. Patient Safety and Quality: An Evidence-Based Handbook for Nurses. Rockville (MD): Agency for Healthcare Research and Quality (US); Chapter 3.

[2] Avery, T., Barber, N., Ghaleb M., Franklin, B. (2012) “Investigating the prevalence and causes of prescribing errors in general practice”, The University of Nottingham.

[3] Ong, B. Q. (2016) “MedR2: A Standalone System for Medication Recognition and Registration.” Department of Electrical and Computer Engineering, National University of Singapore

[4] Medsnap Website. Accessed May 22nd, 2019. Online: http://www.medsnap.com/

[5] Swisslog Website. Accessed May 22nd, 2019. Online: https://www.swisslog.com/en-sg/healthcare/products/medication-management/pillpick

[6] Tang Y. “Colab Notebook”

[7] Kapil, D. (2019) “Stochastic vs Batch Gradient Descent.” Medium Article. Accessed June 10th, 2019. Online: https://medium.com/@divakar_239/stochastic-vs-batch-gradient-descent-8820568eada1

[8] Kelly, J. (2016) “What is the minimum sample size required to train a Deep Learning model - CNN?” ResearchGate Forum. Accessed June 23rd, 2019. Online:

https://www.researchgate.net/post/What_is_the_minimum_sample_size_required_to_train_a_Deep_Learning_model-CNN

# 8. Appendix:

### A. Journal Made Throughout the Research Process

**Thursday May 23rd, 2019**

Tasks Completed:

- Loaded all the data was processed them by cropping and converting the image to hsv

To-do:

- Continue reading about OpenCV functions
- Investigate ROS
- Get a preliminary model working

**Monday May 27th, 2019**

Tasks Completed

- Implemented a functioning neural network that predicts the dataset with 100% accuracy
- Displayed pills, demonstrating their colours
- Saved model as “First Version Pills”

To-do:

- Split my data into test and training sets (find a non-manual way of doing it)
- Incorporate image datagen to create random generators

**Tuesday May 28th, 2019**

Tasks Completed

- Experimented with OpenCV, thresholding and Otsu
- Developed a better “process_image” function, incorporating those features
- Only Pill that does not work with this system is white pill (⅚ sucess)

To-do:

- Develop an algorithm to process the white pill
- Develop an algorithm to process blister packaging

**Wednesday May 29th, 2019**

Tasks Completed

- Split data into training and test sets, currently able to classify test set with 100% accuracy
- Created  a class to more conveniently organize dataset

To-do:

- Investigate TensorBoard, image preprocessing
- Try to get a larger dataset
- Once the OpenCV is working on all pill types, put that in my model

Long-Term Plan:

- Run OpenCV, keep going through the while loop until a pill is detected
- Once a pill/pill blister is detected, use model with image processing to detect pill
- Show on screen what pill was detected and a list of its attributes (use PostgreSQL)
- Once you close the screen, keep detecting object

**Friday May 30th, 2019**

Tasks Completed

- Able to crop most of the images to their pill dimension
    - Used Gaussian Blur to filter out noise
    - Bilateral filter on top of that
    - Closing to remove noise inside the white image

**Monday June 3rd, 2019**

Tasks Completed:

- Read more on YOLO, transfer learning and RNNs (consider implementing YOLO)
- Able to isolate contours on most images of blister package

To-do:

- Test CNN on individual pills

**Tuesday June 4th, 2019**

Tasks Completed:

- Model saves by weights, not entire model
- Attempted to read blister package with Neural Network, unable to classify any pills (predicts black pill every time for some reason)
- Tested camera at different angles, each having less success in returning a binary image, thus pill contour is seldom successfully extracted

To-do:

- Test with Logitech and PCB Camera, see which images are better
- Test images at different heights to see which images are better
- Test with blister and loose
- Create a larger dataset from the better method

All the images are shiit, how the hell did he get them to be such high qualiity

**Thursday June 6th, 2019**

Plan:

- Separate Neural Network from OpenCV functions
- Cross reference OpenCV and CNN

To-do:

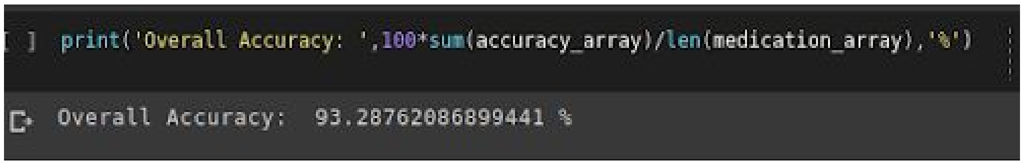
- Take a million more images of blister/non-blister at different angles
- Work the OpenCV to actually detect the colour of the image

**Friday June 7th, 2019**
**Presentation Summary**
Introduction:
- Purpose of this project
- Reasons for the project

Tasks Completed:
- Finished creating new dataset
- Neural Network also works on this dataset 100% accuracy on test and training

To-do:
- Extract colour from opencv

**Tuesday June 11th, 2019**
Tasks Completed:
- Successfully classified 93% of the pills using image processing
- 100% of pills are classified with ML (potential overfitting problem?)



```
[ ] print('Overall Accuracy: ',100*sum(accuracy_array)/len(medication_array),'%')

Overall Accuracy:  93.28762086899441 %
```

To-do:
- Begin putting the algorithm together with neural network to classify pills
    - Come up with a method of combining results from neural net and opencv

Combining Results:
Display OpenCV Results AND Neural Net results at the same time (model.predict and process_image) on the screen

**Friday June 13th, 2019**
Tasks Completed:
- Combined Results to display on screen
- Predicted random image

To-Do:
- Qt Display
- Test On live video

Qt Pipeline:
- Display constant video of the pill area until you press “Capture”
- Capture button will capture image of the pill and display it in a dialog box

- Instructions at the top to tell you what to do
- Dialog box will include essential pill information: Name, colour, use etc.
- Buttons:
    - Add Pill
    - Remove Pill
    - Capture

Paths For Improvement:

- Detect multiple pills at a time
    - YOLO?
- Detect in any lighting
    - Kmeans with HSV values instead of RGB values
    - Improve dataset for NN
- Detect from any range
    - Already works for OpenCV
- Add and Remove Pills
    - Improve Image Processing Techniques to account for shape, pill size (would conflict with “any range”), number of sides etc.

**Pills Table:**

| | Pill1 | Pill2 | Pill3 | Pill4 | Pill5 | Pill6 |
|---|---|---|---|---|---|---|
| Name: | Hurix’s Laxative Pill | Diclofenac | Salbutamol | Renapro Tab MultiVitamin | BKC KAPSUL SELSEMA | Vitamin B Forte |
| Dosage: | 5mg | 25mg | 2mg | nicotinamide 20mgriboflavin 1.7mgfolic acid 1mgcalcium pantothenate 10mgpyridoxine HCl 10mgbiotin 300ugthia | 300mg | B1 - 250mg B6 -250mg B12 - 1000mcg |

| | | | | | | |
|---|---|---|---|---|---|---|
| | | | | mine nitrate 1.5mgascorbic acid 97% 61.9mgcyanocobalamin 6ug | | |
| Colour: | Dark Orange | Orange | White | Brown | Red and Yellow | Dark Red |
| Shape: | Round | Round | Round | Elliptical | Elliptical | Round |
| Usage | **Bisacodyl** is used on a short-term basis to treat constipation. It also is used to empty the bowels before surgery and certain medical procedures. **Bisacodyl** is in a class of medications called stimulant laxatives. It works by increasing activity of the intestines to cause a | Diclofenac is used to relieve pain, swelling (inflammation), and joint stiffness caused by arthritis. Reducing these symptoms helps you do more of your normal daily activities. This medication is known as a nonsteroidal anti-inflammatory drug (NSAID). | It is used to treat asthma, including asthma attacks, exercise-induced bronchoconstriction, and chronic obstructive pulmonary disease (COPD). | Multivitamins are used to provide vitamins that are not taken in through the diet. Multivitamins are also used to treat vitamin deficiencies (lack of vitamins) caused by illness, pregnancy, poor nutrition, digestive disorders, and many other conditions. | Used traditionally to relieve headache, fever, signs of colds, cough and lethargy | This product is a combination of B vitamins used to treat or prevent vitamin deficiency due to poor diet, certain illnesses, alcoholism, or during pregnancy. Vitamins are important building blocks of the body and help keep you in good health. B vitamins include |

| | | | | | | |
|---|---|---|---|---|---|---|
| | bowel movement | | | | | thiamine, riboflavin, niacin/niacinamide, vitamin B6, vitamin B12, folic acid, and pantothenic acid. |
| Manufactured In: | Hurix, Malaysia | Drug Houses of Australia Pte. Ltd. | Sunward Pharmaceutical, Singapore | Samnam Pharm,South Korea | Ban Kah Chai, Malaysia | Drug Houses of Australia Pte. Ltd. |

Reshape Algorithm:

| | |
|---|---|
| X1: 0.23148 | Y1: 0.3645 |
| X2: 0.69444 | Y2: 0.625 |

For OpenCV Camera:

| | |
|---|---|
| x1=111 | y1=233 |
| x2=333 | y2=400 |

**Friday June 21st, 2019:**

Tasks Completed:

- Basic GUI display ready
- Started testing product
- Recoded image processing section to add and remove pills from the list

Problem:

- Lighting condition and camera type cause the product to achieve lackluster results on both ML and OpenCV ends

Potential Solutions:

- Their project kept constant lighting conditions
- Expand training set to include both
- Use HSV colour extraction instead of RGB extraction:
    - Lighting conditions only affect the S and V channels increasing accuracy

**Monday June 24th, 2019**

HSV Solution:

- Kmeans on two HSV values, take the one with the higher saturation value (meaning it is not the grey colour)
- Use the Hue value as basis for the values

**Tuesday June 25th, 2019**

Hierarchy:

1. If saturation is under 25, it is the gray colour around the pill, discard colour, continue
2. If the V is under 20 it is a darker colour, classify as Dark Brown
3. H within 5 of 23, classify as orange
4. H within 5 of 35, classify as light orange
5. H=340: Pink
6. If all the colours were discarded, pill is white

- Everytime there is a valid colour from the above list (meaning it is not discarded as grey), add it to the list of pill colours (this will account for when two colours are detected)
- Do not add if already added

**Wednesday June 26th, 2019**

Structure of OpenCV Code:

Classes:

- pillAccuracy: keeps track of how many pills are classified correctly
- processImage: crops, filters and classifies the pill

Tasks Completed:

- Successfully classified 99% of the new dataset using Neural Network and 90% using Image Processing
- New dataset includes images of different lighting for blister and for loose

Notes:

Major Problems:

A. Neural Network continuously misclassifies as Orange and White

   Potential Reasons:

   1. Too many images in orange and white dataset, overfitting
   2. Wrong model?
   3. Poor quality images in most of dataset
   4. Too many epochs
   5. Not normalizing, dividing by 255.0

B. S value may be too high

C. Contour Image is contour taken around the empty blister in Pill number 6

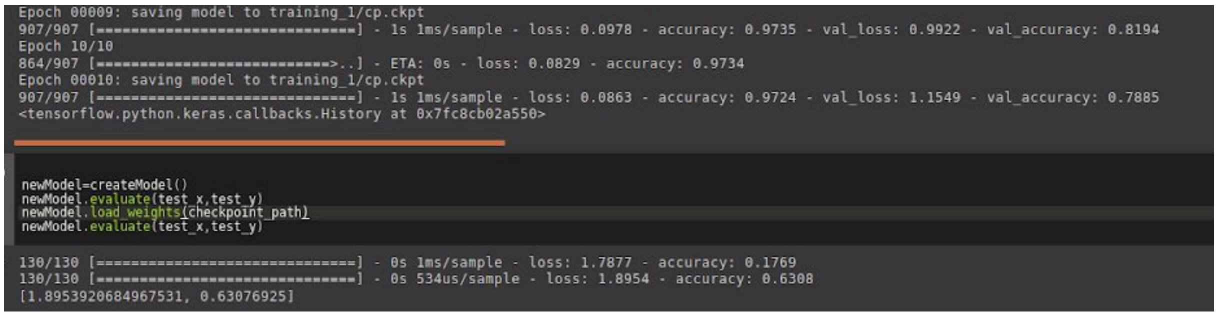
To-do:

- Optimize training set, less epochs for neural network (take more pictures)
- More dropout layers
- Less convolutions/pooling
- Attempt transfer learning

**Friday June 26th, 2019**

**Test Results:**

- Less epochs did not matter much
- Changing the dataset did not do much either
- Reducing the batch size improved accuracy by 7 percent
    - **Batch size**: number of training examples used in one iteration
        - Larger batch size = lower accuracy, faster training
    - **Stochastic Gradient Descent**: use one example every iteration
        - Need to randomly shuffle data, noiser path than batch gradient descent
- Removing irrelevant data had minimal effect
- Final accuracy after adjusting for all changes was 63% on the new test set



```
Epoch 00009: saving model to training_1/cp.ckpt
907/907 [==============================] - 1s 1ms/sample - loss: 0.0978 - accuracy: 0.9735 - val_loss: 0.9922 - val_accuracy: 0.8194
Epoch 10/10
864/907 [===========================>..] - ETA: 0s - loss: 0.0829 - accuracy: 0.9734
Epoch 00010: saving model to training_1/cp.ckpt
907/907 [==============================] - 1s 1ms/sample - loss: 0.0863 - accuracy: 0.9724 - val_loss: 1.1549 - val_accuracy: 0.7885
<tensorflow.python.keras.callbacks.History at 0x7fc8cb02a550>
```

```
newModel=createModel()
newModel.evaluate(test_x,test_y)
newModel.load_weights(checkpoint_path)
newModel.evaluate(test_x,test_y)
```

```
130/130 [==============================] - 0s 1ms/sample - loss: 1.7877 - accuracy: 0.1769
130/130 [==============================] - 0s 534us/sample - loss: 1.8954 - accuracy: 0.6308
[1.8953920684967531, 0.63076925]
```

Figure: Final Accuracy for Neural Network: 63%

**Tensorboard Analysis:**

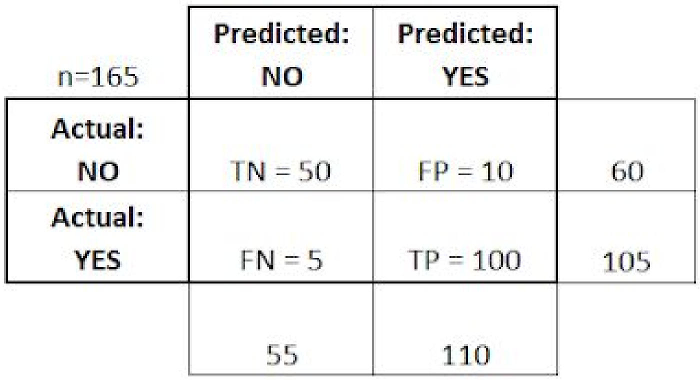
How to set up TensorBoard:

1. Write the following code, include tensorboard as a callback variable
   NAME="Model 5 Pills {}".format(int(time.time()))
   tensorboard=TensorBoard(log_dir='logs/{}'.format(NAME))
2. Download into folder called logs on desktop
3. Run the code:
   tensorboard --logdir=logs

*Figure: Modelling Accuracy After Ten Epochs*

Loss was quite high for both training and validation, increasing for validation while training kept a slight decrease. This demonstrates

**Confusion Matrix Results:**



| n=165 | Predicted: NO | Predicted: YES | |
|---|---|---|---|
| Actual: NO | TN = 50 | FP = 10 | 60 |
| Actual: YES | FN = 5 | TP = 100 | 105 |
| | 55 | 110 | |

*Source: [https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/](https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/)*

- All diagonal means 100 percent accuracy (as seen in the previously heavily biased dataset)
- Our confusion matrix showed a heavy bias, misclassifying as Pill 1 and Pill 6

*Figure: Confusion Matrix for Neural Network*

**Week 6 Conclusion**: CNN is an unreliable method of classification, also would take too much on the part of hospital technologists to train a dataset in a specific manner of lighting. Using image processing and unsupervised machine learning for feature extraction is the optimal method of classification due to the nature of the pill. It is fast (can extract features of a pill immediately), accurate, and works in various lightings. The rest of the research project should focus on creating a product from the latter method.

**Tuesday July 1st, 2019**

**Week To-Do:**

- Put everything together in a GUI application with Qt
- Create a database to enter/remove pills, be able to Query database
- Start final report

**SQL Overview:**

- mySQL is server report
- Language: SQL query language (you can use SQL language for mySQL, postgreSQL etc.)

**Setting Up A Database:**

1. Create a database:

```
CREATE DATABASE testdatabase;
SHOW DATABASES
```

2. Create tables in database, each comma separates a column

```
CREATE TABLE books(
        -> title VARCHAR(50) NOT NULL,
        -> price  INT NOT NULL,
        -> language VARCHAR(50) DEFAULT "ENGLISH",
        -> author VARCHAR(60) NOT NULL)
        -> ;
```

3. Insert into Database

```
INSERT INTO books VALUE("Harry Potter", 50.00, "Portuguese", "JK Rowling")
```

4. Query Database (extracts data and presents it in a readable form)

```
SELECT * FROM books
```

**Monday July 8th, 2019**
Tasks Completed:
- Started report

To-do:
- Put GUI and database together

**Tuesday July 9th, 2019**
**Qt Creator Notes:**
QMainWindow:
- Built around common needs of a main window (built with

QWidget:
- Base class of all user interface objects

QDialog:
- Based on QWidget designed to be shown as a window

How to convert Qt Design to python file:

```
pyuic5 test.ui -o test_ui.py
```

**Widgets To Add:**
- Search Bar with pop up display pill information
- Form to add information about the pill

Tasks Completed:

- Working functionality where GUI can interface with the database (able to search and insert), after editing the code to add a constructor

To-do:

- Figure out how to connect image processing functions to database to search for all
    - Problem:
        - outliers of white pill and brown pill do not rely on the hue
        - Multi coloured pills
    - Solution:
        - Set Hue as 0 if white, 1000 if brown
        - Have three hues in database for each colour extracted
        - Query database for each of these

**Thursday July 11th, 2019**

To-do:

- Separate  image processing function so it can extract HSV without classifying at the same time
- Make the classifying search through database instead of temporary dictionary

Plan:

- We get extract 3 HSV values using Kmeans
- Insert 3 values in the database
    - Use the same structure of identifyColour
    - First check if each value is brown or white
        - If it is, set 1000, 0
    - If not set the normal hue value
    - Eg. Pill1 Test will be 33, 0, 0
- Classifying
    - The three values in the pill to detect must match in any order with whats in the database
    - If we extract 33, 0, 0 (converted) as the three colours, then we have detected Hurix's Laxative Pill. Get that from database

**Friday July 12th, 2019**

Tasks Completed:

- Tested final product with a success rate of
    - 94% on already classified pills
    - 90% on 5 new pills inserted into database
- Finish and Submit report and log sheet
- Push code to github for Doctor Tang to review