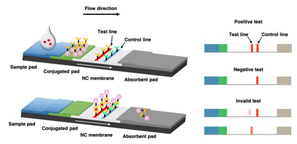
Rapid Diagnostic Test Prediction via Convolutional Neural Network

Daniel Leon, Applied Mathematics 563

Smartphone images of rapid diagnostic devices (RDTs) can be automatically interpreted by convolutional neural networks

- RDTs are a molecular detection platform that have a broad application set. For example
  - Infectious disease: COVID-19, HIV, Malaria
  - Home-pregnancy tests
  - Tests for street-drugs
  - Tests for water impurities, etc.
- The need for quick and accurate interpretation of RDTs could be advanced via machine learning techniques and could have a strong positive impact on global health.



[1]

- Smartphones offer a practical method for image collection, but variable conditions present many challenges:
  - Many models of phone
  - Lighting conditions
  - Orientation and size of RDT in images, etc.
  - User access to quality devices and network connection

The Dataset

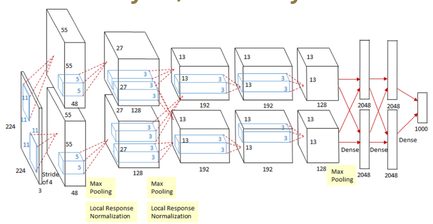
COVID-19 RDTs from Quidel Corp using recombinant Ag

- Over 2,000 COVID-19 RDT strip images
  - 10% CV/10% testing set splits
- 5 phone models, incl. Apple, Samsung, Google
- 5 lighting conditions, incl fluorescent, incandescent, halogen, ambient, and multiple lighting angles
- 10 analyte concentrations, negative control, & invalid
  - Max conc. 10 ng/ml
  - Min conc. 10/2^9 = 19.5 pg/mL
  - Serial dilutions by ½ times 10
  - Negative and invalid (no control line) included
  - Total of 12 “classes”

[1] Wesley Wei-Wen Hsiao et al. (2021). ISSN: 2079-6374.

Convolutional Neural Network

5 convolution layers, and 3 fully-connected layers

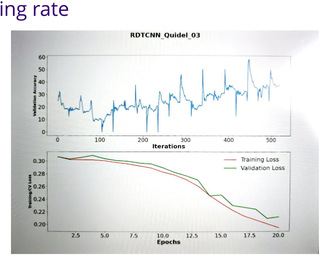


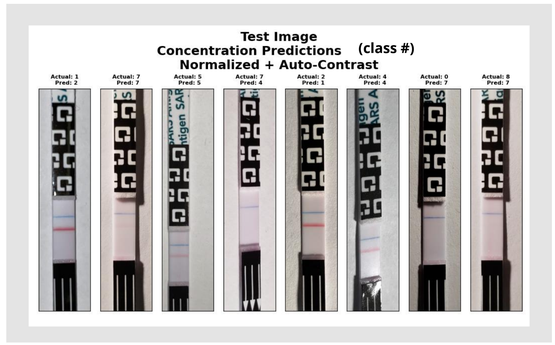
- Benefits from still being small enough to run on home computer
- Trained 4 GB NVIDIA gpu, and 12-thread multithreading
- My architecture is based on AlexNet, but modified to accept rectangular input, aspect ratio: 4.3
- Preprocessing included scrubbing mislabeled data, cropping the relevant region (went with 350 px by 1500 px), application of auto-contrast
- Building custom data-loaders
- Originally trained as a classification problem, but switched the problem to one of regression with better results

Results: Classification

With the original classification method, I was able to achieve 35% accuracy in classifying the images to one of the 12 classes.

- For classification, I used the cross-entropy loss function and predicted accuracy by extracting the highest probability class as the prediction
- The loss curve indicates that learning was occurring, but that it was slow to take off.
- This could be addressed several ways, but the easiest is to adjust the initial learning rate
- I had not developed my plot of true class vs predicted class by this point, but that would be a task for further development





Test set predictions using the classification approach. In this batch we see 3/8 correct predictions

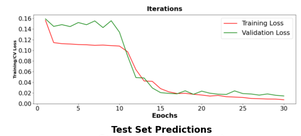
Results: Regression

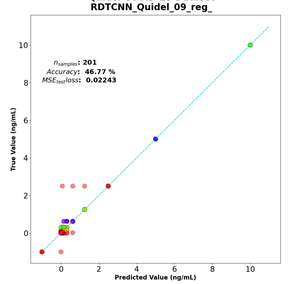
Since all the real positive test results are numeric, I took the advice that the task was suited for regression.

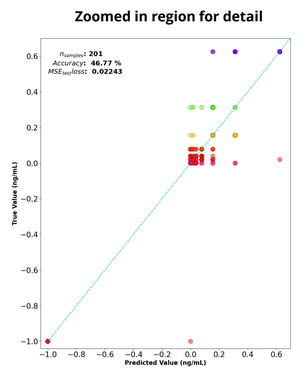
- With regression, we switch the output layer to give dimension 1, instead of the total number of classes
- Transformed data to be within range $[0, \log_2(10)]$
- The loss function was switched to mean-squared-error loss
- Learning rate scheduler decreased every 5 epochs
  - Loss curve shows where learning takes off

MSE is low and predictions are good at higher concentrations, but still have trouble with high accuracy in the tightly

Test set predictions using the regression approach. I chose some of the more difficult images, some that were very dark, some with bright glare, some where the cropping was not perfect



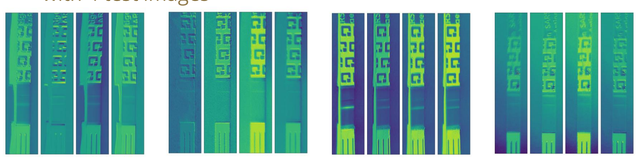




Filter Inspection

Looking at the filters in the convolutional layers gives an idea of what the model is working with

- Shown are the first 16 filters of the first convolutional layers with 4 test images



- The test and control lines are seemingly enhanced under certain filters, which is great
- The labels on the test images stick out a lot, but they are uniform across all cases
- The shadows in the first set of four images are very noticeable

Conclusions & Future steps

I have demonstrated RDT test result predictions are possible using CNNs

- 46% accuracy with a higher percentage being within one class of true
- Model did very well with predicting invalid tests
- The number of false positives is higher than I want
- Exact analyte concentration can vary with many factors

Next steps

- Better preprocessing, including more accurate cropping, better contrast correction, and other methods could likely yield better results
- Improved model architecture could address the vanishing gradient problems that I was constantly combatting
- Execute training on a cluster to allow larger models and bigger batches

Special thanks

- Audere for granting access to the dataset
- Lutz group members for their support