Project Individual Reports

The inspiration of the project was that we want to create an object detection program. However, we do not have the proper data to train a network for the object detection. That give us an inspiration to create an object detection model using trained CNN or MLP classifiers. The idea of the project is if we have a well trained deep neural network classifier, we can get locations and shapes of the objects of our classifier without additional training and a little addition rescues. It will help reduce the need of human annotation, also do not require backpropagation to calculate the bounding box. The main task of this project is to train a network, developing the Cropping – Predicting – Mapping algorithm, analyzing the results, evaluating potential improvement.

Individual work:

1. Train and test a classification model.
2. Use trained model as base to develop cropping and mapping algorithm to do object detection.
3. Design the hyper perimeter of cropping and mapping algorithm
4. Design the output of the program, so it can be output can be interpreted.

I designed our own Cropping – Predicting – Mapping algorithm to map the output to each class output. It's the same concept as the CNN network where the algorithm have convolving kernel and stride over the original image, cropping the kernel size of original image, put the cropped image into the classifier to predict the image class. The assumption is the classifier have all the classes in the image.

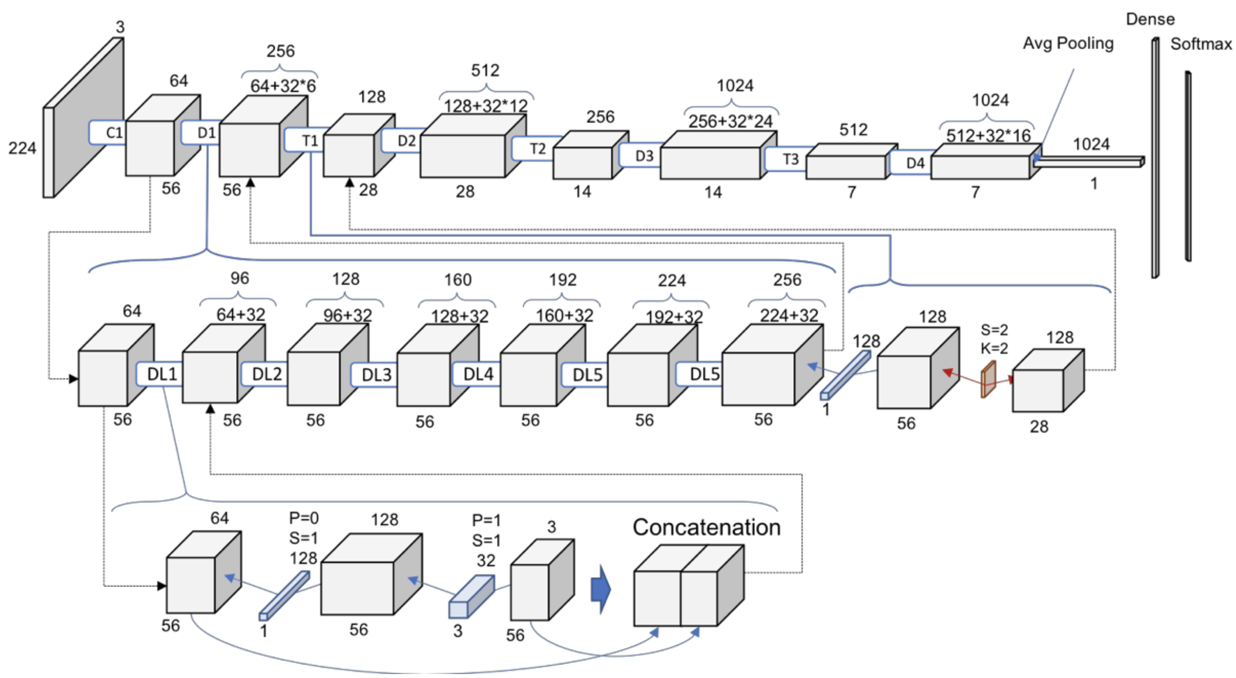
For Cropping, the function start the top left corner of the image move to the right. The width and high of the kernel is depending on the input image and the stride is based on kernel size. The bigger kernel size can go through the image faster, and product the mapping for bigger objects in the image. However the smaller kernel size produce better result base on the finding so far.

Kernel size and stride for cell data example:

```
intal_start_right_point = int(width * 0.025)
moving_right = int(intal_start_right_point * 0.2)
intal_start_bottom_point = int(height * 0.025)
moving_down = int(intal_start_bottom_point * 0.2)
```

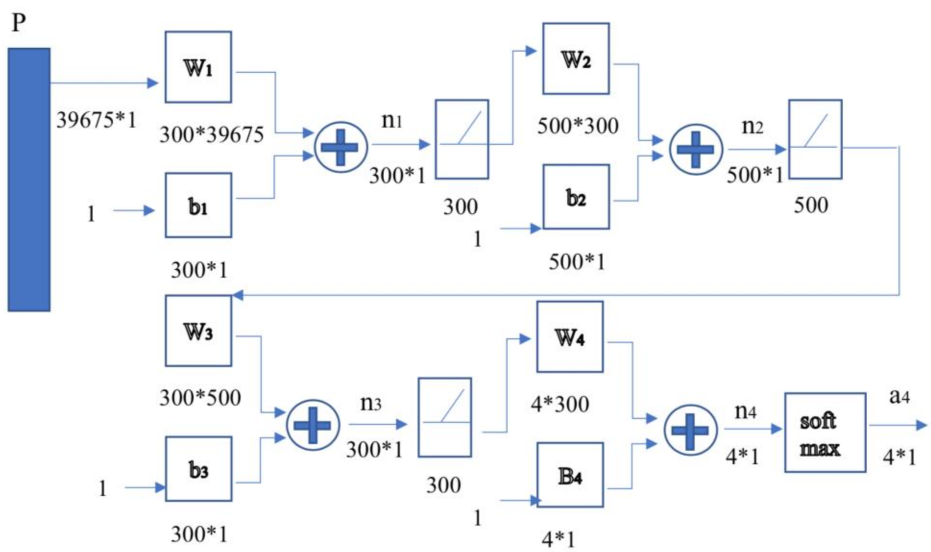
For predicting, I used DenseNet121 to train on the fruit dataset, DenseNet121 is a CNN network that pass the output of previous network layers to the next layer. I added additional dropout layer and Global average pulling layer at the end to prevent overfitting. The result was 98.3% accuracy and 0.96 F1-score. The problem of this dataset for our purposes is the image is in isolated white background, that doesn't prove enough background noise to help training the model.

DenseNet Architecture :



I used my own Exam1 MLP model to train on the cell dataset form Exam1, my MLP model did well one predicting the exam's holdout set with 96% accuracy and 76% F1-score. It's not the best F1-score due to the imbalance of the data, and how little one image class is. The model also has Dropout layers and BatchNormalization layers to help reduce overfitting and help with neuron imbalance and speed up the training significantly. This model has really good results for 'Red blood cells', and that really helps Cropping – Predicting – Mapping algorithm.

MLP Architecture:



For mapping, I create a dictionary of class maps as output, the perdition in mapping also have a threshold for the softmax output, so we can be more confident in the output class and class maps. It uses the same kernel and stride from cropping to go through the image, start the top left corner of the image move to the right then move down.

I trained a Densenet121 to predict 120 classes of fruits. I trained a MLP model to predict 4 classes of cells. I wrote the base function Cropping – Predicting – Mapping algorithm. I produce all the outcome of the cell's data.

The result of our Cropping – Predicting – Mapping algorithm really convinced us this will be a the right direction to move forward. Figure 1 is the original cell image with annotation. Figure 2 is the red blood cell class map output produced by the Cropping – Predicting – Mapping algorithm. The result shows the algorithm can map out each cells correctly and the cell that did not map out properly is the green color cells with is trophozoite, and the algorithm was able to ignore the noise background.

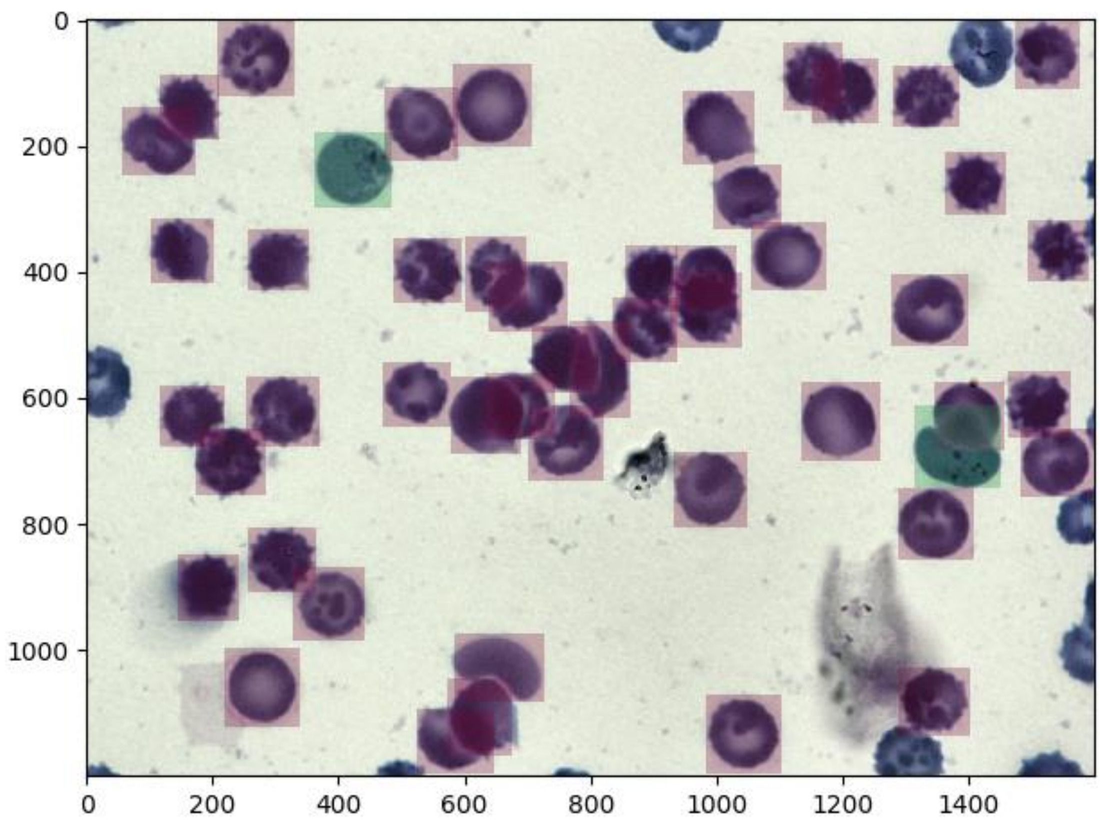


Figure 1

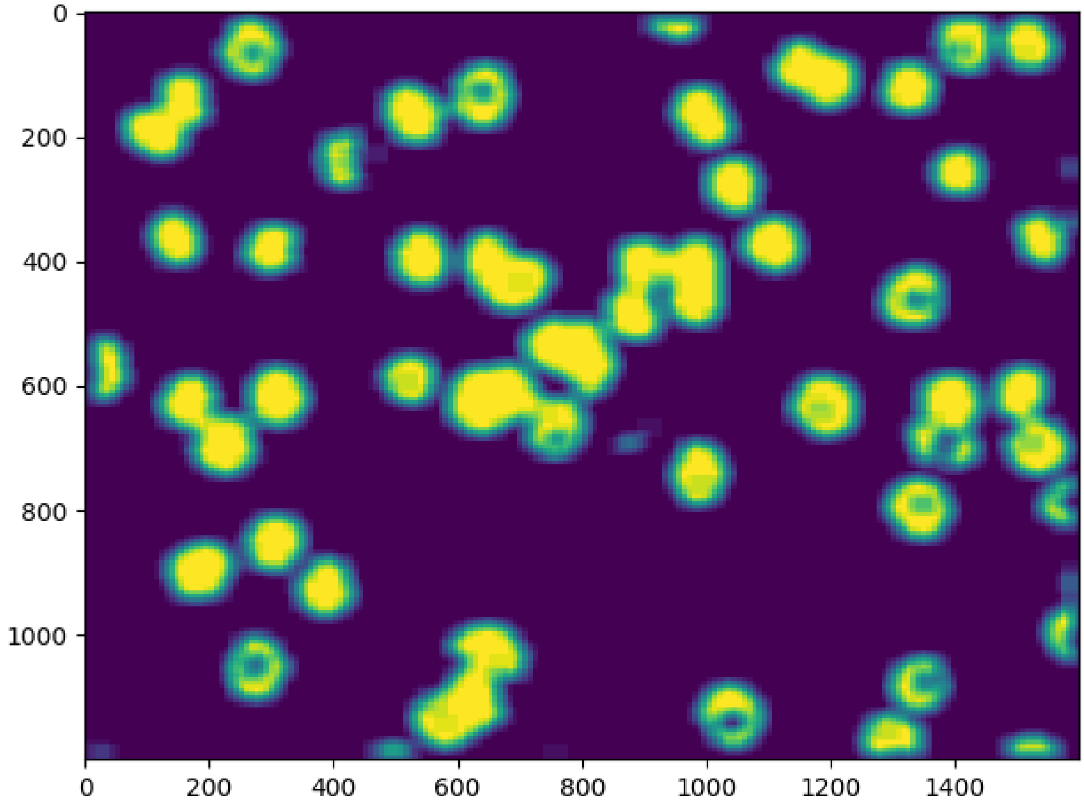


Figure 2

Figure 3 is the original cell image with annotation that's all red blood cells. Figure 4 is the red blood cell class map output produced by the Cropping – Predicting – Mapping algorithm. The result shows the algorithm can map out each cells correctly and the ones that researcher didn't annotate also show the up on the red blood cell class map. The result of the mapping is prefect. However, due to the limitation of the training dataset, we don't not have null class, which forced one of the four class to be identify as background. For this cell's example, trophozoite class is being misidentify as background, which Figure 5 shows.

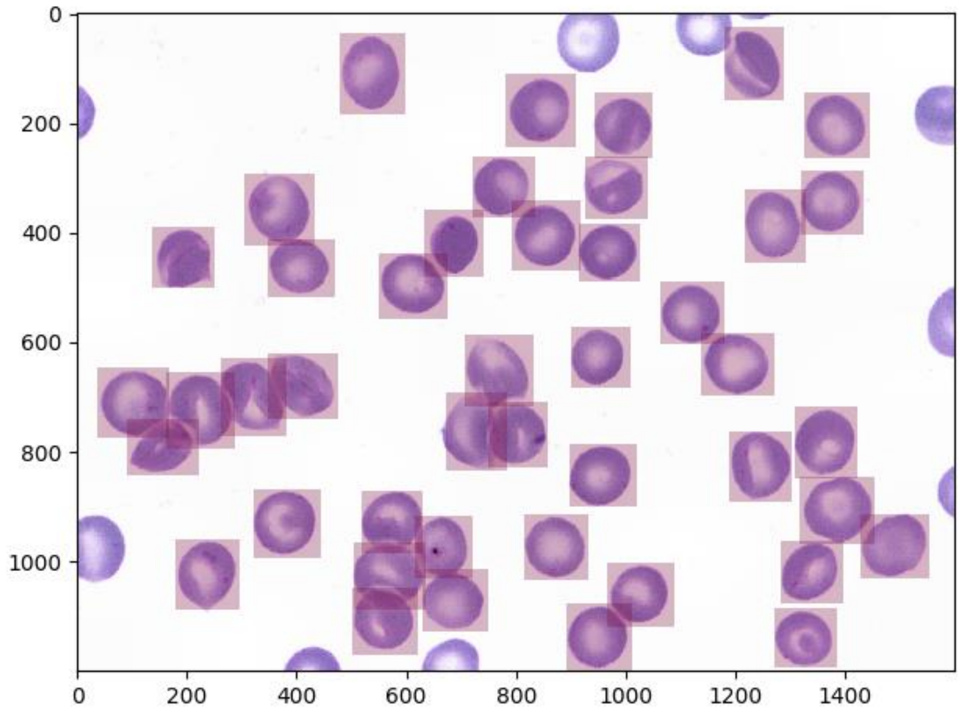


Figure 3

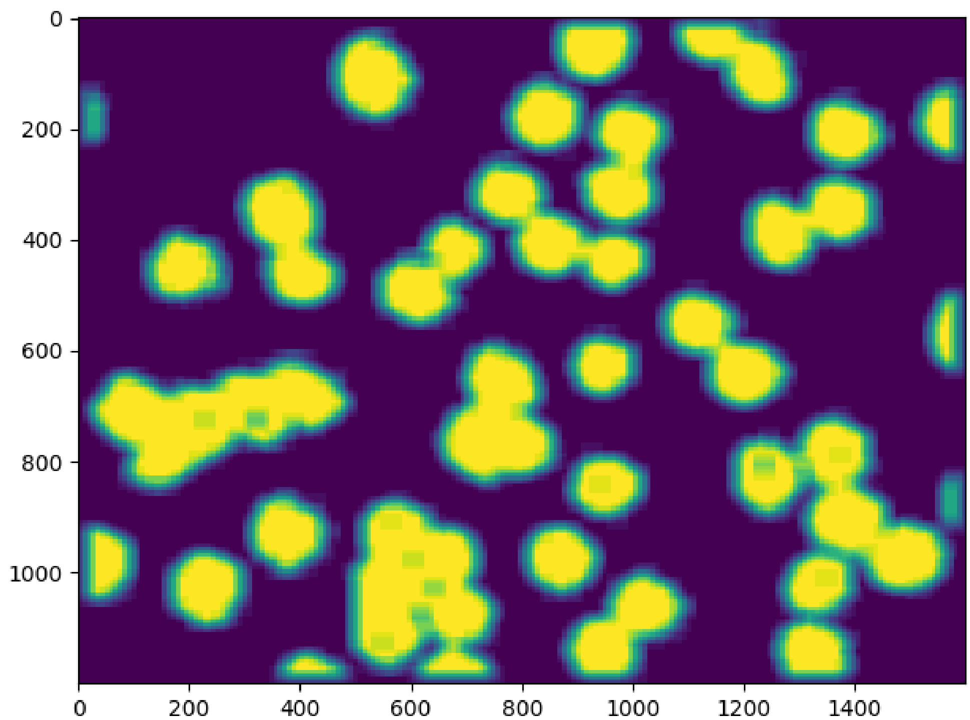


Figure 4

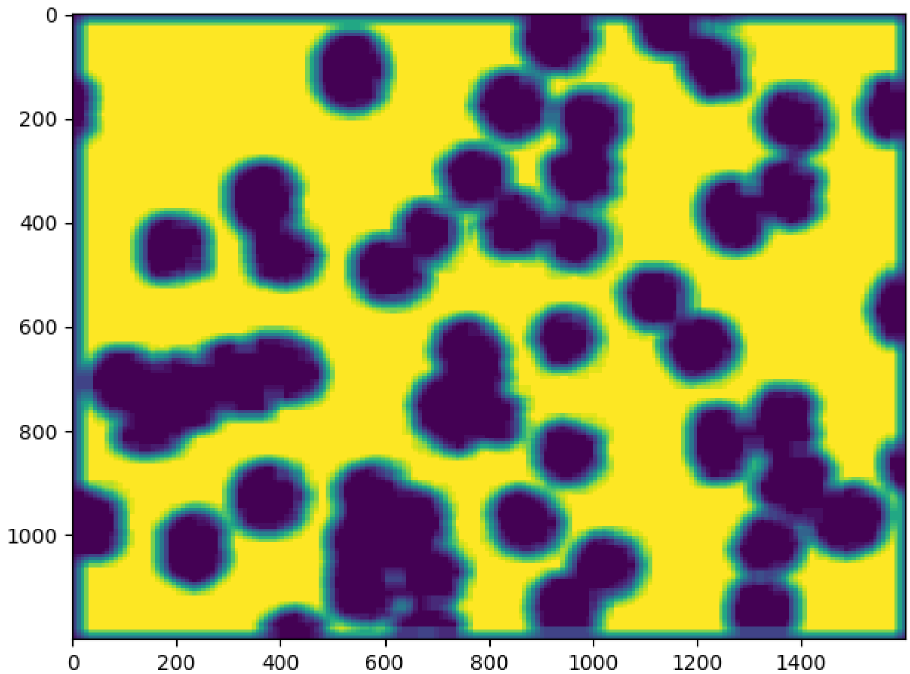


Figure 5

Basing on the finding so far, we can prove that we can use classifier to do object detection, masking, or multilabel classification. We need to make a better classifier using relevant data that includes a null class, the classifier need to have high F1-score. We can to built the Cropping and Mapping Algorithm with CUDA to improve image processing speed. We could also use multiple kernel sizes to better establish an image shape and location. We could also explore using kernels to trace boundaries directly. Finally, we need more computational power. All the Cropping – Predicting – Mapping algorithm code are original, The predicting model is from DenseNet and Exam 1 code.

References:

Horea Muresan, Mihai Oltean, Fruit recognition from images using deep learning, Acta Univ. Sapientiae, Informatica Vol. 10, Issue 1, pp. 26-42, 2018.

Oquab, M., Bottou, L., Laptev, I., & Sivic, J. (2015). Is object localization for free? - Weakly-supervised learning with convolutional neural networks. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi: 10.1109/cvpr.2015.7298668

Papadopoulos, D. P., Uijlings, J. R. R., Keller, F., & Ferrari, V. (2016). We Don't Need No Bounding-Boxes: Training Object Class Detectors Using Only Human Verification. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. doi: 10.1109/cvpr.2016.99

Ribera, J., Guera, D., Chen, Y., & Delp, E. J. (2019). Locating Objects Without Bounding Boxes. *Computer Vision and Pattern Recognition*. Retrieved from https://arxiv.org/abs/1806.07564

We used image set BBBC041v1, available from the Broad Bioimage Benchmark Collection [Ljosa et al., *Nature Methods*, 2012].