ECE324 Final Report

Arnaud Deza (1005336147)
Mark Lee (1002516538)
Elisa Ding (1004903795)

December 6, 2020

Word Count: 1957, Penalty = 0%

Permissions

Permission	Arnaud	Elisa	Mark
Permission to post video	Yes	Yes	Yes
Permission to post final report	Yes	Yes	Yes
Permission to post source code	Yes	Yes	Yes

1 Introduction

In recent years, advanced agricultural automation has increased productivity by reducing production costs and manual labor while increasing yield and quality [1]. However, many tasks are still completed manually and limitations exist in automating tasks on a large scale. As a team we have decided to create FRINN, a project whose goal is to identify the type of an input fruit and its state, either being ripe, unripe, or rotten. The main motivation and uses for this project are:

- Decrease food waste by identifying rotten fruit in a batch at an early stage to prevent spreading to other fruits during transportation or storage. Can be employed at large agricultural scales and also at local grocery stores when receiving new fruit stock.
- Increase efficiency at the agricultural level by automating the prediction of fruit states and therefore optimizing harvests
- Identifying fruit with a network can ease cashiers' jobs when processing fruit as currently cashiers need to remember serial codes to input into the register (minimizing human computer interactions)

The motivation behind the use of a neural network relies on the fact that currently humans are required to manually sift/look through fruit which can potentially decrease yield and increase labour. Classification of fruit ripeness with machine learning can be leveraged as fruit images are translationally invariant and mimic the wide range of possible orientations they can be in [2]. Additionally, some traditional methods using computer vision without machine learning are currently lacking and under performing and do not capitalize on the use of large fruit image databases [3-4].

2 Illustration/Figure

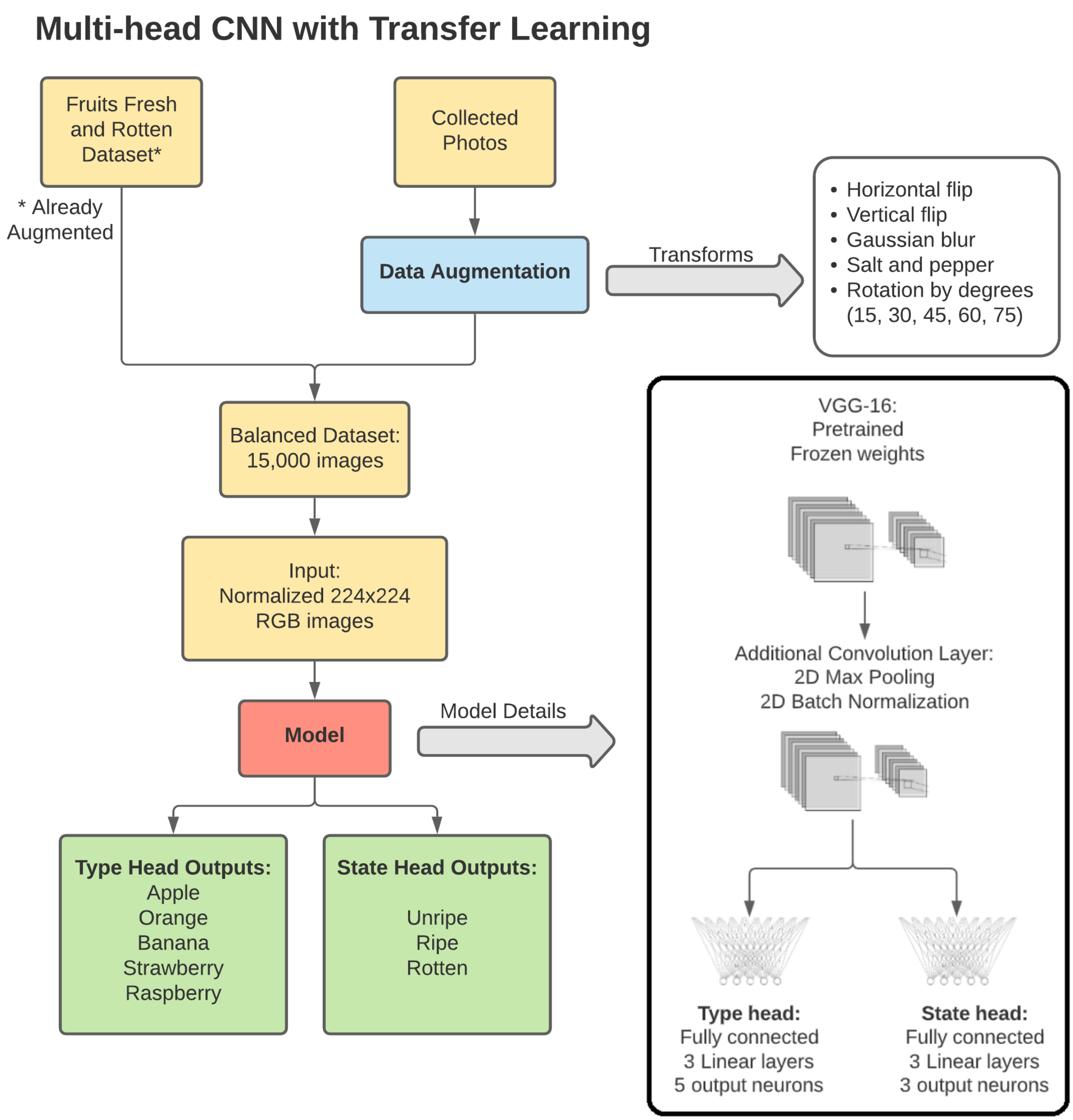
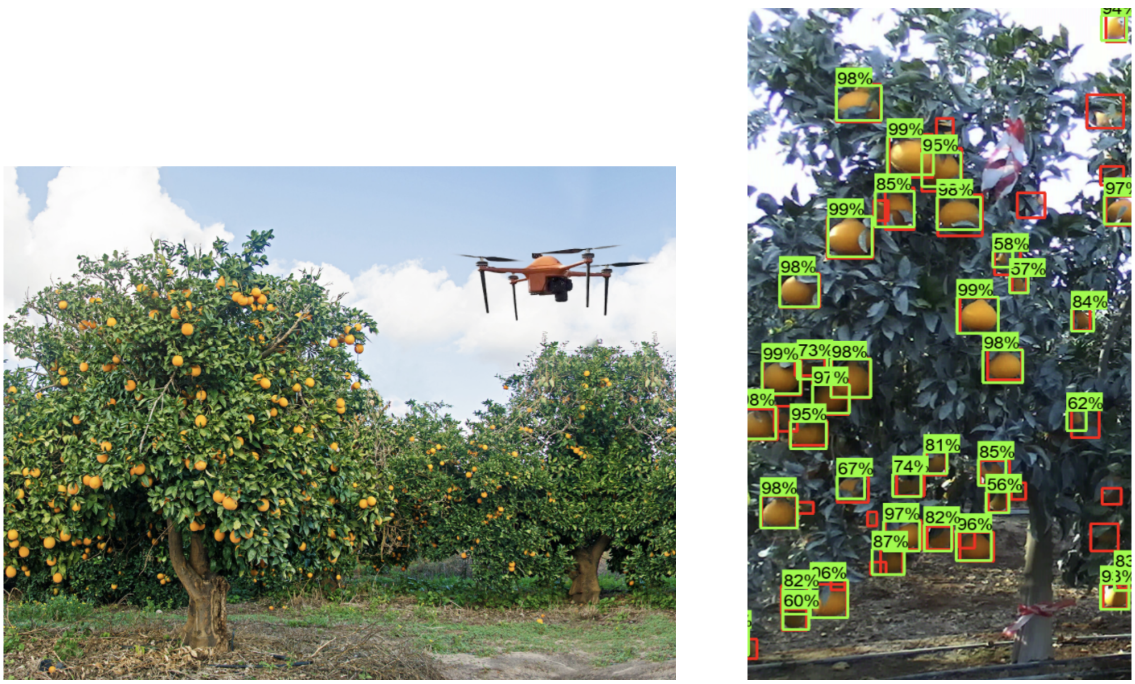


Figure 1: Illustration of project idea and model architecture

3 Background & Related Work

Before the recent machine learning (ML) advances, the main approach for fruit state identification was with OpenCV and image segmentation, which produced decent results but has limitations. Recently ML approaches for quality of fruit identification have been investigated by many researchers, resulting in papers regarding feature analysis, architecture choice, and reframing fruit feature detection as an image segmentation problem (i.e fruit vs. background) [5-7]. Upon further research, we have found that the use of a CNN model outperforms non ML computer vision approaches [8].

Fruit identification using ML has also been explored by agricultural companies. For example, SeeTree is a leading agri-tech company that uses deep learning to help robots optimize the harvesting process [9]. They implemented drones to automatically identify ripeness of oranges on trees as shown below, and overcame issues of small datasets and noise in photos.



(a) Data collection by SeeTree using drones [9] (b) Detection of ripeness using CNNs [9]

Figure 2: Fruit state identification by SeeTree [9]

4 Data and Data Processing

Our data comes from two sources: the Fruits Fresh and Rotten (FFR) dataset from Kaggle and self-collected images. Manually collected data involved screenshotting from stock image websites, as well as photos taken on our phones for later testing. The sources of all data are described in detail in Table 1.

Source	States (3 labels)	Fruits (5 labels)	Number of Images	Applied Data Augmentation
FFR (Kaggle) [10]	Ripe Rotten	Apples Bananas Oranges	12,772	No
Online	Unripe	Apples Bananas Oranges	450	Yes
Online	Ripe Rotten	Apples Bananas Oranges	600	Yes
Online	Ripe Rotten Unripe	Strawberry Raspberry	900	Yes
Phone	Ripe	Apples Bananas Strawberries	50	No

Table 1: Sources of Data

Data augmentation was performed using nine transforms from the Python imgaug library, shown in Figure 1 [11]. These transforms were chosen as they retain colour information, as opposed to transforms that adjust alter brightness and saturation, which can affect predictions of type and state. The augmentation resulted in approximately 27,000 usable images, but these were distributed unequally (Figure 3). To minimize pre-biasing, we equalized the dataset to contain 1000 images in each of the 15 classes (Figure 4). Despite removing a large percentage of examples, our model reached high final accuracies and we had the option of increasing the dataset if needed. We used a train/validation/test split of 64%/16%/20%.

Figure 3: Data Visualization after Augmentation

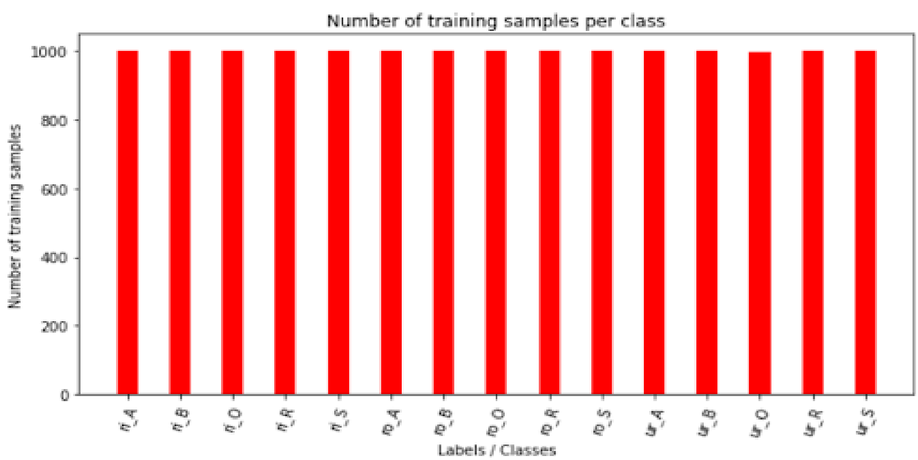


Figure 4: Data Visualization of Balanced Dataset

Below in Figure 5 we are also showing input examples to our model and data augmented images, as well as how it is labelled.

Figure 5: From left to right: original, vertical flip, rotation by 45 degrees, salt and pepper

5 Architecture

By using random hyperparameter search, we derived two final models, one multi-headed CNN model with transfer learning and one without transfer learning. Since the transfer learning outperformed the non-transfer learning model, only the transfer learning model will be discussed. The model takes 224 by 224 RGB image inputs which go through pretrained, frozen VGG-16 as well as one additional trainable convolutional layer with batch norm, pooling, and ReLU. Then it is passed onto two separate 3 layer MLP heads where the first head classifies ripeness and the second classifies the type of fruit. Each MLP has ReLU applied after each linear layer and Sigmoid applied at the last layer. The cross-entropy loss of the two MLP heads is summed before using an SGD optimizer to adjust the weights. The model was trained using a batch size of 64. The model-specific hyperparameters are seen in Table 2 and the number of trainable and non-trainable parameters are shown in Figure 6.

	Number of Kernels	Kernel Size	Learning Rate	# of Neurons in state head	# of Neurons in Type Head	# of Epochs
Multi Head with transfer learning	512	3	0.1 with reduce on plateau with patience 3	4096,4096,3	4096,4096,5	20

Table 2: Model Specific Hyperparameters

```
============================================
============
Total params: 256,208,456
Trainable params: 241,485,320
Non-trainable params: 14,723,136
--------------------------------------------
------------
Input size (MB): 0.57
Forward/backward pass size (MB): 322.80
Params size (MB): 977.36
Estimated Total Size (MB): 1300.74
--------------------------------------------
```

Figure 6: Model summary screenshot

6 Baseline Model

Our baseline model was composed of two random forests, one for state prediction and one for type prediction. The 224 by 224 RGB image input was normalized then converted into 64 mean value and 64 standard deviations by extracting them from 64 non-overlapping quadrants. The 128 numbers each representing a single image were used in a random forest algorithm that contains 100 trees and uses information entropy.

7 Quantitative Results

To quantitatively demonstrate the results of our models and project as a whole, we have decided to include the final test accuracy table, recall and precision, accuracy/loss plots, and the confusion matrices for state, type, and overall classification. The final test accuracy table shows our model's performance in comparison to other models. We included recall and precision for state and type to compare the impact of false negatives and false positives in each category. The accuracy plots show whether the model overfits during training. The loss plots show whether two MLP heads converged to the same loss during training rather than competing. The confusion matrices give insights on which classes performed best or worst.

Model	State	Type	Total
Random Forest Baseline	87.5%	82.2%	84.9%
Multi-head CNN (No transfer learning)	98.1%	95.9%	97.0%
Multi-head CNN (with transfer learning)	99.3%	99.4%	99.3%

Table 3: Final Test Accuracy Table

State	Precision	Recall
Ripe	0.98	0.99
Rotten	0.99	1.00
Unripe	1.00	1.00
Type	Precision	Recall
Apple	1.00	0.99
Banana	1.00	1.00
Orange	0.99	1.00
Raspberry	0.99	1.00
Strawberry	1.00	0.99

Table 4: State and Type Precision and Recall

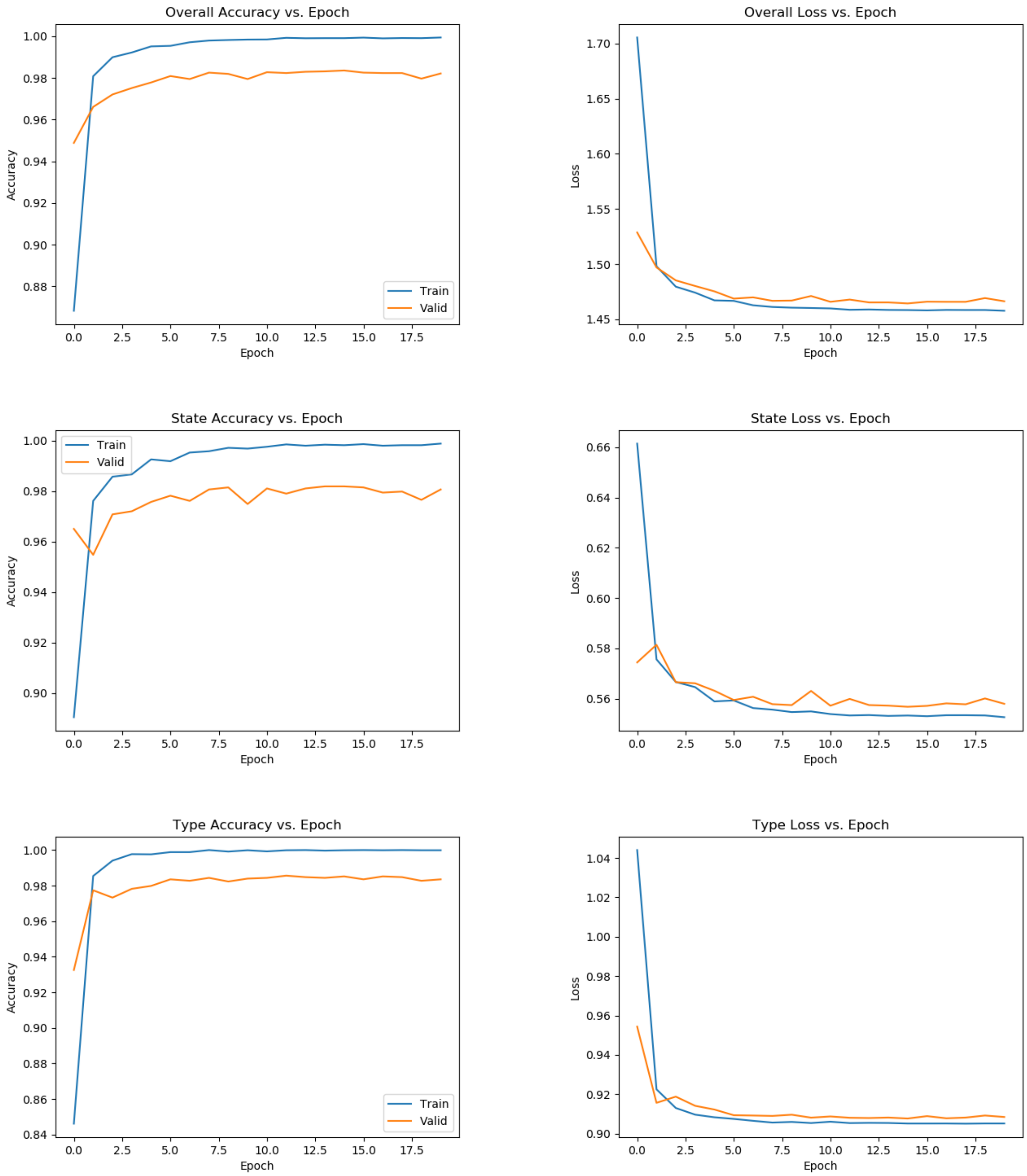
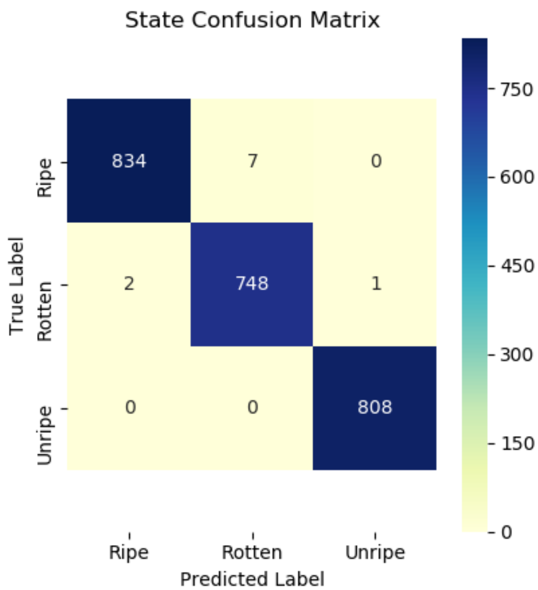
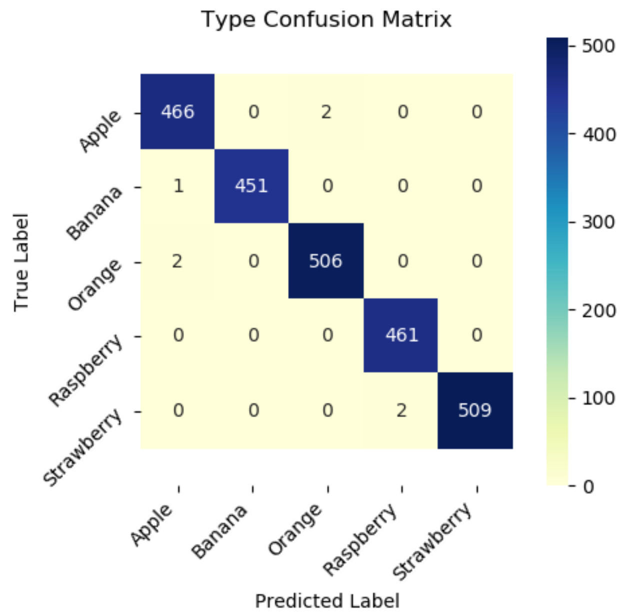


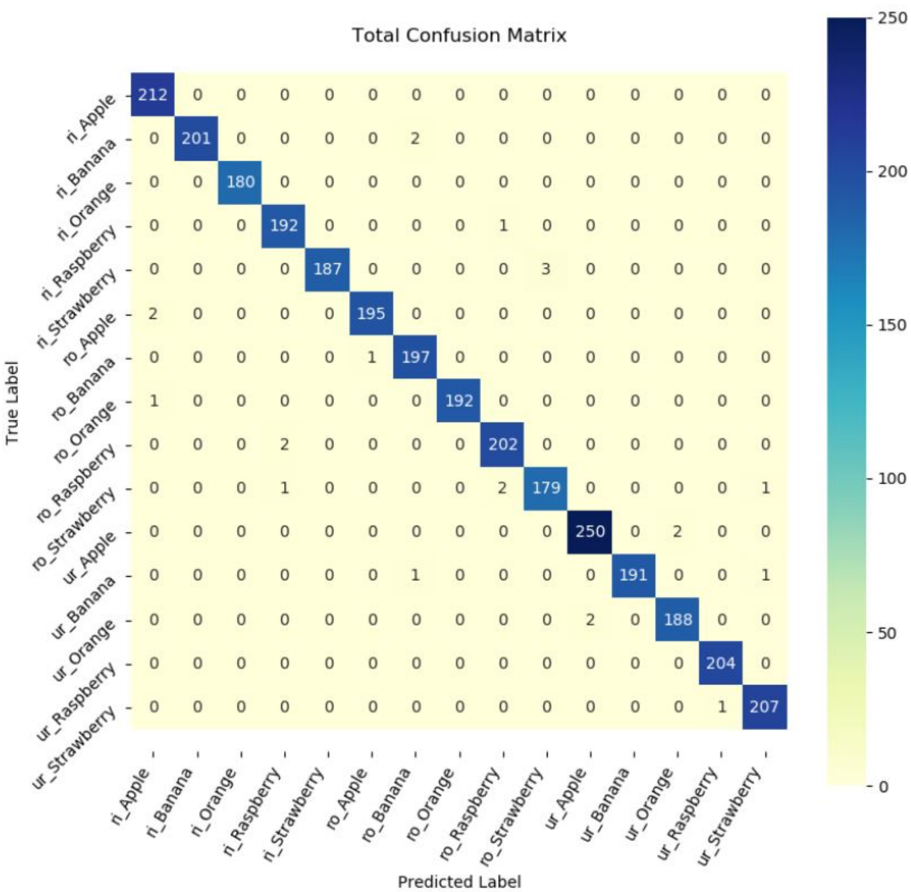
Figure 7: Training and Validation Accuracy and Loss Plots



(a)



(b)



(c)

Figure 8: Confusion Matrix

8 Qualitative Results

Our model was able to achieve a high accuracy on our dataset, but we did encounter a small percentage of misclassifications. We believe these occurred due to inherent fruit properties such as shape, color, and rotting stages and also prompted the use of specialized test sets to test some hypotheses.

Correct Classifications

Our model did have many correct classifications with some shown below. We believe our model's high accuracy is attributed to the shared feature space of state prediction and type prediction leveraged by shared convolutional layers.

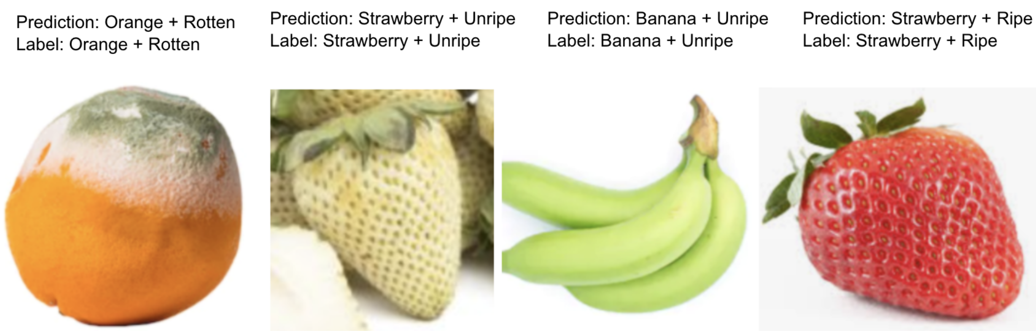


Figure 9: Sample of Correct Classifications

Misclassifications

- **Type prediction:** Given the above confusion matrices we know that the model struggles with differentiating apples and oranges. As shown in the misclassified images below, we think this is due to the inherent similar circular shape between apples and oranges compared to other fruits. This is even more apparent when differentiating unripe oranges vs. unripe apples which are not only both circular but also both green.

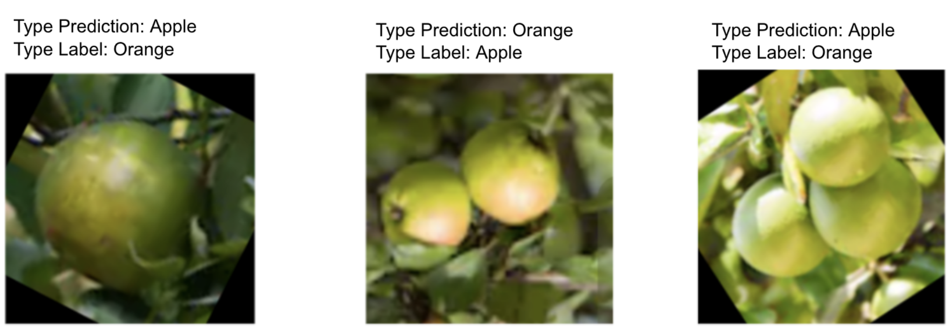


Figure 10: Sample of Misclassifications

- **State prediction:** We believe that the model is sometimes confused when trying to predict the rotten class as the stages of a fruit rotting are numerous and the rotting of some fruits are not as apparent as others. For example, some fruits rot from the inside whereas some grow white mold. The apple shown below on the left is rotten and was classified as ripe. It does not have white mold, so we believe the model is classifying it to be ripe due to the lack of white mold that is very apparent on a majority of training samples such as rotten oranges, raspberries, strawberries in Figure 11. This could be mitigated during the data collection phase by choosing more explicit rotten cases and finding rotten pictures for different fruit that have common features (i.e. rotting of apple at later stage where there is white mold).

Figure 11: Rotting of different fruits (right fruits have white mold, left fruits rot differently)

Specialized Test Sets

With the use of specialized test sets we were able to test hypotheses we had about our model. We were concerned that the model was being biased by the background of the fruits, as for example unripe fruits are green found on a green background (i.e still growing on a tree/bush). We created 3 specialized test sets (white background, green background, varied color background) to test our model and ultimately found that the background did not affect the model's prediction.

Figure 12: Samples from the specialized test sets

9 Discussion and Learnings

From the test accuracies shown in Qualitative Results, it is evident that our final model performed well in completing fruit classification and ripeness identification. A key factor in the high accuracies is likely the shared convolution layers in the multi-head architecture. When exploring different neural network architectures, we encountered the multi-head as a way to reduce parameters when solving related problems. This is because the common convolution and pooling layers produce a feature map usable by both heads.

The accuracy plots show that our model did not overfit while the loss plots show convergence in the overall loss. Since we are summing up two losses from each output head, the losses must not compete for the model to be successful. The confusion matrices show that the model performed well, but was confused between fruits with similar shapes or colors. The state confusion matrix shows that the model performs worse on classifying the rotten state. It is important to reduce false negatives in the rotten state as one rotten fruit could spoil the others. As seen in Table 4, recall and precision analysis on the rotten state shows the number of false negatives is lower than that of false positives. Despite the relative weakness of rotten, the model is still proficient in classifying states overall.

Observing our qualitative results, the small percentage of misclassifications are due to inherent properties of the fruit and a wide variety of conditions that are considered "rotten". Our incorrect hypothesis of background variety was resolved with specialized test sets, which we learned is an effective strategy to investigate potential problems. However, our model is still vulnerable to image properties, specifically in photos of fruits collected using our iPhones. As exemplified in Figure 15, images where the fruit is covered in shadows would be incorrectly predicted as rotten. This suggests real world data can be different from training data and should be accounted for, and neural networks should train on data similar to the expected input of stakeholders.

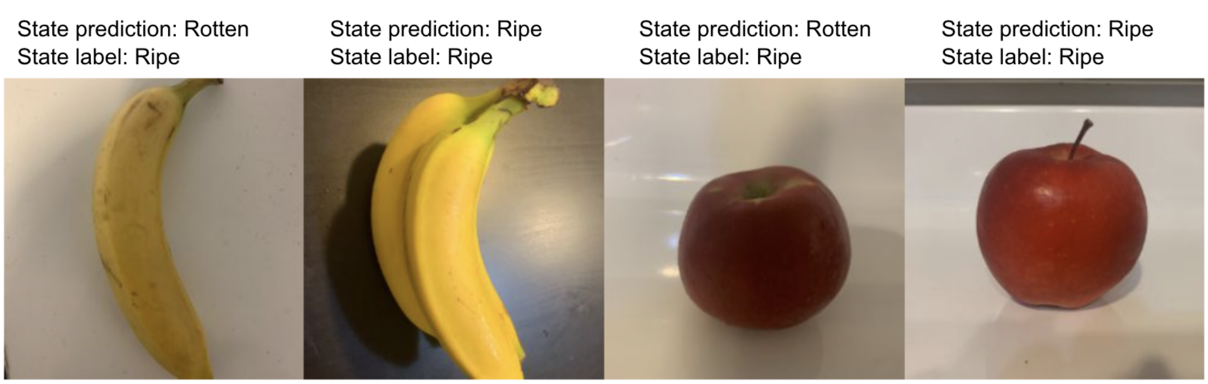


Figure 13: iPhone image examples and predictions demonstrating effect of shadow

Other changes we would implement if redoing the project are ensuring data is high quality before training and starting automated hyperparameter search earlier. To explain, most of the datasets from initial research were unusable due to repeated photos. We would also ensure the data is as similar to real-world data as possible and explore sources other than internet websites. Also, as training times are long it would have been more efficient to automate training over ranges of hyperparameters to allow for overnight training.

10 Ethical Framework

To explore the ethical implications of our project, we first identify possible stakeholders. This includes different institutions such as farms and grocery stores, as well as individuals such as employees, customers, and transport workers.

Our project is largely framed around beneficence and nonmaleficence, creating value by saving time and reducing food waste. The corresponding risks are loss of profit and wasted fruit due to misidentifications, but these have been minimized as our model reaches 98-99% accuracy. Customers and transport workers are minimally affected, but benefit from encountering less rotten fruit.

Regarding respect for autonomy, only stakeholders in management positions have control over the impact of the model. For example, they could employ a single worker to oversee mass-sorting by the model, or use the model as a final check after hand-sorting. However, employees at lower levels are at risk of losing jobs if automation is successful. Therefore, despite the benefit of faster fruit sorting and giving more choice to some stakeholders, other stakeholders may experience a decrease in autonomy.

Lastly, a major strength of the project is exemplifying justice by being equally useful for all stakeholders. For example, farms of all sizes (family, corporate) can use the model. Images are universally available if users have a camera, and labels can be easily translated to allow foreign language accessibility. Furthermore, the model can be modified for different fruits if appropriate datasets are obtained.

References

[1] Atkinson, Robert D. Robotics and the Future of Production and Work, Information Technology and Innovation Foundation, 15 Oct. 2019.

[2] "Deep learning implementation using convolutional neural network in mangosteen surface defect detection.", ResearchGate.

[3] Hussain, Israr, et al. "Fruit Recognition Dataset." zenodo.org/record/1310165.

[4] "FIDS30." Downloads/FIDS30 - Visual Cognitive Systems Laboratory,

[5] De-la-Torre, Miguel, et al. "Multivariate Analysis and Machine Learning for Ripeness Classification of Cape Gooseberry Fruits." MDPI, Multidisciplinary Digital Publishing Institute, 5 Dec. 2019.

[6] Ni, Xueping, et al. "Deep Learning Image Segmentation and Extraction of Blueberry Fruit Traits Associated with Harvestability and Yield." Nature News, Nature Publishing Group, 1 July 2020.

[7] "Fruit Recognition from Images Using Deep Learning.", ResearchGate.

[8] Kamilaris, A., and F. X. Prenafeta-Boldú. "A Review of the Use of Convolutional Neural Networks in Agriculture: The Journal of Agricultural Science." Cambridge Core, Cambridge University Press, 25 June 2018,

[9] "See Tree: AI Prototype for Ripe Fruit Detection." Ciklum.

[10] Kalluri, Sriram Reddy. "Fruits Fresh and Rotten for Classification." Kaggle, 24 Aug. 2018.

[11] Aleju, "aleju/imgaug," GitHub. [Online].