Fruit Image Classification: An Exploration of Deep Learning Models

Michigan State University – Applied Machine Learning

Jack Haas

Abstract:

As the title of this project suggests, its primary focus was the application of deep learning models to the problem of image classification on images of produce. A majority of the data was sourced from an online data store, though additional samples were created for the purpose of evaluating model generalizability. Closely examining model generalizability was one of the two major goals of the project, with the other being the rigorous exploration of a selection of deep learning architectures for image classification. The methodology behind choosing the selection of architectures was to evaluate how models of various complexity perform on the problem of fruit image classification. As such, models ranging from relatively simplistic artificial neural networks to state-of-the-art computer vision models were assessed. Because of the resource intensive nature of the project, a majority of model training and evaluation was performed with resources provided by Michigan State University's High Performance Computing Center.

Introduction

The broad problem under investigation in this report is the application of deep neural networks for image classification. The dataset used to explore this problem is Fruits-360. This dataset contains a large number of high-quality produce images taken under standardized conditions. This dataset represents an interesting application of

image classification as the problem of classifying fruits is something humans are quite good at. However, with a large variety of produce represented in the provided dataset, many of which differ in appearance in only subtle ways, the problem is not as trivial for machine learning algorithms. Additionally, as will be detailed shortly, the direct results of being able to accurately classify fruit images are less of a focus than the general machine learning insights.

There are two primary goals for this project. This first goal is to explore how deep neural network architecture plays a role in model performance on image classification. This will be explored by testing out five different architecture types ranging from a relatively simple artificial neural network to a state-of-the-art network designed for performance on computer vision tasks. The second goal is to explore how generalizable these models are. As a part of this process, each model will be trained on both raw images and images which have been significantly augmented in a number of ways. These models will then be evaluated on raw and augmented validation and testing images, allowing for the thorough analysis of how well each of the models has learned the high-level signals which lead to accurate produce classification.

To gain further insight into how well machine learning models trained on clean samples generalize to test samples created under slightly different conditions, part of the project also included gathering a number of produce images similar to those included in the Fruits-360 dataset. These images would then be prepared in the same way as the training images, and each of the models would be evaluated on this novel image dataset. This would allow for an understanding of how well a model was able to generalize beyond just the same standardized data collection practice used to prepare

the training images. To maintain some level of uniformity the novel data would be prepared with practices which would result in extremely similar end results to the training images. This would also allow for a deeper examination into the role image augmentation plays in making more generalizable models.

Answering, or at the very least exploring, these two main questions should lead to relevant results for anyone interested in performing image classification. This is because creating models which generalize well is an extremely important part of image classification. For example, if image augmentation is feasible for a problem and is able to significantly boost test prediction accuracy then its application would be extremely useful. Alternately, if an application does not require a complex model to achieve a high level of performance, then the additional resources required to train and maintain that model are likely not justified. As such, this project should yield valuable insights into the large-scale optimization problem which is machine learning for image classification.

Task Definition

The core undertaking of the project is the application of deep learning to image classification. As previously detailed, the specific image classification problem being addressed is the classification of produce images. As such, the primary inputs of the project are fruit images. There are two major kinds of fruit images used in the project. The first kind are those provided in the Fruits-360 dataset. These images are separated into training, testing, and validation sets. The second kind are novel images created as a means to test how well each of the models are able to generalize to samples of a

similar format to the training images, but which have been prepared under different conditions.

There were a number of technical inputs required to complete the project. Salient among them was Michigan State University's High Performance Computing Center. Using this system to take advantage of large amounts of computing resources greatly enhanced the project. From a software perspective, each of the models was created with TensorFlow, though functionality included in the Scikit-Learn library was used for various exploration and evaluation purposes.

The strict outputs of the project would be in the form of model performance metrics on each of the validation, testing, and novel image sets. However, in looser terms, the more interesting outputs of the project would be insights gained in comparing how produce class prediction performance varied across model type and training data type. As previously noted, it was the big-picture insights into how factors involved in the training of image classification models influence model generalizability which were the focus of the project.

Algorithm Definition

Five distinct model architectures were used in the project. The aim in selecting a range of deep neural network architectures was to adequately capture the prediction behavior which occurs on a spectrum of model complexity.

The simplest model used was a sequential neural network. This network was composed of two layers. The first was a dense layer of 300 nodes, what was selected as an arbitrary value thought to be wide enough to capture some of the complexity

required to make distinctions about the kinds of produce contained in each of the images. The second layer had a width of 128 nodes. This layer was kept consistent across each of the models.

The second model was a convolutional neural network with a single convolutional layer. This layer contained 16 nodes before being connected to the dense layer of 128 nodes. The next model was a convolutional neural network with two convolutional layers of widths 16 and 32, before the dense layer of 128 nodes. Fourth was a three convolutional layer convolutional neural network with convolutional layers of size 16, 32, and 64 before a dense layer of 128 nodes. The use of three convolutional neural networks was chosen as they have been a powerful breakthrough in performing computer vision tasks.

The final model architecture was a version of MobileNetV2. MobileNetV2 is a cutting-edge computer vision model developed by Google. MobileNetV2 was chosen out of a wide range of advanced models because of its comparatively small parameter count, intended to make it viable for use on mobile devices. Via a process called transfer learning, it was possible to train a version of the MobileNetV2 model specifically for the purpose of produce image classification. The training process was kept as consistent across all model types as possible to ensure comparable results were achieved. For a more extensive description of the MobileNetV2 model architecture please see the references included below.

Machine Learning Approach

Novel Data Creation

A core component of the model evaluation process was exploring how the models performed on produce images of the same format as the original dataset, but which were taken under different conditions. This process required a number of steps.

The first step was acquiring produce to take pictures of. Once the samples were purchased from a local grocery store, one strawberry, golden apple, white grape, orange, banana, and red apple were imaged. These images were then scaled down to 100 pixels by 100 pixels, the same size as the images included in the original dataset.

The next step, crucial for achieving fair performance comparisons with the original dataset, was to segment the images. While the novel fruit images were taken on a white background with high quality lighting, it was still possible to see some shadows in the images. Since the original images had been processed to make the backgrounds uniformly white, the same was done on the novel images. This was performed with a Python API for the service Remove.bg. This made it possible to quickly and accurately perform image segmentation to remove the image backgrounds. After segmentation was performed, all that was left was to fill the background with white and convert the images to the same file type as the original images.

Exploratory Data Analysis and Data Science Methods

The next step in the project was to explore the data and perform any changes which would be likely to improve the performance of the model. This was done in two

major ways. The first was to look at the class distributions. Ensuring that there was some level of balanced class distribution would be important for creating unbiased models. The second was to attempt to visualize how the classes differed in attributes, and specifically how the novel images fit relative to the original images.

The original dataset contained separate classes for multiple instances of the kind of produce. For example, it contained separate classes for ‘Red Apple 1’ and ‘Red Apple 2’, both of which were red apples. Since the goal of this project was to classify produce type rather than individual produce instances, the two classes were manually collapsed into a single class for each of the classes with multiple instances. After this was done the class distributions could be visualized.

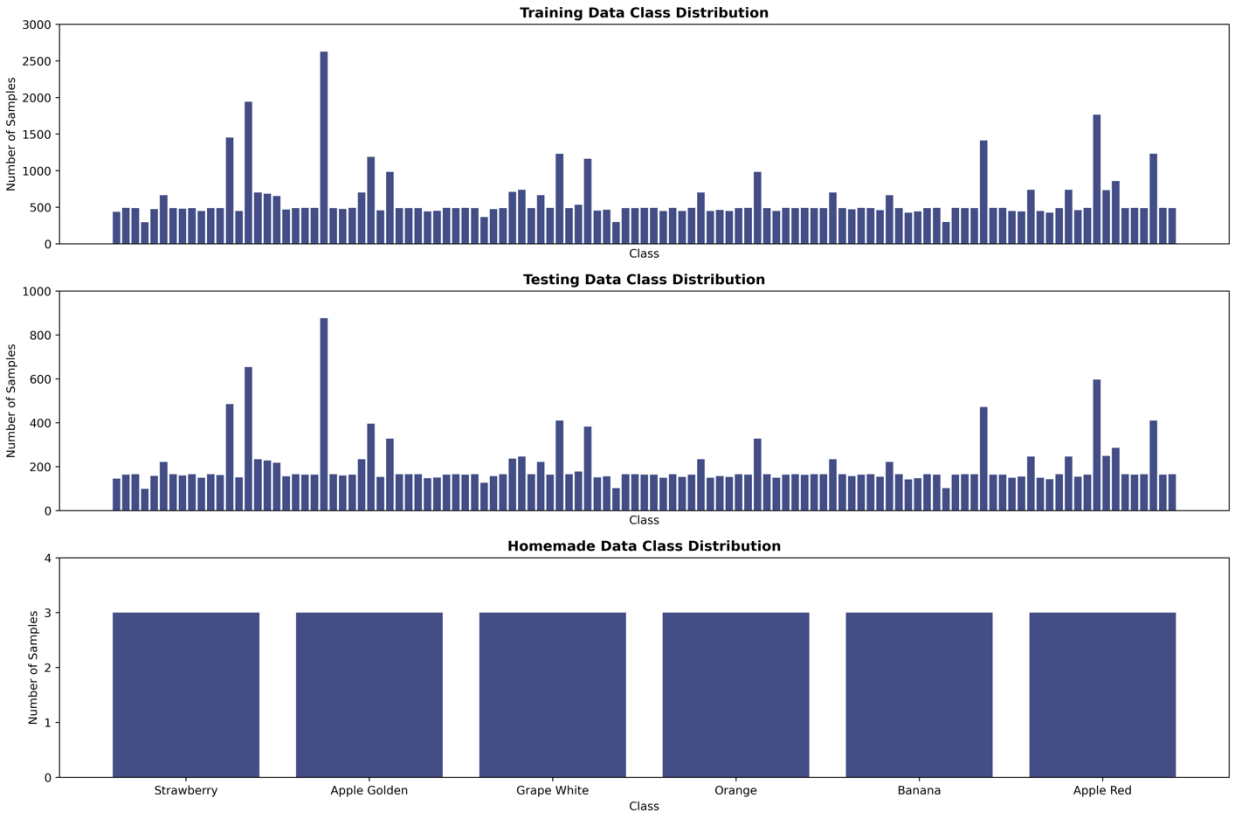

Figure 1 Training, Testing, and Novel Image Class Distributions

As the figure reveals, collapsing the overrepresented produce classes led to some classes having noticeably more sample images in the training and testing sets than other classes. However, since these classes represented many of the most common kinds of produce, such as apples and grapes, it seemed reasonable to keep this slight imbalance. Additionally, since the collapsed classes still represented a relatively small proportion of the total images, it was unlikely that any serious bias would be introduced by them.

Next, sample images from both the original dataset and the novel dataset were visualized to determine if there were any visual discrepancies existing between the two datasets.

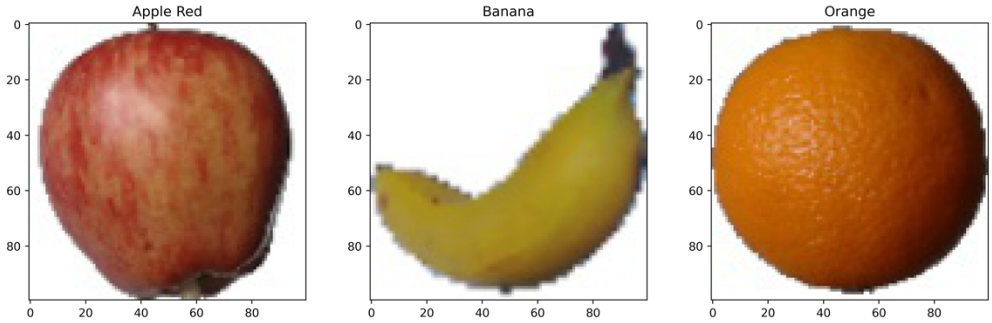

Figure 2 Original Dataset Images

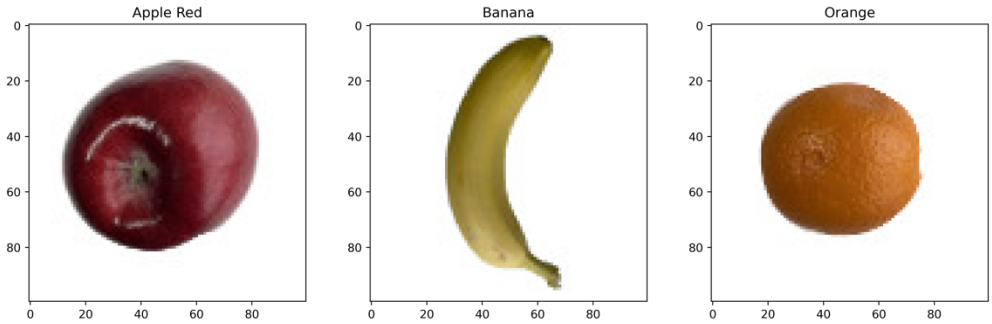

Figure 3 Novel Dataset Images

Visualizing the sample images was successful in revealing that the image preparation techniques had been successful. To the human eye, each of the fruit images from both the original and novel dataset would be easily recognizable. It was noticed that the images seemed to differ slightly in size and orientation. These aspects would be addressed below, where the influence of image augmentation would be explored.

To visualize how image features differed and how the features of the novel images compared to the features of the original images, a parallel coordinates plot was made with the principal components of the training dataset which captured 90% of the variance. It required 77 components to capture that threshold of variance, a surprisingly small amount compared with the original image size of 30,000. The same principal components created with the training dataset were then applied to the novel images, and both the training and novel images were shown against one another. For the sake of visualization each of the principal components are normalized between a value of 0 and 1.

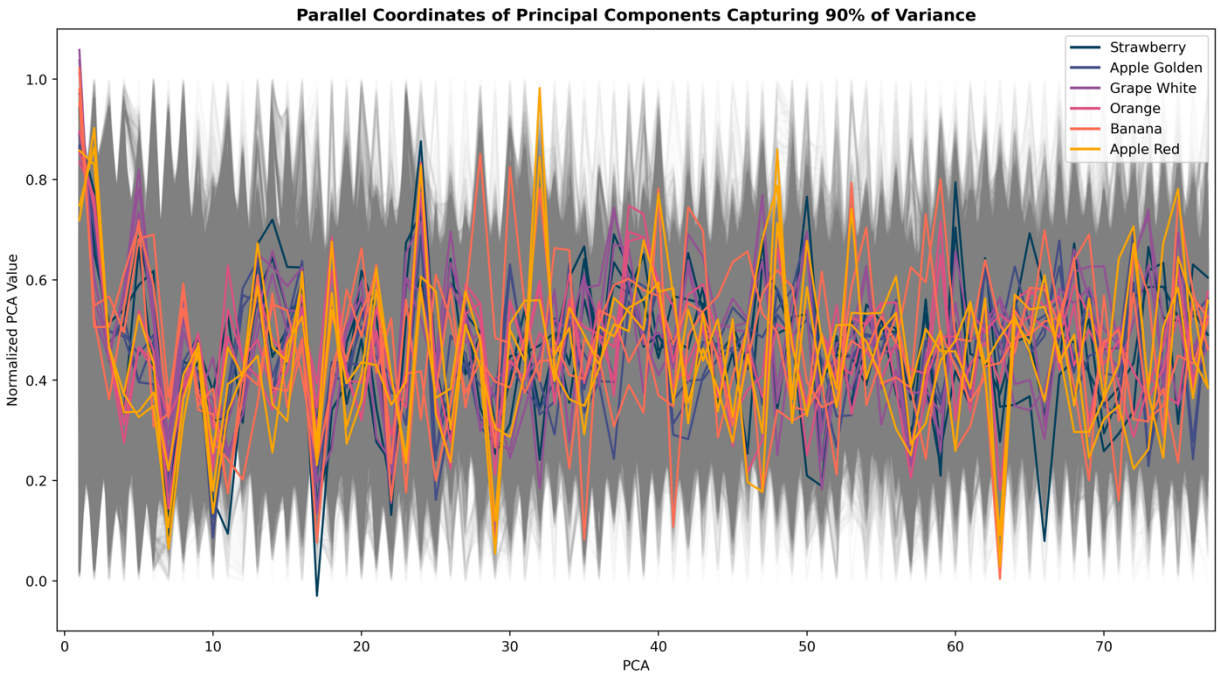

Figure 4 Principal Components Parallel Coordinates Plot

Since the novel images seemed to fit within the bounds of the training data it appeared that a classifier trained on the original dataset would be able to perform reasonably well on the novel images. Additionally, visualizing these principal components seemed to show that classes like Apple Red and Orange might be more easily classifiable as their components seemed better separated on multiple principal components than some of the other classes.

It should also be noted that each of the images were scaled to pixel values between either 0 and 1 or -1 and 1 depending on which architecture they were being used with. This was done to ensure high performance and was performed within the TensorFlow models themselves via rescaling layers prior to learning layers.

Machine Learning Approach

Each of the previously described model architectures were trained on two kinds of data. The first were raw images taken directly from the original dataset. The second were augmented images created with image augmentation functionality provided by TensorFlow. Image augmentation was performed to flip, rotate, and zoom the images randomly. The aim in performing image augmentation was to evaluate how well models were able to learn the general signals required to accurately perform classification on images of produce oriented in a variety of ways with respect to the imaging device.

The learning curves for each of the model architectures are shown below, with the left curves representing models trained on raw images and the right curves representing models trained on augmented images.

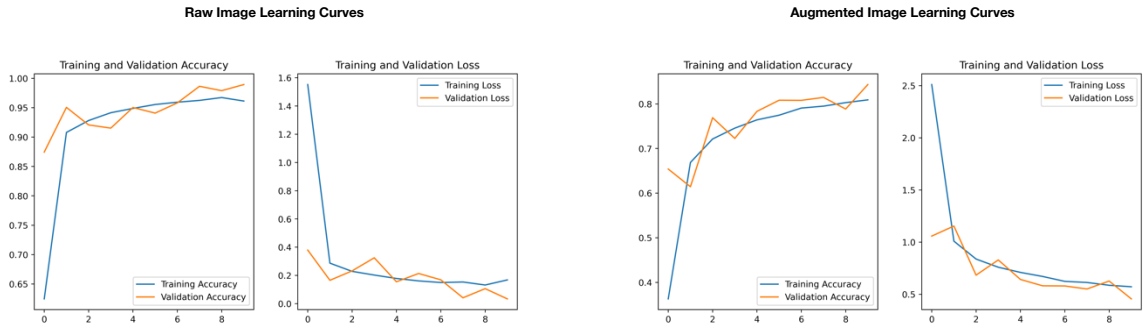

Figure 5 0 Convolutional Layer ANN Learning Curves

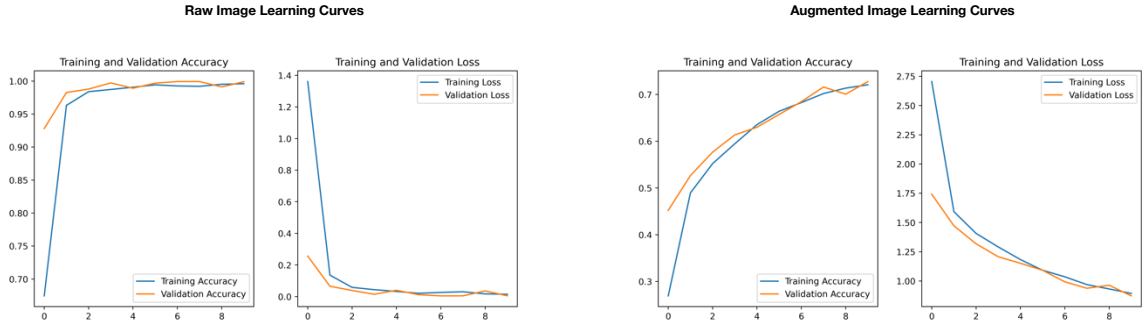

Figure 6 1 Convolutional Layer CNN Learning Curves

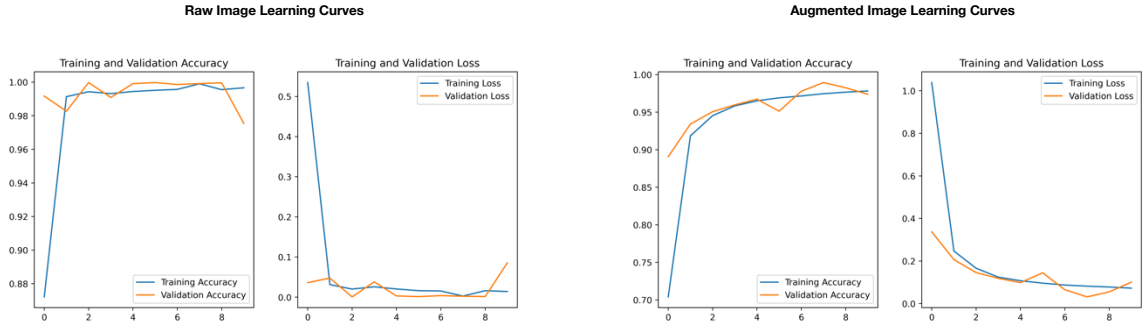

Figure 7 2 Convolutional Layer CNN Learning Curves

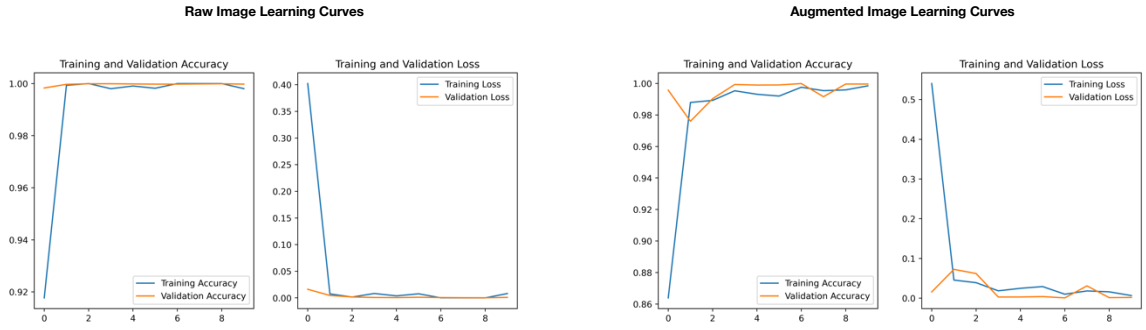

Figure 8 3 Convolutional Layers CNN Learning Curves

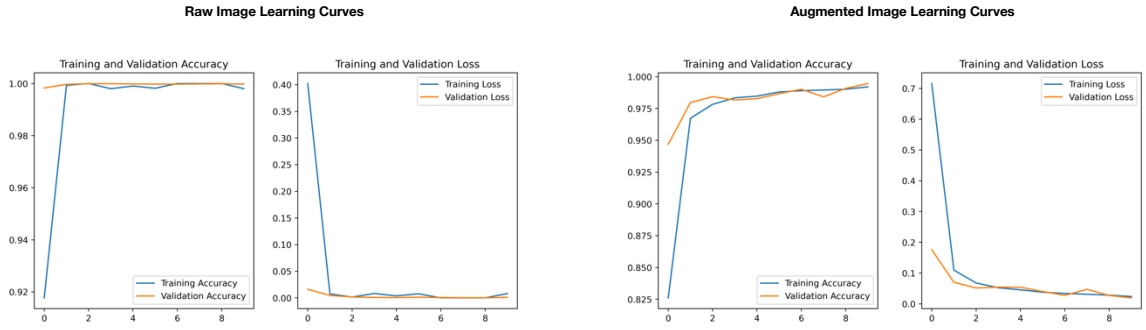

Figure 9 MobileNetV2 Learning Curves

The learning curves reveal some interesting relationships. For example, the model with no convolutional layers was able to perform quite well, with competitive validation accuracy against the simpler convolutional neural networks. Further, it appears the more sophisticated models were able to learn much quicker than the simpler models. This is seen in the extremely steep jumps in training and validation accuracy and falls in training and validation loss from the deepest convolutional neural network model and the MobileNetV2 model. Within a single epoch each were able to achieve validation accuracies over 98 percent when training on either the raw or augmented images, a level of performance not seen in any of the other models.

Another interesting point to note from the training curves is the difference in behavior between the model version trained on raw images and the model version trained on augmented images. As expected, learning the augmented images is a noticeably more difficult task as it requires a model to find more general image features. This is represented as learning curves which are both shallower and terminate at a lower level of performance than the learning curves of models trained on raw images. In the next section each version of the model will be evaluated to determine how successful image augmentation was in creating generalizable models.

Results

The table below represents the validation accuracy, testing accuracy, and novel accuracy for each version of model architecture over each form of training data.

Model Type	Training Data	Validation Accuracy - Raw	Test Accuracy - Raw	Novel Accuracy - Raw	Validation Accuracy - Augmented	Test Accuracy - Augmented	Novel Accuracy - Augmented
0 Conv Layers	Raw	0.989	0.912	0.000	0.281	0.262	0.000
0 Conv Layers	Augmented	0.860	0.787	0.056	0.845	0.769	0.111
CNN - 1 Conv Layer	Raw	0.999	0.927	0.000	0.328	0.315	0.056
CNN - 1 Conv Layer	Augmented	0.752	0.708	0.111	0.727	0.691	0.167
CNN - 2 Conv Layers	Raw	0.975	0.884	0.000	0.267	0.266	0.056
CNN - 2 Conv Layers	Augmented	0.969	0.941	0.000	0.974	0.942	0.000
CNN - 3 Conv Layers	Raw	0.999	0.949	0.167	0.290	0.278	0.056
CNN - 3 Conv Layers	Augmented	0.981	0.952	0.111	0.973	0.937	0.056
MobileNetV2	Raw	1.000	0.967	0.111	0.799	0.753	0.111
MobileNetV2	Augmented	0.986	0.940	0.167	0.993	0.956	0.167

Table 1 Model Accuracies

There are a number of interesting and relevant insights captured within these accuracies. The simplest of which being that more sophisticated models tend to perform better than the less sophisticated models. This can be seen clearest by comparing model architectures trained on the same training data. For example, the MobileNetV2 models dominate their corresponding 0 convolutional layer models across the board. Similarly, the 3 convolutional layer CNN dominates the 0 convolutional layer models in nearly every evaluation method, getting outperformed only in predicting augmented novel images, a task which proved to have the most stochastic behavior.

There were some interesting discrepancies in performance between the models on the simpler end of the spectrum. One surprise was the strong performance of the 0

convolutional layer ANN, and another was the outperformance of the raw imaged trained 2 convolutional layer CNN by the 1 convolutional layer CNN. To address the former, it appeared that selecting what was believed to be an arbitrarily large number of nodes within the 0 convolutional layer model allowed for a model which could learn the problem of produce classification unexpectedly well. In fact, the ANN was able to outperform the simplest CNN on many of the evaluation tasks involving the original dataset. That said, the simplest CNN seemed to be more generalizable than the ANN as it achieved greater performance on novel and augmented prediction tasks. It is expected that selecting a smaller number of nodes would lead to a decrease in performance, though it raises an interesting question about the point at which a sufficiently large ANN performs as well as a much smaller CNN for image classification.

Now to speak to the latter, the 2 convolutional layer model was outperformed by the 1 convolutional layer model when trained on raw images. This was especially surprising as the 2 convolutional layer model outperformed the 1 convolutional layer model when trained on augmented images. Thus, a more complex model performed comparatively better on a more complex problem than a simpler model performed on a simpler problem. This suggests that, to some extent, matching model complexity to the complexity of the problem can lead to benefits in model performance, especially with sufficiently simple models.

The second major set of insights concern how generalizable the models are. The main way this was evaluated was by looking at how models performed at predicting augmented testing and novel images, since these images best represent images which

a machine learning model would be likely to encounter ‘in the wild’. The scatter plot below seeks to explore the interplay between these classification tasks.

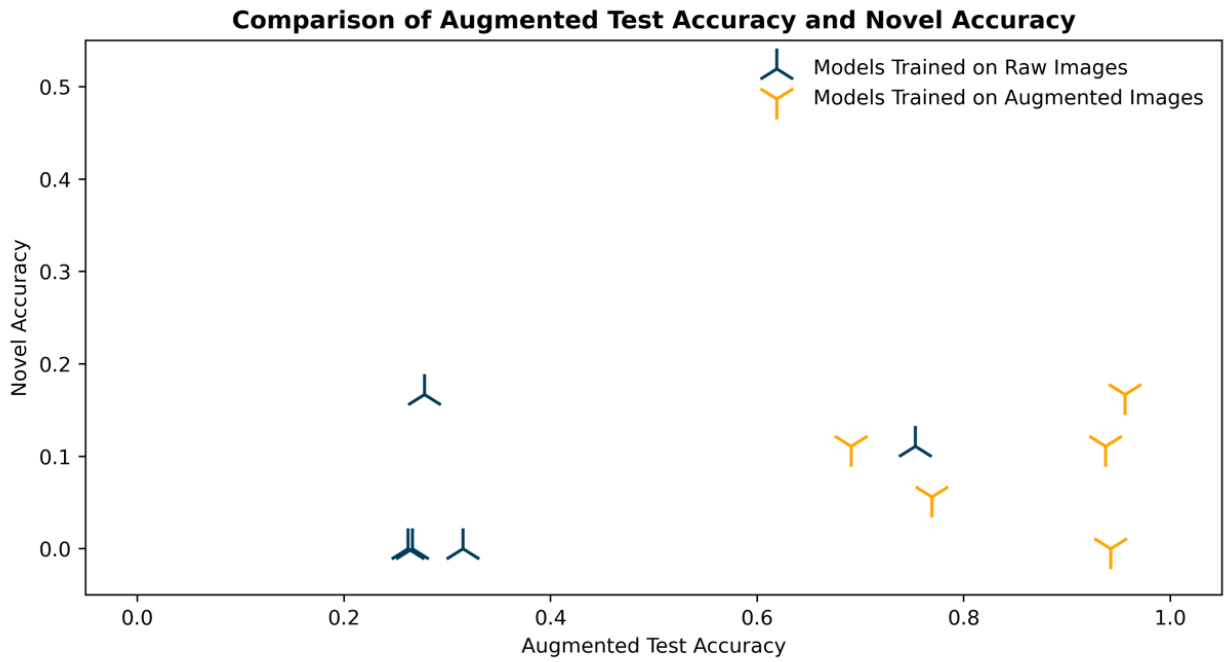

Figure 10 Novel Accuracy and Augmented Test Accuracy

There are a number of significant takeaways to be made about this visualization. The first, which has already been hinted at, is the somewhat stochastic behavior of how well the models were able to predict novel images. While it appears that models trained on augmented images have some advantage in performance, the outliers make it difficult to make such a claim concrete. Of particular note is the extreme outlier of the MobileNetV2 model trained with raw images predicting augmented test images with an accuracy of over 75%, placing it firmly outside the cluster of models trained on raw images. This speaks to power of the MobileNetV2 model architecture to perform image classification tasks.

The figure also depicts the importance of training models on a wide range of images and image orientations if one is looking to create as generalizable a model as possible. This notion is supported by the significantly higher augmented test accuracies of models trained on augmented images as compared with models trained on raw images. While this result is not a profound revelation, the augmented test accuracies increased an average of 48% from models trained on raw images to models trained on augmented images. This represents a massive gain in performance. It should be noted that these benefits in augmented test accuracies typically came at the expense of performance on raw validation accuracy. This behavior is well explained as image augmentation likely prevented models from learning the specific image features which may have been leading to the extremely high raw validation accuracies such as those seen for the more complex models.

Summary, Conclusions, and Outlook

To review the purpose of the project, the first main goal was to explore how a range of image classification models performed on a complex image classification task under a variety of circumstances. The second goal was then to rigorously evaluate those models to determine how generalizable they were. In both regards the project was quite successful.

By exploring a range of models, from a relatively simple neural network to a sophisticated model purpose-built for computer vision tasks, the project captured a slice of the image classification landscape. By then experimenting with these models through training and evaluation on both raw and augmented images, a deeper

understanding of the capabilities of each model was achieved. In this regard, the advantages of the more complex models were revealed. Even so, the simpler models performed surprisingly well, suggesting that making decisions about model choice should incorporate some notion of matching the complexity of the model to the complexity of the problem which it is being applied to.

As for model generalizability, there are a number of significant conclusions coming from the evaluation of each model version on augmented test and novel images. The first is that applying image augmentation to training images can lead to drastic improvements in model performance on images with differing orientations. In this same vein some models, specifically those with more complexity, are better able to handle images of varying orientation when trained on images of uniform orientation. This suggests that more complex image classification models are more generalizable to images differing in qualities from the images they were trained on. However, while performance on augmented test images seemed to behave in an explainable manner evaluation on novel images was not as insightful.

Though training on augmented images seemed to yield some gains in novel image prediction accuracy they were relatively minor. This suggests that even though the novel images were prepared in a nearly identical way to the training data and were just as easily identifiable to the human eye, there were some discrepancies which prevented the models from predicting the novel images well. Applying this point more broadly, it implies that machine learning models trained on relatively clean datasets do not generalize well to samples sourced under differing conditions, even if similar preprocessing techniques are applied. Additionally, while data manipulation

techniques, such as image augmentation, may improve the generalizability of models the gains may be relatively small. With this in mind, ensuring training data is representative of the kinds of samples a machine learning model will make predictions on seems to be the most effective way to ensure a high performing model.

The major shortcoming of the project was the lackluster predictability on novel images. If additional work was to be spent on this endeavor, exploring how to increase performance on novel images would likely be insightful. Applying additional image augmentation techniques to the training images, such as randomly varying image saturation or brightness, may lead to a greater understanding of why the techniques presented above failed to accurately predict on the novel images. Doing so would likely require performing augmentation before segmentation as altering the background color may introduce unwanted bias when making predictions. Taking another approach to address the same issue, one could systematically vary the conditions under which produce images are taken in an attempt to characterize the limitations of the models trained on the original data. Altering qualities like distance from camera, lighting, or orientation could provide a detailed account of the bounds of models trained on the Fruits-360 dataset.

Bibliography

Fruits 360. Kaggle. Retrieved from https://www.kaggle.com/datasets/moltean/fruits

Image classification | TensorFlow Core. TensorFlow. https://www.tensorflow.org/tutorials/images/classification

Transfer learning with a pretrained ConvNet | TensorFlow Core. TensorFlow. https://www.tensorflow.org/tutorials/images/transfer_learning

Team, K. Keras documentation: Keras Applications. Keras.io. https://keras.io/api/applications/

MobileNetV2: The Next Generation of On-Device Computer Vision Networks. (2018, April 3). Google AI Blog. https://ai.googleblog.com/2018/04/mobilenetv2-next-generation-of-on.html

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. ArXiv.org. https://arxiv.org/abs/1801.04381

Kaleido. Background Removal API. Remove.bg. Retrieved from https://www.remove.bg/api#api-changelog