

Grocery Tag

Sahil Sood

College of Professional Studies, Northeastern University

EAI6010: Applications of AI

Professor Abhijit Sanyal

Introduction

This paper addresses challenges related to overtraining in a computer vision classification task using a dataset of vegetable images, encompassing four classes: potato, onion, tomato, and noise (random vegetable market photos). The objective is to customize and fine-tune a pre-trained TensorFlow model to achieve optimal performance while avoiding overtraining.

In our approach, we opted for the ResNet50 model, a widely recognized convolutional neural network architecture renowned for its efficacy in image classification tasks. The model is configured to handle input images with a resolution of 256x256 pixels and three color channels (RGB). Notably, our transfer learning strategy, thereby excluding the top layers responsible for classification. This deliberate omission empowers us to tailor the model to our vegetable classification task by seamlessly integrating custom layers. Such a strategic initialization aligns with transfer learning principles, leveraging the pre-trained ResNet50 as a foundational framework for further adaptation to the specific nuances of our vegetable dataset.

Base Model

In the provided architecture, the ResNet50 base model is used as a feature extractor, followed by a global average pooling layer to reduce spatial dimensions. The final layer is a dense layer with softmax activation, facilitating multi-class classification with four output units. The inclusion of ResNet50 as the base model leverages its ability to learn hierarchical and discriminative features from images, enhancing the overall performance of the model in image classification tasks.

For model compilation, we used the Adam optimizer with a learning rate of $1e-4$, suitable for convergence. The chosen loss function is Categorical Crossentropy, ideal for multi-class classification. Our evaluation metrics include accuracy, precision, and recall, providing a comprehensive assessment of the model's performance in vegetable image classification.

Scenario of Overtraining

Overtraining, also known as overfitting, occurs when a model learns the training data too well, capturing noise and irrelevant patterns, resulting in poor generalization to new, unseen data. In our scenario, overtraining manifested when the model started achieving near-perfect accuracy on the training set. The accuracy on the validation set was lower however it was acceptable.

To mitigate overtraining, several training methods were employed:

Data Augmentation

A methodology for data augmentation is implemented using TensorFlow's Keras API. The augmentation sequence is constructed with three key operations: **RandomFlip** for both horizontal and vertical flips, **RandomRotation** to introduce random rotations within a specified range (up to 0.2 radians), and **RandomTranslation** allowing for random translations in both height and width dimensions (up to 20% of the image height and width).

Following the definition of the augmentation sequence, the methodology involves applying these augmentation operations to the images within the training dataset. This process is executed

through a loop iterating over the images and labels in the training dataset. Each image undergoes the defined augmentation transformations, with the augmented images replacing their original counterparts. By artificially increasing the diversity of the training dataset, data augmentation exposes the model to variations in input data during training. This strategy is designed to enhance the model's generalization performance across different scenarios. (Shorten & Khoshgoftaar, 2019).

Early Stopping

Monitoring the validation loss allowed us to halt training when the model's performance on the validation set plateaued, preventing overtraining. The decision to stop training was based on monitoring the validation loss. When the validation loss began to stabilize and even slightly increase, it was an indication that the model had reached its optimal performance. Further training would likely lead to overtraining and reduced generalization (Li et al., 2022). In our model, the if the validation loss plateaued and increased after 5 epochs the model will stop further training.

Regularized Model

In addition to the augmented data, the model was given several layers that helps us achieve generalization in our model to a larger extent. A global average pooling layer is applied to reduce spatial dimensions. Batch normalization is introduced to stabilize and expedite the training process, contributing to improved convergence. Additionally, dropout with a rate of 0.1 is included for regularization purposes, randomly deactivating 10% of input units during training to prevent overfitting. These are added during fine-tuning to randomly deactivate neurons during training, reducing co-dependency between neurons and preventing over-reliance on specific features. The final layer is a dense layer with softmax activation, facilitating multi-class classification with four output units. The model's enhancements, including batch normalization and dropout, are designed to improve stability during training and prevent overfitting. In comparison to the previously mentioned model, these additions aim to provide a more robust and effective architecture for image classification tasks.

Results

The table below presents the metrics for both the models, Model 1 refers to the baseline model which was trained on resnet model with 30 epochs and no regularisation.

	Loss	Accuracy	Precision	Recall
Model 1	7.9	0.63	0.68	0.57
Model 2	2.9	0.82	0.85	0.78

After applying the regularisation techniques mentioned above we created the model 2, trained on augmented training data. Early stopping at 12 epochs, dropout methods helped us achieve a significant improvement over Model 1.

Class Accuracy	Model 1	Model 2
Indian Market (Noise)	0.48	0.79
Onion	0.36	0.86
Potato	0.56	0.61
Tomato	1	0.97

The table above summarizes the classwise accuracy for both the models. As we can see there is a significant improvement over the initial models. Onion was the worst performing and tomatoes were the best performing class for model 1. Model 2 had the Tomato as best performing and potatoes as the worst performing class. Overall we can say that tomatoes have been the easiest class to generalize for both the models, whereas potatoes have been tougher to generalize leading to low accuracy metrics. Onion Class has benefitted significantly with the regularisation of model.

Recommendations for Improvement

Firstly, incorporating transfer learning with different pre-trained models, such as DenseNet or EfficientNet, may offer diverse feature representations and potentially improve generalization. Additionally, experimenting with different regularization techniques, such as L1 or L2 regularization, can provide alternative ways to prevent overfitting. Furthermore, fine-tuning hyperparameters, such as learning rate and batch size, can be explored for optimization.

To enhance predictions for other classes, it is advisable to focus on specific challenges faced by each class. For instance, for the Indian Market (Noise) class, which has shown a remarkable improvement in accuracy from 0.48 to 0.79, further data augmentation with diverse noise images and fine-tuning regularization parameters may contribute to better generalization. For potatoes, being the challenging class, collecting more diverse examples and experimenting with different model architectures could potentially improve performance.

In conclusion, this paper highlights the challenges of overtraining in a vegetable classification task and proposes effective methods to prevent overfitting. By carefully selecting and customizing a pre-trained Tensorflow model, monitoring training progress, and employing techniques like data augmentation and dropout, we can achieve a well-generalized model for vegetable classification.

References

Li, H., Rajbahadur, G.K., Lin, D., Bezemer, C., & Jiang, Z.M. (2024). Keeping Deep Learning Models in Check: A History-Based Approach to Mitigate Overfitting.

Shorten, C., Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J Big Data* 6, 60 (2019). <https://doi.org/10.1186/s40537-019-0197-0>