Meat Freshness Image Classification Project

Executive summary

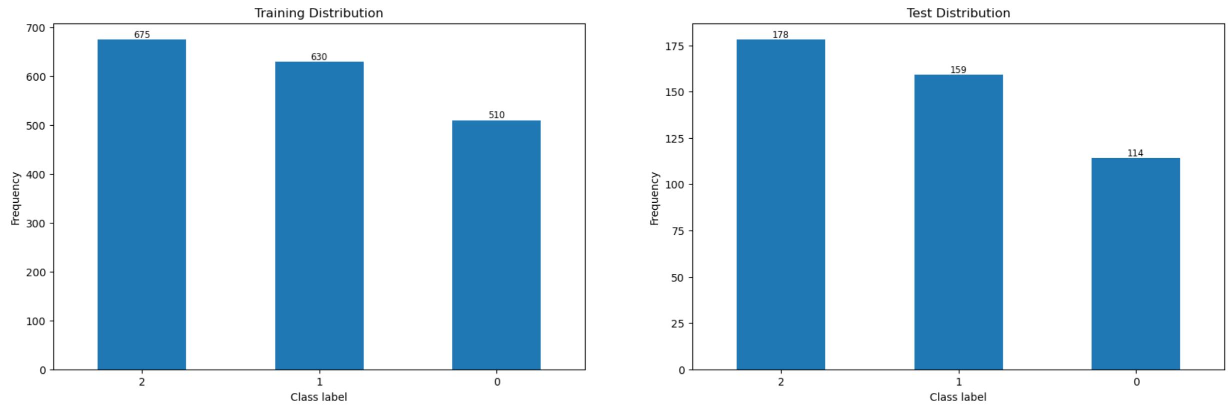
According to the United States Department of Agriculture, foodborne illnesses affect approximately 48 million people in the United States each year, with meat and poultry being among the top sources of contamination. Spoiled meat can be a burden for many food processing companies and retailers like Costco as food-borne illnesses and spoiled products can poison customer satisfaction and brand reputation. This project attempts to deal with that issue by creating a deep learning model to evaluate the freshness of meat based on its image. The dataset analyzed contains pictures of raw meat from a meat freshness image dataset by Roboflow on Kaggle. There are total of 2,266 images with the data split to 1,816 images for training and 452 images for testing. This project is a supervised problem since the data also contains a csv file with the labels indicating whether the images are fresh, half-fresh, or spoiled. Thus, the project will begin by exploring the data. Next is to create a baseline deep neural net model and baseline convolutional neural net. Using validation, the project will explore different methods to reduce overfitting and improve model accuracy. Finally using the best model, this project on the test set can accurately predict meat freshness for 90.24% of the images.

Data description

The dataset contains 2,266 pictures of raw meat from a meat freshness image dataset on Kaggle provided by Roboflow. Pre-processing was applied to each image before the project's analysis that consists of auto-orientation of pixel data (with EXIF-orientation stripping) and resizing to 416 x 416 (Stretch). No image augmentation techniques were applied. The training data has 1,816 images and the test data has 452. A csv is included with both datasets with the following labels to each image file: fresh, half-fresh, or spoiled. During the project's exploration each image shape is 128 x 128 and in full color. The images were divided by 255 to normalize each image. The image below contains some sample data.

FRESH FRESH HALF HALF SPOILED SPOILED

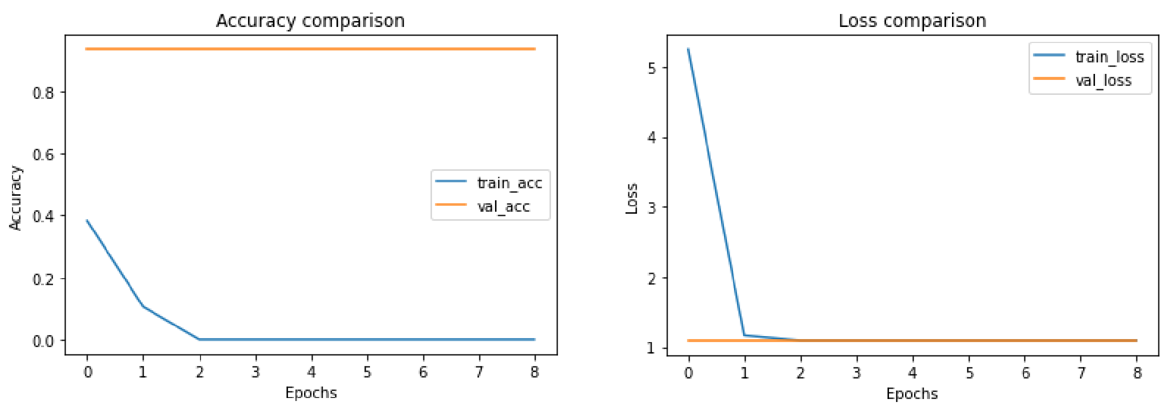
The raw csv file has binary labels for fresh, half-fresh, and spoiled. To prepare for model analysis, the labels have been reencoded to 0, 1, 2 for spoiled, half-fresh, and fresh. The distribution of the labels for each training and testing data are illustrated below.



Baseline Deep Neural Network

Next in the project, a baseline deep neural network (DNN) is constructed to determine if more complex models are necessary. This neural net architecture has an input layer with 49,152 (128 x 128 x 3) nodes, a hidden layer with 128 nodes and ReLU activation function, a hidden layer with 64 nodes and ReLU activation function, and an output layer with 3 nodes for the 3 classes and ReLU activation function. The optimizer is 'adam', the loss function is sparse categorical cross entropy loss, and the evaluation metric is 'accuracy'. Early stopping is used to monitor the validation accuracy with a patience equaling 8. Training stops early if the validation accuracy does not improve for 8 consecutive epochs to prevent overfitting. Model Checkpoint is also present to save the best model based on the validation loss.

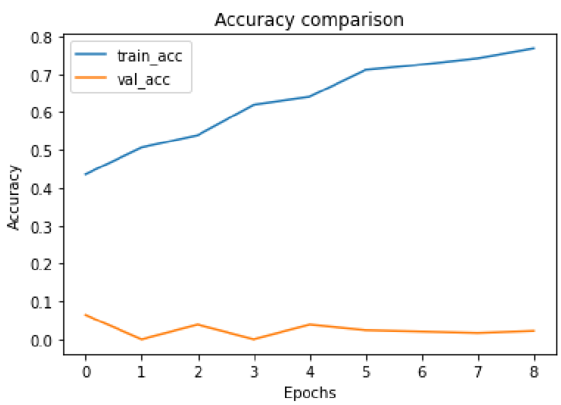
The model is trained on the training data with 100 epochs and a batch size of 500. 30% of the training data is used for validation during training. The results are as follows:

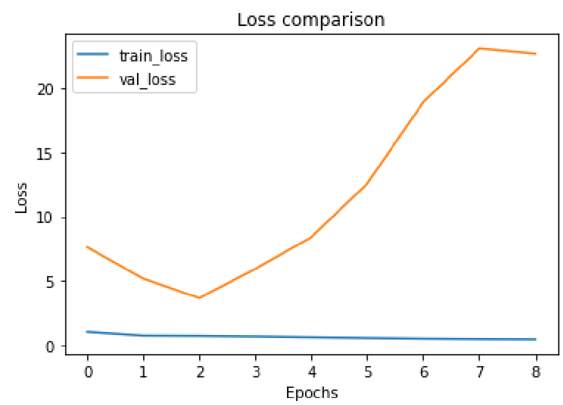


The results are clearly abnormal indicating that this model architecture is not adequate for this type of data analysis. Thus, further analysis will be performed with convolutional neural networks (CNN) that are much better suited for image data by capturing the spatial relationships between pixels.

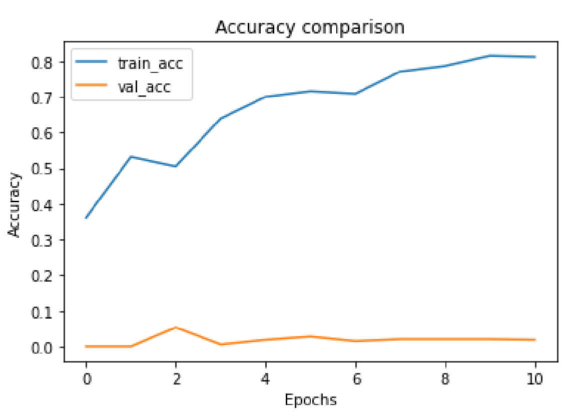
CNN Models

A generic CNN model is first constructed to check whether a CNN is the appropriate model. The architecture is as follows: convolutional layer with 16 filters of size 3x3 and ReLU activation function to an input image of size 128x128x3, max pooling layer with pool size of 2x2, convolutional layer with 32 filters of size 3x3 and ReLU activation function, max pooling layer with same pool size, convolutional layer with 64 filters of size 3x3 and ReLU activation function, max pooling layer with same pool size, flatten layer to flatten the feature maps into a 1D vector, fully conencted layer with 512 units and ReLU activation function, hidden layer with 3 units and softmax activation function. The parameters for compiling and training the model are the same as the baseline DNN. The results below appear more normal. However, the low validation accuracy and high validation loss display massive overfitting. Methods to reduce overfitting are applied to later model iterations.





To reduce overfitting, dropout, which randomly sets a fraction of the input units to 0 during training to prevent over-reliance on specific features and encourage generalization, is introduced to the CNN model. The dropout parameter is 0.5 and is inserted after every layer. Padding is also added so more spatial information is retained. However, the charts below still indicate massive overfitting so the issue may lie with the small sample size data.



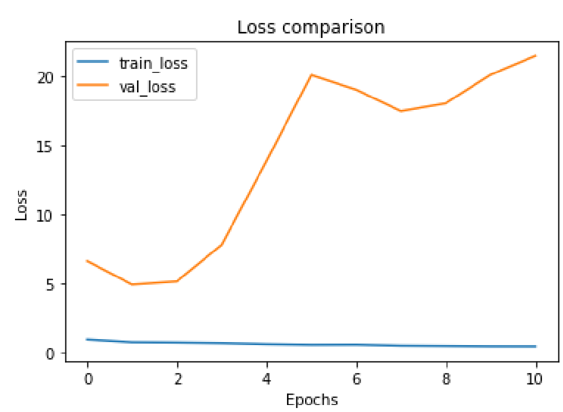
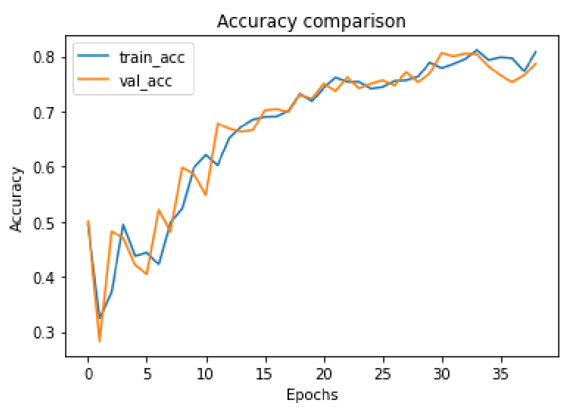
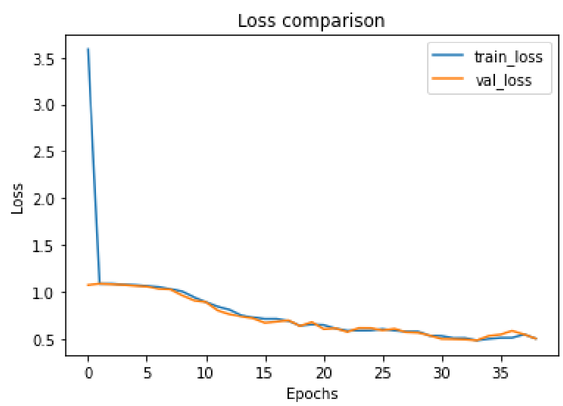


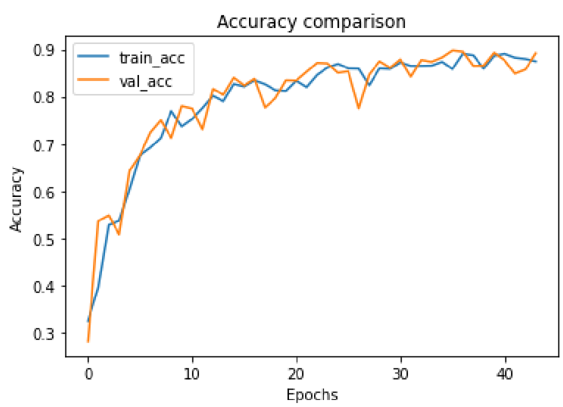
Image Augmentation

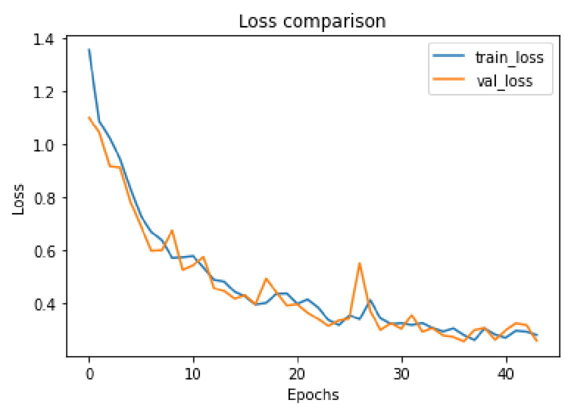
Since the CNN is overfitting due to the small sample size, image augmentation can reduce overfitting by artificially increasing the size of the training data. New images are created in this case by rotating the image randomly by an angle between -40 to 40 degrees, shifting the image randomly along the horizontal and vertical directions up to 20% of the image width/height, applying shear transformation randomly within a range of 0.2 radians, zooming the image randomly by up to 20%, flipping the image horizontally, and filling in any empty pixels created by the above transformations using the nearest pixel value. Because each image will be transformed 6 times, once for each parameter, there are 10,896 new images for training. Using the previous CNN, the results are significantly improved with an 80.61% validation accuracy.





To achieve better performance, dropout is reduced to 0.25 for the CNN and a hidden layer is added to the fully connected layer with 128 nodes and ReLU activation function. This CNN with the augmented images achieved an 89.81% validation accuracy and will be the final model.





Validation Best Results Table

Model	Epochs	Loss	Accuracy	Validation Loss	Validation Accuracy

Baseline DNN	9	1.17	10.55%	1.10	93.50%
Baseline CNN	9	1.04	43.54%	7.64	6.42%
CNN w/ dropout	11	0.69	50.47%	5.12	5.32%
CNN 1 after image augmentation	39	0.53	77.91%	0.50	80.61%
CNN 2 after image augmentation	44	0.31	85.84%	0.27	89.81%

Result summary

In short, the final more complex CNN model after image augmentation performed the best with 85.84% training accuracy and 89.81% validation accuracy. This model is then fitted on the entire training set. It achieved a 0.26 loss and 89.59% accuracy. Finally, it is evaluated on the testing set achieving a 0.22 loss and 90.24% accuracy.

The project began with scaling the images and then encoding them to 0, 1, and 2 for spoiled, half-fresh, and fresh. Then a DNN was constructed as a baseline, but the poor abnormal results led to the adoption of CNNs. However, the CNNs were significantly overfitting even with dropout. Image augmentation was applied and the CNNs were able to achieve high accuracy. This entire process highlights the necessity for having a large sample size for deep learning analysis and how image augmentation is a great remedy for small datasets containing images.

In the future, this project can be further improved by exploring other techniques to reduce overfitting while increasing model complexing to achieve higher accuracy. An example would be using a pre-trained network. Pre-trained networks have already learned useful features from large datasets and can be fine-tuned on smaller, specific datasets for a particular task. By reusing these pre-trained features, the network can learn faster and generalize better to new data, reducing the risk of overfitting. Additionally, collecting more image data would always make the analysis more robust.

The business applications for developing a CNN model to classify the freshness of meat samples includes better quality control, consumer satisfaction, and food safety. Meat processing companies can use a CNN model to automate the process of inspecting meat samples for freshness and quality. This can help reduce human error and ensure consistent quality across batches. Retailers and online marketplaces that sell meat can use a similar model to determine the freshness of meat products on their shelves or website quickly and accurately, reducing the risk of selling spoiled products and improving customer satisfaction. Deep learning models can be used to detect spoiled meat early, which can prevent food-borne illnesses and reduce the risk of product recalls. This can help protect the brand reputation of food processing companies and retailers. Overall, developing a CNN model to classify the freshness of meat samples can provide several benefits to businesses operating in the meat processing, retail, and food safety industries.