Convolution Neural Network to predict Hive Health from Bee Images

Aaditya Hemant 19BAI1098*,
*School of Computer Science and Engineering, VIT Chennai, Tamilnadu, India 600127
Email: aaditya.hemant2019@vitstudent.ac.in*

***Abstract*—Convolution neural networks are used widely in image processing and classification tasks. CNNs simplify the procedure of preparing images and generating a image based neural network model. We analysed a CNN model and convolution and pooling layers used to build it. We created our own model for predicting health of a bee hive from images of Bees and were able to reach 89% accuracy with it.**

1. Introduction

Convolution Neural networks are neural networks that are typically used in Image recognition and image processing. A CNN model takes an input image in the form of a matrix and establishes the spatial and temporal dependencies among to classify images easily and with high accuracies [1]. CNNs also require less feature engineering and are capable of learning the optimum filter by itself. Thus CNNs have become a dominant method of solving image classification and processing problems. We will take a look at creating a cifar10 CNN and then establish our own CNN for Bee Images dataset. Further we will improve our model by utilizing hyperparameter optimization.

2. Methodology

Convolution Neural Networks are an excellent method used for image related tasks. A CNN typically consists of convolution layers, pooling layers and dense layers aside from the input and output layer, The first layer in a CNN is the convolution layer. This layer is responsible for extracting critical features from the input image thereby reducing complexity and maintaining data integrity. It utilizes a filter along with an image matrix to construct a feature map as an output.

As the image is in the form of a matrix of high dimensionality a filter matrix is used to reduce dimensions. A set of pixels at a time are multiplied with the filter to generate the convoluted matrix. For an input of dimensions (h, w, d) and a filter of dimensions (f_h, f_w, d) an output matrix of size $(h - f_h + 1, w - f_w + 1, 1)$ is created. To ensure the input image and the filter are compatible we can either pad the input data by adding zeros or ignoring the data where filter does not match. [2]

Using convolution layer we can reduce the size of the input image. If necessary an additional pooling layer is

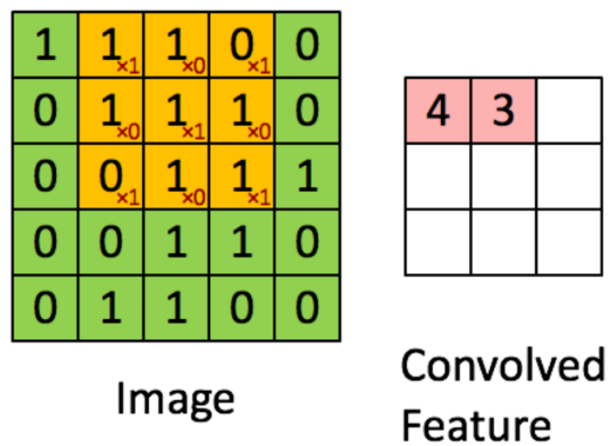

Figure 1. Reducing image dimension using filter
[2]

added to reduce the image size and the improve model time complexity. A pooling layer also uses a filter from which we take either the sum, average or maximum value. Once the image features have been successfully extracted and resized using convolution and pooling layers we can pass the data through a standard interconnected neural network.

3. Database - Bee Image Dataset

The Bee Image Dataset is a set of images and features of bees of various different species from different hives. The dataset contains preprocessed cropped images of bees extracted from videos of hives. Some of the images have additional information about hive health such as healthy, being robbed by other bees, infested by ants, hive missing a queen, Varao(parasitic mites) and beetle infestation. The dataset also mentions species from Italian, Russian and Carniolan. Unknown values are marked as -1. Data also mentions the zipcode of the hive in USA. For our experiment we will focus on the images and hive health.

4. Experiments

First we look at a CNN model for the cifar10 dataset. The baseline model with two convolution layers and 1 dense layer. Then we improve upon the model by implementing a Max Pooling layer. Using keras Image Data Generator function we can augment our input data by modifying existing image scale, rotation and cropping. The final model is created by incorporating both data augmentation and Max pooling layers.

As the data is not balanced in terms of classes we will balance the data by selecting from the annotated data and selecting their corresponding images. As the images of the bees may be from any direction we can use data augmentation by rotating and zooming into portions of the image. The model architecture consists of input convolution layer of 16 filters which takes in an input image matrix of dimensions 32,32 and 3 for each colour channel in the image. The output of this layer will have a shape of 32,32,16 which is then sent to a Max Pooling layer using a 2x2 filter. a second convolution layer then processes the data which is then flattened and passed to a Dense layer. Dense layer has 64 neurons and connects to the output layer to give a softmax prediction of the input image.

Using raytune we can optimize hyperparameters, for our experiment we want to test different activation functions, learning rate and the number of neurons in Dense layer and number of filters in convolution layers. A hyperparameter space is setup with the testing values. Activation functions are relu and tanh, Learning rate is taken in range from 0.001 to 0.1, Varied number of filters for convolution layer from 16,32,64 and neurons for dense layer from 8, 16, 64. Model training with raytune took 2 hours.

5. Conclusion

We successfully implemented a CNN model to predict the health status of a hive from images of bees. Using ray library we found best hyperparameters for the model to be a learning rate of 0.004386627196184818 using tanh activation function. Optimal model was created when number of filters in convolution network was 64 and number of neuron in the dense layer was 16. Our CNN model was able to predict with accuracy of 89%. Excluding varroa and beetle infested hives classes which only had an f1-score of 0.60 our model performed well.

References

[1] S. Saha, “A comprehensive guide to convolutional neural networks— the eli5 way.” [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

[2] R. Prabhu, “Understanding of convolutional neural network.” [Online]. Available: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148