

DEEP LEARNING APPROACH TO MUSHROOM SPECIES CLASSIFICATION

Yanni Alan Alevras

Student# 1009330706

yanni.alevras@mail.utoronto.ca

Nicholas Biancolin

Student# 1009197726

n.biancolin@mail.utoronto.ca

Eric Liu

Student# 1009098450

ey.liu@mail.utoronto.ca

Jason Ruixuan Zhang

Student# 1008997631

jasonrx.zhang@mail.utoronto.ca

1 INTRODUCTION

Fungi identification is an increasingly critical task, with implications in food security, industrial use, conservation efforts, and biosafety. However, visual and image classification of fungi is a difficult task due to the wide variety of species (?).

Nearly 87% of Ontario's land is Crown Land, filled with expansive forests and a diverse array of plant species. Among these, mushrooms stand out as both common and challenging to identify, due to their wide variety and subtle differences between species. Thus, identification is particularly important because some species are edible and others are highly poisonous, which could pose a risk to foragers and nature enthusiasts.

Our project developed a deep learning model based on a convolutional architecture to accurately identify macrofungi (fungi with large bodies) based on their genus. Deep learning formed a suitable choice for this task, as it constitutes a powerful and accurate method for image recognition and classification tasks, including in ecological settings (?).

We used the MIND.Funga dataset described in ?, which has approximately 17 000 images of more than 500 species of fungi spanning 216 genera. This dataset is well-suited for our project, as it is built primarily for use in deep classification models. Images are also curated to be of a high quality and are labelled by species.

Our code can be found at this GitHub repository. It contains Jupyter notebooks with pre-processing, baseline, main model, and evaluation code.

2 BACKGROUND & RELATED WORK

Previous applications of fungi identification models have prioritized food safety and satisfaction. In Bangladesh, a country with a large mushroom production, a farming method was developed by ? using an ensemble machine learning classifier (including SVMs, k-nearest neighbours, random forests, and naïve Bayes) to predict which mushroom was harvested. This system was designed to remove toxic species that may have grown in the same area. ? created an OpenCV-based algorithm to identify disease, discolouration, freshness, as well as other factors contributing to the commercial quality of a white button mushroom.

Keeping with the theme of food quality, in Taiwan, ? developed a deep convolutional image recognition model to provide an estimate of mushroom growth progress, similar to the underlying architecture of our model. In the Chinese province of Yunnan, ? created a wild mushroom identification system using a CNN to identify edible and medicinal mushrooms due to increasing popular interest in mushrooms.

Other field work includes smartphone applications for recreational use, like ShroomID, which details mushroom species, while providing a heatmap of its location and seasonality. This tool provides useful information about a mushroom after it has been identified using a classification model (?).

3 DATA PROCESSING

The original dataset contains images of 509 classes. Many species have a small number of images associated with them, insufficient for training a deep learning model. To mitigate this, we grouped images into larger "buckets" based on their genus to reduce the number of classes to 216 classes, which increases the number of images per class. Genera form an ideal way to group images together, as species of the same genus share physical characteristics and the same root name (first word in the species name) (?).

Once this combination is done, we take the top 10 genera with the most images to form an intermediate dataset of 6131 images. We then randomly split the dataset into training, validation, and testing sets with a 75%, 15%, and 10% split, respectively. This split ensures the model has enough data to train on, while ensuring validation and testing better reflects its performance.

Finally, we apply data augmentation techniques described by ? solely to the training set, to avoid poisoning the validation and testing sets. An example of this applied to one image is described in Figure ?? . We applied flips, rotations (90, 180, 270 degrees), Gaussian noise, and random erasing (of black rectangular regions). These manipulations were chosen as they would not alter the image's quality or classification. As mushrooms are identifiable based on visual traits, preserving key details is necessary to distinguish between genera. As our model is agnostic to colour, colour space transformations were not applied.

Our final dataset contains 16 962 training images, 3379 validation images, and 2324 testing images. To generate a usable set, we save a comma-separated value (CSV) file for each set, containing the image path and corresponding label. We also serialize the dataset into a binary Pickle object.

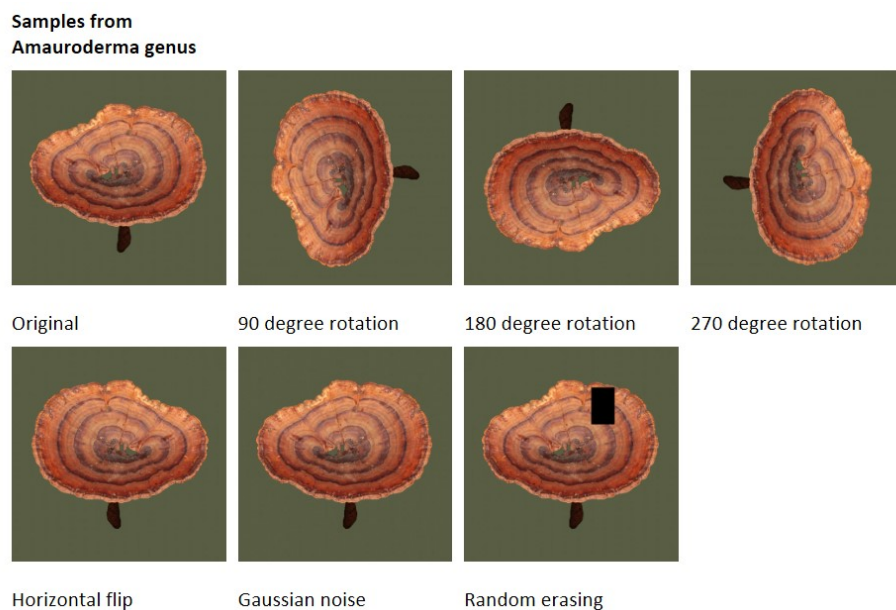


Figure 1: Data augmentation techniques applied to a sample image

4 ARCHITECTURE

Our primary model relies on two distinct sections: a non-tunable transfer learning section, and a tunable convolutional and fully-connected section. Our data pre-processing and model training pipeline is described in Figure ??.

We also use cross entropy loss, as it is a standard loss function for multi-class classification tasks. We also use a stochastic gradient optimizer, with a momentum of 0.8, weight decay of 0.001, and learning rate of 0.001. Our training flow uses 100 epochs with a batch size of 64.

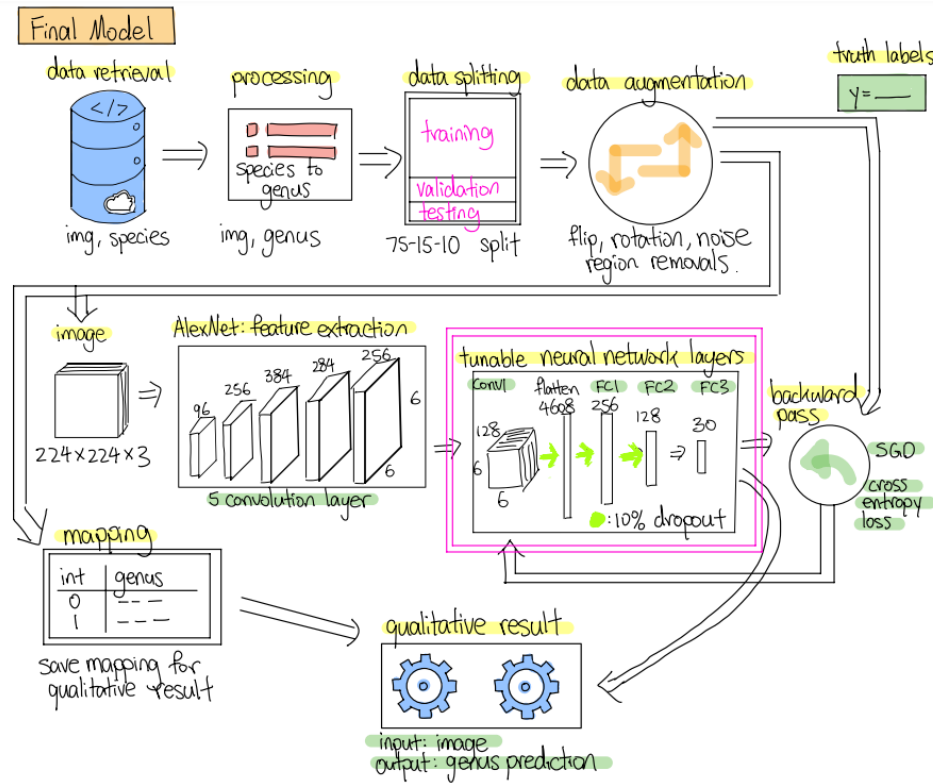


Figure 2: Primary model processing and training pipeline

4.1 NON-TUNABLE SECTION

We use AlexNet for high-level feature extraction, sourced from PyTorch’s pre-trained models. AlexNet is a well-known convolutional architecture for image classification tasks (?). Since the model needs to differentiate between mushrooms with similar appearances, AlexNet excels in extracting the fine features that set them apart.

Its architecture is described in Figure ??. It processes a $3 \times 224 \times 224$ input image, and the feature extraction outputs a $256 \times 6 \times 6$ tensor. There are five convolutional layers and three pooling layers (?).

AlexNet Architecture

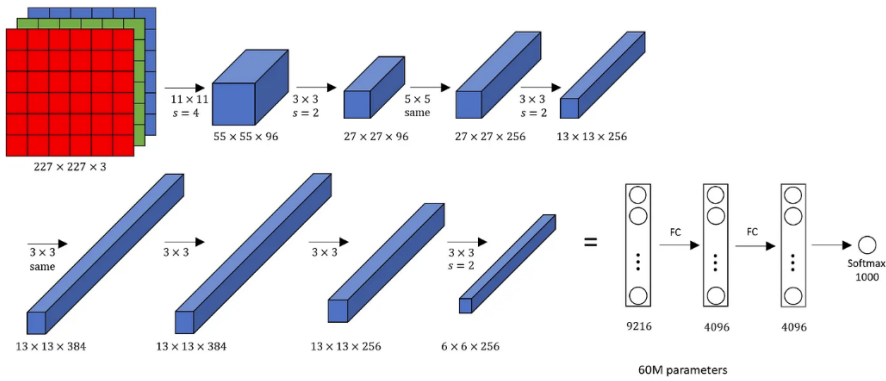


Figure 3: AlexNet architecture

4.2 TUNABLE SECTION

To make the model specific to our team's project, we add an additional convolutional layer, which outputs a $128 \times 6 \times 6$ tensor. After this layer, the output is flattened and passed through three fully-connected layers with ReLU activation functions. The fully-connected layers have 256, 128, and 10 neurons, respectively, which matches the number of output classes in our dataset.

In total, there are $5 + 3$ layers in the non-tunable section, and $1 + 3$ layers in the tunable section, making our model architecture 12 layers in total.

5 BASELINE MODEL

Our baseline model is a random forests classifier trained with scikit-learn on the same dataset as our deep learning model. Decision tree classifiers, like random forests, are generally well-suited for multiclass problems (?). We follow similar processing steps as the primary model, including a train-test split of 75%-25%, feature extraction, training, and evaluation. We do not apply data augmentation, as this benefits CNNs more.

We apply feature extraction on grayscale images with histogram of oriented gradients (HOG), an edge extraction descriptor algorithm (?) and computed with scikit-image. As random forests does not automatically perform feature extraction (compared to deep convolutional networks), this step is done to provide the model with a more effective set of features to learn from. We compute HOG features on both the training and testing set with default parameters, in line with the process outlined by ?. We also use default hyperparameters for training, with the exception of the number of estimators, set to 500, which suggests is an optimal number of trees for increasing accuracy.

Our final baseline model achieves an accuracy of 61%. We achieve an average precision of 0.7, recall of 0.51, and F1 score of 0.54.

6 QUANTITATIVE RESULTS

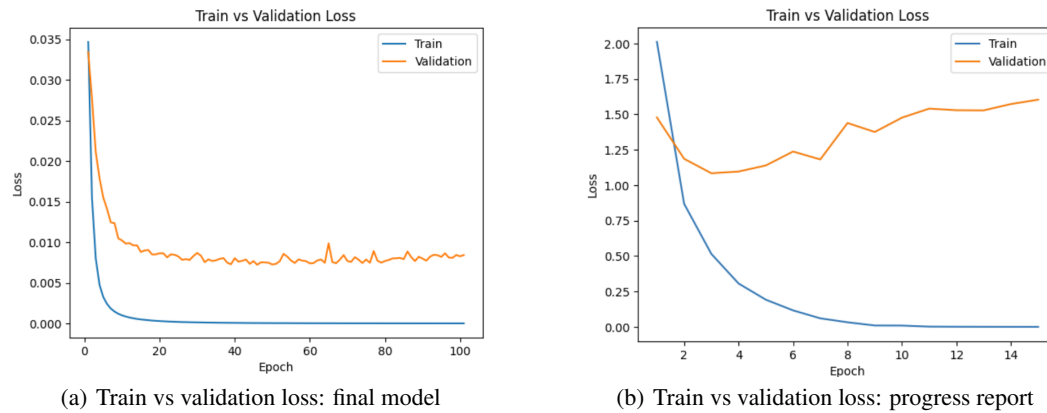


Figure 4: Comparison of training and validation loss for the final model and progress report

The team achieved a testing accuracy of 81.4% with our final model. More than a 16% increase from the progress report.

During model training, the group prioritized validation loss over training loss, as validation loss serves a more unbiased measure of the model's performance, bringing better generalization. After training with a unique set of hyperparameters, we select epochs with low validation loss, saving the models then run the testing set to determine a testing accuracy. Finally, the group selected the model with the highest testing accuracy.

From the above progress report graph, the validation curve does not overfit immediately as the one seen in the progress report. This improvement is due data augmentation, weight decay, dropout, and hyperparameter tuning discussed above and in previous sections.

Our final hyperparameters were:

- Learning rate: 0.0006
- Number of epochs: 101, chosen epoch for model: 90
- Batch size: 64
- Weight decay: 0.003

7 QUALITATIVE RESULTS

To better understand what our team's model can achieve, the group done qualitative analysis on the final model. Firstly, we created a customized function accepting images and predicts the genus to to which it belongs to. To test the function, we selected random genus names chosen from the possible categories, and then chose a random image from the test set belonging to each selected genus.

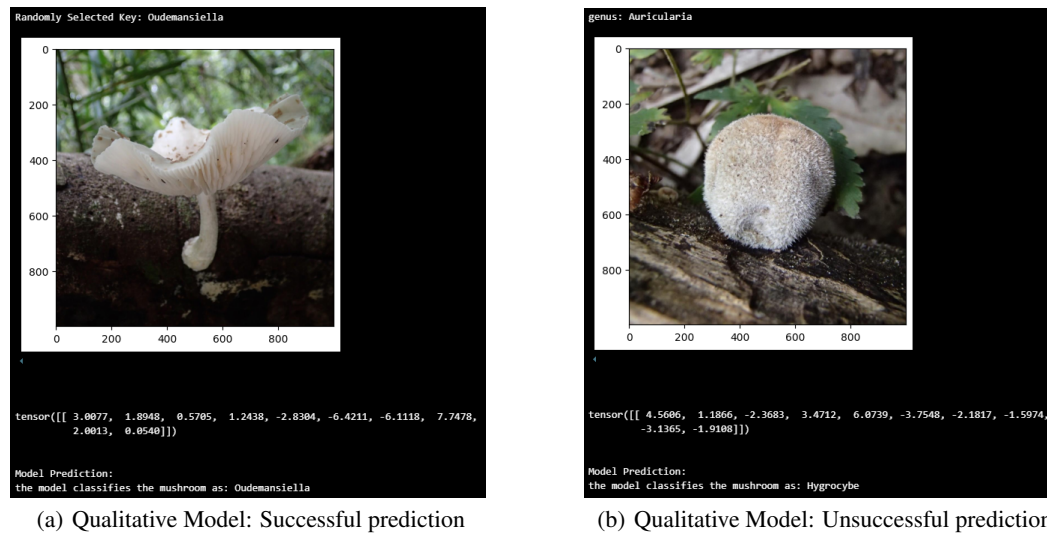


Figure 5: Comparison of successful and unsuccessful predictions

In the top of the above left figure, the team generated a random genus name and randomly selected an associated image. The image is then passed through the function, which returns the prediction. In the above sample, the prediction matches the generated name, showcasing that the model successfully identified the genus. The output tensor for the model's prediction is also displayed on the figure, showing the probability distribution of the image belonging to each of the classes.

In total, the team tested five sample outputs from the model, with three out of the five samples successfully identified.

The right figure above shows one of the wrong predictions made by the model. The two incorrect predictions share some similarities. Firstly, the model is more likely to identify an image if it contains more defining characteristics, such as unique colors, with the fungi zoomed-in and centered. Another common reason for the false prediction is when the image resembles those in other genres; a common occurrence in our data set.

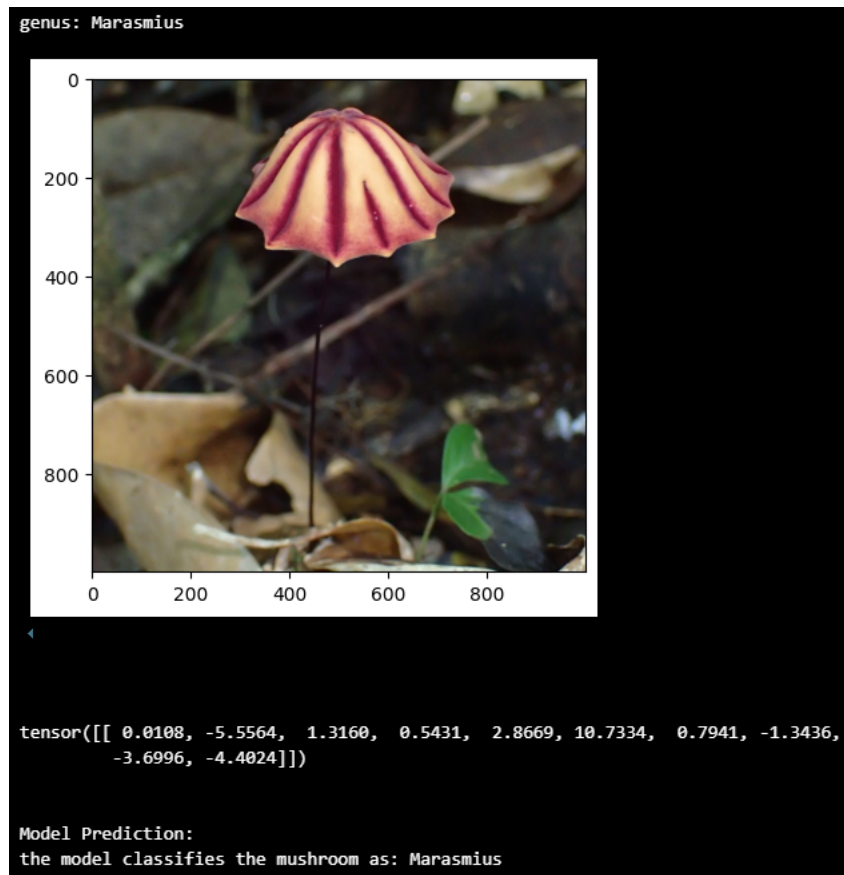


Figure 6: Qualitative Model: Successful prediction

The above figure shows another correct prediction made by the team’s model, supporting the claim that images with unique colors, patterns, and identifiable characteristics has a higher chance of being predicted correctly.

8 EVALUATION ON NEW DATA

We also evaluated our model on data collected from outside the MIND.Funga dataset. We collected 26 images of our dataset’s genera, with about 2-3 images per genus. This was collected from Wikimedia Commons, an online repository of permissively-licensed images. We took special care to ensure that the images were not included in the original dataset. As MIND.Funga is primarily made up of images from the public and academics, we are confident that no images from Wikimedia Commons were included in the original dataset (?).

Notably, our baseline model performs significantly worse than our deep learning model, with correct predictions on only 8% of new images. It overpredicts the genus *Hygrocybe*, the second most frequent genus in the dataset, on 61% of images. This may be a result of the baseline model’s lack of robustness with non-curated data, and largely suggests the rigidity of non-deep learning models without excessive hyperparameter tuning in handling image classification tasks.

Our main model, successfully identifies 65% of the new images, which can be attributed to the new images having varying picture resolutions (some with blurred backgrounds), and some with varying amounts of mushrooms and background differences than the photos found in the dataset.

9 DISCUSSION

Our baseline model incurred an accuracy of 61%, while our main model improved the accuracy by 20%, totaling 81.4% testing accuracy. Since the progress report, we have implemented data augmentation, to artificially increase the size of our dataset, and optimized which types of data alterations are most efficient for the model. This was paired with many hyperparameter changes, and a variety of tests with varying epochs and learning rates. We would have liked our final model to have a higher accuracy, but we are satisfied with our results, especially since we are comparing between 10 classes.

An area in which we believe we could have improved is in our evaluation on new data. Due to many of the images gathered as new data having potentially different environments and sources as the ones from our dataset, there were some discrepancies in the final accuracy for the model, especially with the baseline mode, while the main model did okay. As explained in the previous section, this mostly has to do with the amount of hyperparameter tuning in our main model, as our accuracy between models goes from 8% to 65% respectively.

Overall, we are satisfied with the results of our model. If we were to continue with this project, one area that we would like to potentially expand in is the amount genera our model can predict. Although it is important to note that the main reason for the limiting of the genera was due to the lack of sizable data for many of the genera in the dataset. The dataset also included some poor quality images, that were either not cropped well, or varied in the amount of mushrooms shown, some with one and others numerous. In addition to expanding to more genera, being able to predict both the genera, and provide an estimate for the specific species would be something we would be interested in trying, as many species within a genera can share many physical characteristics. Another interesting area we could look into is trying to classify an unidentified mushroom, and try to determine a genera for it, even if that mushroom's species has not been trained on in our model. Outside of new things we could try with our model, we would definitely like to focus on improving the accuracy first, and ensuring that our model is not overpredicting species with larger datasets.

This was an informative and interesting project to introduce ourselves into deep learning, especially in a topic like image recognition which is something we have been seeing used more frequently in our everyday lives.

10 ETHICAL CONSIDERATIONS

Many machine learning models rely on the "unauthorized uses of copyrighted" data for training purposes. This broadly results in infringement of the intellectual property rights and personal interests of the data owners (?). The dataset used in this project, MIND.Funga, is publicly licensed under a permissive copyleft license (CC BY NC 3.0) (?). Images were sourced from contributions from the public and academics. However, the organizing project does not disclose that public contributions will be released under this license (?), suggesting that the dataset may contain images that were not intended to be shared under this license as a result of inadequate informed consent.

The dataset also mostly contains images of tropical fungi from Brazil, with a small subset (approximately 500 out of 14 000) of the globally recorded number of fungi species (?). This results in a model that is primarily representative of Brazilian fungi species. As some fungi species can be poisonous, any practical real-world applications of our model may result in misclassifications, which may result in harm to users that rely on the model. This is a significant ethical concern, as the model may be used by individuals who are not experts in mycology, and may not be able to identify the potential risks of consuming certain fungi species.