DEEP LEARNING APPROACH TO MUSHROOM SPECIES CLASSIFICATION

Yanni Alan Alevras
Student# 1009330706
yanni.alevras@mail.utoronto.ca

Nicholas Biancolin
Student# 1005678901
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 (Lücking et al., 2020). This goal of this project is to develop a deep learning model that can accurately identify macrofungi (fungi with large bodies) based on their genus.

Broad applications of this model include identification in parks and other areas where humans may interact with nature. Deep learning forms a suitable choice for this task, as it constitutes a powerful and accurate method for image recognition and classification tasks, including in ecological settings (Schneider et al., 2019).

We propose the use of the MIND.Funga dataset (Drechsler-Santos et al., 2023), which has approximately 17 000 images of nearly 500 species of fungi. 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.

2 BACKGROUND & RELATED WORK

One key application of a fungi identification model is food safety and satisfaction.

In Bangladesh, a country with a large mushroom production, a farming method was developed by Rahman et al. (2022) using machine learning to classify which mushrooms are being harvested. This system intended to remove toxic species that may have grown in the same area as the target mushroom. Wang et al. (2018) created an algorithm to identify disease, discolouration, freshness, as well as other factors contributing to the commercial quality of a white button mushroom.

Our software would sort these mushrooms into genera, which would assist in identifying different types of mushrooms instead of certain physical features. Keeping with the theme of food quality, in Taiwan, Lu et al. (2019) developed a system to determine how much a mushroom has grown using an image recognition model, which produces an estimate based on images from different times. This used a convolutional neural network (CNN) to provide these results, which will be similar to our model's architecture. In the Chinese province of Yunnan, H. Zhao et al. (2021) created a wild mushroom identification model that used 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 educational tool provides useful information about a mushroom, after it has been identified using a classification model (.Sh, 2023).

3 Data Processing

Many species in the dataset have a relatively small amount of images associated with them (< 30 images), which may be detrimental when training, testing, and validating our model. To mitigate this, we intend to group images into larger "buckets" corresponding to a higher level of taxonomic classification, i.e., grouping by genus instead of by species. This is a simple way to reduce the number of classes our model must be trained on, and to increase the amount of data to a sufficient amount that model features can be reasonably trained. Genera form an ideal way to group images together — biologically species of the same genus share many physical characteristics, and they share a root name (the first word in the species name), which enhances ease of processing. Existing literature (Hollister et al., 2023) also suggests that models trained to classify genera tend to have better accuracy and are less difficult to train than species-level classification.

Once we have performed this combination, any genera with less than 75 images will be discarded from the dataset. This matches the lower threshold of published ecological classifiers, like iNaturalist (Shepard, 2022) or models trained on the ETHZ human dataset (Schneider et al., 2019). It also matches what is considered by Farley et al. (2024) to be the lower end of images per class to train a model without overfitting.

4 Architecture

Since our project primarily involves images, we plan to follow a workflow similar to that described by S. Ren et al. (2017). We will first apply a regional proposal network (RPN) that first isolates the macrofungi from the image. RPNs are convolutional networks that are able to identify regions of interest (proposals) in an image.

We will then apply a convolutional neural network (CNN) for detection and identification training. Prior literature (Barré et al., 2017) suggests that a CNN is best suited for computer vision classification problems, especially in ecology-related settings.

We also intend to use a stochastic gradient descent-based optimizer, as it is a common choice for training deep neural networks (James et al., 2023).

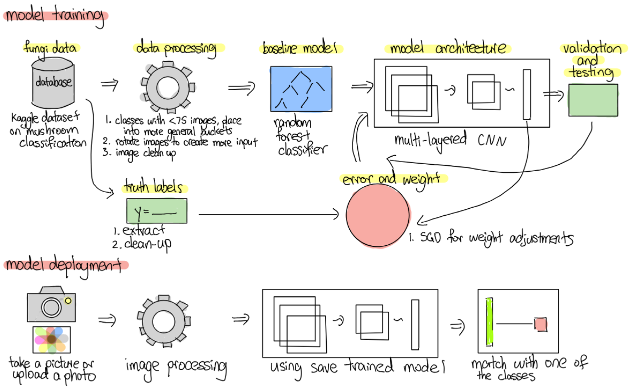

Figure 1: Proposed model training and deployment pipeline

5 Baseline Model

For the baseline model we are comparing our model against, we selected a random forest classifier model. Our goal is to identify several hundred species of macrofungi, making this a multiclass classification problem. Decision trees like random forest are generally suited for multiclass problems (Gall et al., 2012). Other methods, like support vector machines, are generally suitable primarily for binary classification problems (James et al., 2023).

We will compare the classification accuracy and training time for both models. The random forest model will be trained using the scikit-learn library with the same data and data processing steps as the neural network model, following the steps outlined in Shafl (2023).

6 Project Plan

Task Breakdown and Responsibilities (Green = completed, Yellow = in progress, Red = not started)
W = wrote/coded/drawn, ED = edited, MR = major revision, FP = Final section proofread

Project Proposal	**Eric Liu**	**Jason Zhang**	**Nicholas Biancolin**	**Yanni Alevras**
Introduction (June 4th, 11:59pm)	W	W	W ED	W
Illustration/Figure (June 5th 11:59pm)	W	ED		
Background and Related Work (June 5th 11:59pm)		ED		W
Data Processing (June 5th 11:59pm)		W		
Architecture (June 5th 11:59pm)	ED	W		
Baseline Model (June 5th 11:59pm)	ED	ED	W	
Ethical Considerations (June 5th 11:59pm)			ED	W
Project Plan (June 5th 11:59pm)	W			
Task research and distribution (June 5th 11:59pm)	W			
Risk Register (June 5th 11:59pm)			W	
Github, References (June 5th 11:59pm)	ED	W	W	W
Structure, Grammar and Mechanics (June 5th 11:59pm)	ED	W		
Document in LaTeX (June 5th 11:59pm)	W	W MR ED		
Editing (June 6th 11:59pm)	ED	ED		
Final Document Proofread (June 6th 11:59pm)	ED	ED		

Table 1: Project Proposal Task Breakdown

Project Progress Report	**Eric Liu**	**Jason Zhang**	**Nicholas Biancolin**	**Yanni Alevras**
Brief Project Description (June 30th, 11:59 pm)	W	W	W	W
Individual Contributions and Responsibilities (June 30th, 11:59 pm)	W	ED		
Contributions - Data Processing (June 30th, 11:59 pm)		W	ED	
Contributions - Baseline Model (June 30th, 11:59 pm)			W	ED
Contributions - Primary Model (June 30th, 11:59 pm)	ED			W
Illustrations (July 1st, 11:59 pm)	W	ED		
Latex format (July 2nd, 11:59 pm)	W	W		
Editing (July 3rd, 11:59 pm)	ED	ED	ED	ED
Final Proofread (July 4th, 6:00 pm)	W	W	W	W

Table 2: Project Progress Report Task Breakdown

Project - Training and Testing	**Eric Liu**	**Jason Zhang**	**Nicholas Biancolin**	**Yanni Alevras**
In charge of code connection/solving merge conflicts (August 10th, 11:59pm)				
Data Cleaning (June 16th, 11:59pm)				
Image Grouping (June 16th, 11:59pm)				
Transfer data to training format (June 16th, 11:59pm)				
Data annotations, splitting (June 16th, 11:59pm)				
Model implementation (June 19th, 11:59pm)				
CNN architecture (June 19th, 11:59pm)				
Training Loop (June 19th, 11:59pm)				
Hyperparameter adjustments (July 15th, 11:59pm)				
Training (July 15th, 11:59pm)				
Validation (July 15th, 11:59pm)				
Testing (August 10th, 11:59pm)				
Iterative (if needed) (August 10th, 11:59pm)				
Evaluation (August 10th, 11:59pm)				
Documentation (August 10th, 11:59pm)				
Resource management (August 3rd, 11:59pm)				
Analyze Result for Presentation and Project (August 3rd, 11:59pm)				

Table 3: Project Training and Testing Task Breakdown

Presentation	Eric Liu	Jason Zhang	Nicholas Biancolin	Yanni Alevras
Presentation Brainstorm (August 5th, 11:59pm)				
Problem - slides (August 5th, 11:59pm)				
Data Processing - slides (August 5th, 11:59pm)				
Model - slides (August 5th, 11:59pm)				
Result - slides (August 5th, 11:59pm)				
Slides Editing (August 5th, 11:59pm)				
Individual Practice (August 7th, 11:59pm)				
Group Practice (August 7th, 11:59pm)				
Record Presentation (August 7th, 11:59pm)				
Editing (August 10th, 11:59pm)				

Table 4: Presentation Task Breakdown

Project Final Report	Eric Liu	Jason Zhang	Nicholas Biancolin	Yanni Alevras
Latex Formatting (August 12th, 11:59pm)				
Introduction (August 7th, 11:59pm)				
Illustration (August 7th, 11:59pm)				
Background and Related Work (August 7th, 11:59pm)				
Data Processing (August 7th, 11:59pm)				
Architecture (August 7th, 11:59pm)				
Baseline Model (August 7th, 11:59pm)				
Qualitative Results (August 7th, 11:59pm)				
Quantitative Results (August 7th, 11:59pm)				
Evaluation of Model (August 7th, 11:59pm)				
Discussion (August 7th, 11:59pm)				
Ethical Considerations (August 7th, 11:59pm)				
Project Difficulty (August 7th, 11:59pm)				
Editing (August 12th, 11:59pm)				
Final Proofread (August 14th, 11:59pm)				

Table 5: Project Final Report Task Breakdown

Note: The tasks and their assignments may change, particularly when a major task requires splitting into smaller more feasible ones.

We (the team) plan to tackle all the tasks listed in the table above individually, as all tasks are planned out and distributed individually with clear deadlines. If any member cannot complete their task(s) on time, they must inform the team at the earliest convenience. A team meeting will then be held to determine whether to reassign the task, extend the deadline, or remove the task if it is of low priority. Given the team's extensive experience working with each other, we (the team) have gained mutual trust and can resolve any potential issues together.

Coding tasks have been distributed so members will work on completely different sections, then push their functions into the main program. In addition, there is a member in charge of function integration and solving merge conflicts to ensure we will not write over each other's code.

We (the team) aim to meet at least twice a week via Discord calls, where the team discusses progress, tackles issues, and brainstorms ideas. The current meeting time is Wednesday and Saturday at 9 pm (Eastern Time), though this may change in the future. Our team's main source of communication will be Discord, and all members will check and reply to messages at least once every 12 hours.

7 Risk Register

As with any large scale project, there are many associated risks:

- What if a team member elects to drop the course?
- What if the model isn't training properly?
- What happens if there's scope creep?
- What happens during time crunches?

We are privileged to have been friends for over a year. We have respect for and have worked with each other in the past. We recognise that a group member dropping the course, while far from ideal, is a choice that is made for a reason. Should such a case occur, we will meet and discuss how extra work will be divided and make a plan to ensure the project continues as planned with an extra workload.

We intend to avoid quality issues with the model by setting a timeline that leaves time for multiple rounds of iteration, further training, and debugging. If the model's performance is subpar, we intend to first make changes to the data processing and architecture. With a relatively small amount of data per class, deliberate improvements to both areas may be able to immediately improve performance.

As with many projects, this proposal was written with the understanding that circumstances may change as the semester continues. If left unchecked, project requirements can change drastically from the original proposal. We will keep this in mind as we discuss and work on the project to avoid any misconceptions.

Finally, we have experienced many short time constraints in previous projects and understand how they may affect the quality of the final product. The timeline we intend to set leaves extra room for unexpected issues. Should a time crunch occur, we will evaluate our responsibilities and progress and work according to a priority queue.

REFERENCES

ShroomID, November 2023.

Pierre Barré, Ben C. Stöver, Kai F. Müller, and Volker Steinhage. LeafNet: A computer vision system for automatic plant species identification. *Ecological Informatics*, 40:50–56, July 2017. ISSN 1574-9541. doi: 10.1016/j.ecoinf.2017.05.005.

Elisandro Ricardo Drechsler-Santos, Fernanda Karstedt, Thiago Zimmermann Loureiro Chaves, Mahatmā Titton, Kelmer Martins da Cunha, Eloísa Leopoldo, Genivaldo Alves-Silva, Diogo Rezende, Thiago Kossmann, Emerson Luiz Gumboski, and Aldo von Wangenheim. MIND.Funga, July 2023.

Patrick Farley, Nitin Mehrota, and Eric Urban. Improving your model. *Microsoft Azure*, February 2024.

Juergen Gall, Nima Razavi, and Luc Van Gool. An Introduction to Random Forests for Multi-class Object Detection. In Frank Dellaert, Jan-Michael Frahm, Marc Pollefeys, Laura Leal-Taixé, and Bodo Rosenhahn (eds.), *Outdoor and Large-Scale Real-World Scene Analysis*, pp. 243–263, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. ISBN 978-3-642-34091-8.

H. Zhao, F. Ge, P. Yu, and H. Li. Identification of Wild Mushroom Based on Ensemble Learning. In *2021 IEEE 4th International Conference on Big Data and Artificial Intelligence (BDAI)*, pp. 43–47, 2021. doi: 10.1109/BDAI52447.2021.9515225.

Jack D. Hollister, Xiaohao Cai, Tammy Horton, Benjamin W. Price, Karolina M. Zarzyczny, and Phillip B. Fenberg. Using computer vision to identify limpets from their shells: A case study using four species from the Baja California peninsula. *Frontiers in Marine Science*, 10, 2023. ISSN 2296-7745.

Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, and Jonathan Taylor. *An Introduction to Statistical Learning with Applications in Python*. Springer International Publishing, 2023.

Chuan-Pin Lu, Jiun-Jian Liaw, Tzu-Ching Wu, and Tsung-Fu Hung. Development of a Mushroom Growth Measurement System Applying Deep Learning for Image Recognition. *Agronomy*, 9(1), 2019. ISSN 2073-4395. doi: 10.3390/agronomy9010032.

Robert Lücking, M. Catherine Aime, Barbara Robbertse, Andrew N. Miller, Hiran A. Ariyawansa, Takayuki Aoki, Gianluigi Cardinali, Pedro W. Crous, Irina S. Druzhinina, David M. Geiser, David L. Hawksworth, Kevin D. Hyde, Laszlo Irinyi, Rajesh Jeewon, Peter R. Johnston, Paul M. Kirk, Elaine Malosso, Tom W. May, Wieland Meyer, Maarja Öpik, Vincent Robert, Marc Stadler, Marco Thines, Duong Vu, Andrey M. Yurkov, Ning Zhang, and Conrad L. Schoch. Unambiguous identification of fungi: Where do we stand and how accurate and precise is fungal DNA barcoding? *IMA Fungus*, 11(1):14, July 2020. ISSN 2210-6359. doi: 10.1186/s43008-020-00033-z.

Hasibur Rahman, Md. Omar Faruq, Talha Bin Abdul Hai, Wahidur Rahman, Muhammad Minoar Hossain, Mahbubul Hasan, Shafiqul Islam, Md. Moinuddin, Md. Tarequl Islam, and Mir Mohammad Azad. IoT enabled mushroom farm automation with Machine Learning to classify toxic mushrooms in Bangladesh. *Journal of Agriculture and Food Research*, 7:100267, March 2022. ISSN 2666-1543. doi: 10.1016/j.jafr.2021.100267.

S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39 (6):1137–1149, June 2017. ISSN 1939-3539. doi: 10.1109/TPAMI.2016.2577031.

Stefan Schneider, Graham W. Taylor, Stefan Linquist, and Stefan C. Kremer. Past, present and future approaches using computer vision for animal re-identification from camera trap data. *Methods in Ecology and Evolution*, 10(4):461–470, April 2019. ISSN 2041-210X. doi: 10.1111/2041-210X. 13133.

Adam Shafl. Random Forest Classification with Scikit-Learn. *Datacamp*, 2023.

Alex Shepard. The Latest Computer Vision Model Updates. *iNaturalist*, April 2022.

Fengyun Wang, Jiye Zheng, Xincheng Tian, Jianfei Wang, Luyan Niu, and Wenjie Feng. An automatic sorting system for fresh white button mushrooms based on image processing. *Computers and Electronics in Agriculture*, 151:416–425, August 2018. ISSN 0168-1699. doi: 10.1016/j.compag.2018.06.022.