Group members: Tianming Tan, Zhenshuo Xu, Chenghao Wu

Updated version of Proposal

Problem:

The problem we aim to solve is the classification of mushroom species based on images and determining whether they are edible or poisonous. The dataset we are using contains 96 images of 8 different mushroom species. In the 8 mushroom species, panther cap and amanita gemmata are known for being poisonous while the other 6 are edible. Our goal for this project is to develop a model that can accurately classify the different mushroom images into their respective classes and predict their edibility.

Motivation:

Mushrooms, a delicious food, are notably challenging to identify safely due to the striking resemblance between many toxic and edible varieties. Numerous poisoning incidents occur when people consume wild mushrooms without sufficient knowledge. We plan to use various image processing methods to analyze the visual features of mushrooms, including shape, color, and texture. We hope this will effectively identify the types of mushrooms and determine their toxicity, thereby helping to reduce poisoning incidents.

Dataset:

https://www.kaggle.com/datasets/daniilonishchenko/mushrooms-images-classification-215

Method:

We would use machine learning and methods of data augmentation from computer vision to complete the task of classification. We plan to modify and improve the CNN model to classify different classes of mushrooms and whether they are poisonous by adding and changing more codes for more fittable functions. During the preparation period, we may use data augmentation to increase the dataset, like flip and rotation. We then prepare the dataset and divide it to the train dataset and test dataset.

After the data preprocessing, we would adjust the code for the model and parameters so that the model can be trained to classify whether they are poisonous and their classes. Finally, based on the pattern reasoning, we can determine the accuracy and correctness of the classification of the model during the evaluation period. By analyzing the result, we can find out the problems and further improve the model so that it can better predict the classes and poison mushrooms.

For the ethics and safety consideration ,given the life-threatening implications of misidentifying edible and poisonous mushrooms, it's essential to include disclaimers about the potential for error.

Process/Steps:

For our model, we randomly selected a subset of eight types of mushrooms. Those include Amanita gemmata, Bronze Bolete, Crimped Gill, Fairy Ring Champions, Grey Knight, Jelly Ears, Lion's Mane, and Panther Cap. As we stated in our proposal, our goal for this project is to create a model that can sufficiently classify between various types of mushrooms to distinguish between those that are edible and those that are poisonous.

After selecting eight different types of mushrooms, we separated them into training and validation sets by utilizing the **train_test_split** function from the **skilearn.model_selection** module, **os**, and **shutil**. The script is designed to handle images organized in subdirectories with each directory representing a different class of mushrooms.

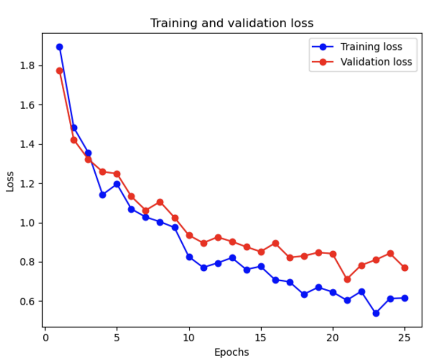
Before training the model, we applied various data augmentation techniques to increase the size of our training dataset by creating modified versions of the existing mushroom images. This technique will allow the model to learn to recognize each mushroom under certain conditions, which can enhance its performance on unseen data or images. Without data augmentation, the model tends to overfit quickly. To apply data augmentation to our data, we utilized the "torchvision.transform" module to define multiple augmentation pipelines to apply to the training data. To make training more efficient and effective we applied various transformations like horizontal flips, rotations, color jitter, and normalization. Then after the transformation we prepared three different dataset with each representing different augmentation techniques and then combined all three into a combined dataset that allows the model to learn from and diverse data. during training. We then randomly split the combined dataset into training, validation, and test sets with a batch size of 8. We experimented with different batch sizes such as 16, 32, and 64. We noticed that smaller batch sizes provide more delicate gradient updates, while large batch sizes accelerated the training speed. Through various experiments from batch size of 8 to 32, we observed a noticeable increase in training speed, but the accuracy on the test set tends to decrease.

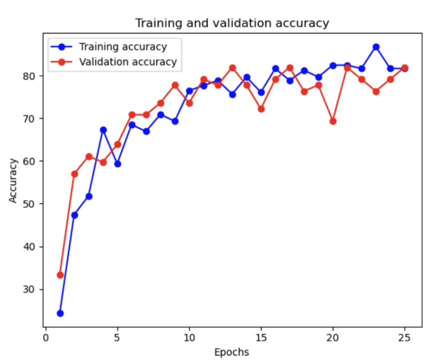
After finishing all the data augmentation, we fed our data into a convolutional neural network (CNN). Our CNN consists of two convolutional layers with normalization followed by max pooling, dropout, and two fully connected layers for classification. Regarding the number of layers, we experimented with adding more convolutional and fully connected layers. Although increasing the number of layers sometimes improves the model's expressive power. However, we did not observe any significant improvement in model performance.

Learning rate is also a key step in optimizing model performance. In our experiments, we started with a relatively high learning rate of 0.01 and made our way down to 0.0005. Throughout the experiment, we noticed that adjusting the learning rate could significantly reduce the validation loss from 2 down to 0.8. To prevent overfitting, we also implemented an early stopping function to halt the training process early if the validation loss stops improving after a specific number of epochs. Our model trains for 25 epochs with each outputting the loss, accuracy, validation loss, and the validation accuracy.

Results

After training for 25 epochs, our results came to loss: 0.6155, accuracy: 81.67%, validation loss: 0.7709, validation accuracy: 81.94%. Below is a plot for loss, validation, accuracy, validation loss, and validation accuracy. The plot on the left shows that as the number of epochs increases the training loss and validation loss tend to decrease. The plot on the right shows that as the number of the epochs increases the training and validation accuracy, which is what we want to see for our model.



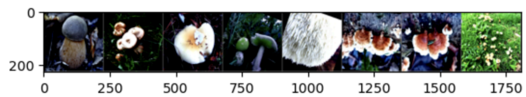


Then we test our model using the images in our dataset to see if the model accurately predicts each mushroom type correctly. Below is the code snippet and the output where we compare the ground truth and the predicted mushroom types.

```
43]:
dataiter = iter(test_loader)
images, labels = next(dataiter)

outputs = model(images.to(device))
_, predicted = torch.max(outputs, 1)

imshow(torchvision.utils.make_grid(images))
print('GroundTruth: ', ' '.join(f'{classes[labels[j]]}' for j in range(len(images))))
print('Predicted: ', ' '.join(f'{classes[predicted[j]]}' for j in range(len(images))))
```




```
GroundTruth:  bronze_bolete fairy_ring_champignons amanita_gemmata amanita_gemmata lions_mane crimped_gill crimped
_gill fairy_ring_champignons
Predicted:  bronze_bolete fairy_ring_champignons amanita_gemmata jelly_ears lions_mane crimped_gill crimped_gill f
airy_ring_champignons
```

To make sure that our model can accurately predict the type of mushroom, we decided to use images of the same species, but outside of the dataset. The image below shows how our model successfully predicts an outside grey knight image (111.jpg) that we found on Google. The results turned out to be accurate and it also states that grey knight is edible and not poisonous, which is correct according to wikipedia (https://en.wikipedia.org/wiki/Tricholoma_terreum)

Predicted: grey_knight

```
The predicted class is: grey_knight - non-poisonous
Test Loss: 0.6277, Test Accuracy: 83.78%
```

Presentation link:
https://youtu.be/LT9Y-jNaK0c

Github link:
https://github.com/Ttan0728/174_project/tree/main

Contribution:
Tianming Tan: worked on final report, worked on code but was unsuccessful (accuray ~60%)
Zhenshou Xu: Worked and finished the model(hyperparameter) and test, worked on the PPT
Chenghao Wu: Worked on the PPT, the presentation, and the code for model(CNN), training.