# Deep Shrooms!

*Group members: Jeremy Allen, Clas Blank, Tristan Broughton, Amanda Cheng and Brian Hicks*

## *Project Overview*

The purpose of our project is to use machine learning to classify an image of a mushroom by name and determine whether it is either edible or poisonous. The goal will be measured in terms of our model's accuracy when classifying new images, with the intention of reaching a level compatible with the industry standard. We found a data set from the 2018 Fungi Classification Challenge, an FGVCx competition as part of the FGVC5 workshop at CVPR 2018. The set has over 100,000 images and 1300 different species. We have a good start to our project and are already training an AutoML model!

We do not expect our project to be used in a realistic context, since there are potential legal issues associated with misclassification. If we are able to get the model to first classify different fungi taxonomies, we can then add the option of edibility and non edibility. If we have time near the end of the quarter, we thought it might be helpful to integrate our model into a mobile application as a final deliverable. That being said, we do not intend on putting it on the market, as it could potentially harm users.  Therefore, the intended users of our project are computer scientists who want to learn more about using computer vision and machine learning for image classification.

## *Background, Difficulty and Relevance*

Cyrill Glockner used a TensorFlow based model to classify mushrooms as either poisonous or not, and posted his research and findings online. The dataset for this project included 600 images of edible mushrooms, and 600 images of poisonous mushrooms. These images were extracted from a google search query.
https://towardsdatascience.com/deep-learning-and-poisonous-mushrooms-4377ea4c9b80

Here is the GitHub for a team from Denmark competing in the 2018 Fungi Classification Challenge: https://github.com/visipedia/fgvcx_fungi_comp

Google has REST APIs for computer vision and machine learning--specifically, Cloud Vision and Cloud AutoML APIs. This will make it relatively easy to construct a model, train it, and then implement an application that uses our model for classification. The more difficult task will be finding an accurate and large-enough dataset.

Our project is to use machine learning to identify various types of mushrooms, which touches upon many ideas that we will be covering in the course of the quarter, such as using learning agents to better identify the mushrooms.

## *Features, Requirements, and Evaluation Criteria*

The targeted user base for this product is outdoor enthusiasts and those who are interested in learning more about mushroom species. Deep Shrooms is a mobile application that will identify mushroom species based on the user's uploaded image or the camera roll on their phones. This is particularly helpful if the user wants to learn more about a wild mushroom that he/she has found.

### Features

1. Allow the user to be able to upload a mushroom image, or take a photo of one, so that the app will identify it with a certain accuracy.
2. Allow user to access the camera/upload option or their basket[1] of mushrooms within a couple taps of their screen.
3. In the basket, click on the mushroom to get more info:
    a. Mycological characteristics
    b. More images
    c. Environment
    d. Edible Trait
    e. Date discovered
    f. Location discovered

[1] A *basket* is a collection of previously discovered mushrooms.

4. An interface to prompt for different angles of the same mushroom to get more samples to compare with.
5. An interface to give the user multiple options to start distinguishing the mushroom based off touch and smell versus just its physical features.

## Requirements

6. The performance of the model after training will be close to 90% accurate.
    a. The database is based off of mushrooms in Denmark.
    b. Google's Cloud Vision and Cloud AutoML API will allow us to identify the mushroom species based on the data it accumulates from our 12GB of images.
7. The application will connect with the phone's camera and camera roll to access the images.
8. The application will access the additional mushroom data through websites dedicated to sharing mushroom species information.
9. The application will ask for more images of the same mushroom with different angles if the model cannot provide identification above a certain threshold.
10. If the application returns two or more very similar, indistinguishable mushrooms by physical features, it will instruct the user to base the distinction off touch and smell.

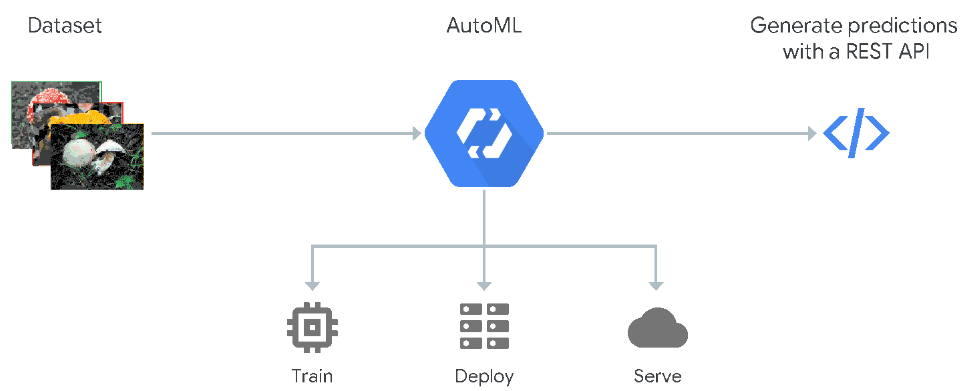
## Evaluation Criteria

11. The model should conform to industry standard of >90% accuracy for mushroom identification. The dataset used will be split into training and testing sets.
12. The app should not crash when in use and should always have access to the phone's camera and camera roll. Evaluation will be done with unit testing of each app feature. Bugs will be documented and addressed.
13. The app should display accurate additional information according to the identified mushroom species.
14. The app should provide an easy-to-use interface for managing images, as well as prompting the user for more information when appropriate; this can be evaluated via user-testing.

15. The app should provide accurate but distinct information that will help the user identify the mushroom through non-physical aspects.

## *System Design and Architecture*

The backend is all self contained within the AutoML platform. A user simply uploads a clean dataset that is properly labeled. Our goal is to implement this model within a mobile application in order for real world use. The front end is designed to guide the user to correctly identify a particular mushroom. The backend is designed to store the models used to identify the mushroom and information about the user's collected mushrooms. The system is expected to correctly identify a particular mushroom and, if not, generate a small group of possibilities for the user to work with outside of the app.




The data flow for the front end is as follows. First the user uploads a picture from their phone or takes a picture with their camera. The photo is then fed into the first model trained on every angle of all the mushrooms and generates a list of 3 possible species with a score next to each one. The system will ask the user for a new picture of angle two (haven't decided the specific angles yet). This second picture will be fed into a model trained on the second angle of the mushrooms and generate a new list of possible species with corresponding scores. The user will be prompted to upload a third and final picture. With each step, the score will often vary due to the particular angle of the mushroom. The system will average out all three scores and if the score for the expected mushroom falls below a certain threshold the system will tell the user "Unable to identify" and give a small list of possible species for the user to compare with outside of the app. Due to the extreme difficulty of mushroom identification,this is not a one picture identification app. We hope through the use of multiple angles and models, we can more precisely identify mushrooms.

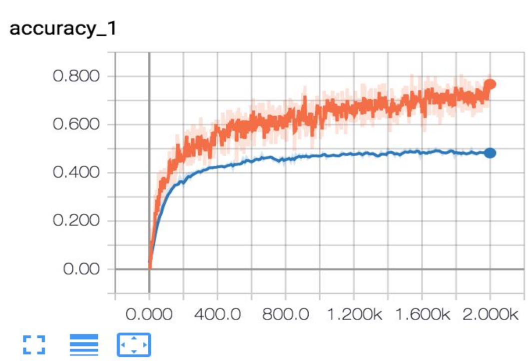
# *Implementation*

When we were ready to implement our model we decided on a few things. First off, since we were getting 50% accuracy with the tensorflow model, we did not see it necessary to transition into the AutoML platform. While AutoML provides better resources for analyzing the results, the time and storage required to upload our large data set got in the way. So our final implementation worked like so:

1. Open a command line terminal. Run the code to train the model. Here we could specify things like number of training steps and learning rate.

```
Jeremys-MBP:tensorflow-for-poets-2 Jeremy$ python3 -m scripts.retrain \
>   --bottleneck_dir=tf_files/bottlenecks \
>   --how_many_training_steps=2000 \
>   --model_dir=tf_files/models/ \
>   --summaries_dir=tf_files/training_summaries/"${ARCHITECTURE}" \
>   --output_graph=tf_files/retrained_graph.pb \
>   --output_labels=tf_files/retrained_labels.txt \
>   --architecture="${ARCHITECTURE}" \
>   --image_dir=tf_files/shroom_photos
```

2. Once the training finished, we could check TensorBoard to see how well our model did. An example of the graph is shown below. The orange line indicates the accuracy of the model on the training data while the blue line indicates the accuracy on the test set. The latter being a more accurate performance of the model.



3. Finally, we could use the model to test on our own images. The code for that looks like this:

```
Jeremys-MBP:tensorflow-for-poets-2 Jeremy$ python3 -m scripts.label_image \
>     --graph=tf_files/retrained_graph.pb  \
>     --image=tf_files/shroom_photos/agaric1.jpeg
```

# *Validation*

When we first started the project, we had high hopes for accuracy. Looking at other image recognition projects, we kept seeing this 90% accuracy floating around out there. So for a model to be considered “accurate”, it needed to pass this threshold. With that expectation in mind, getting our initial model accuracy was a bit of a blow to the gut. What we failed to realize was that these other projects were classify things like candy bars, types of cars, houses, etc. A lot of information can be extrapolated from a picture of these specific things. On top of that, these objects are very distinct from each other on a visual basis. For mushrooms, the story is a little different. Touch, sense, and smell, to name a few, all play a role in identifying mushrooms. Also, two completely different mushroom could look exactly alike! So our initial criteria may seem slightly far fetched as we stand here at the finish line.

1. **The model should conform to industry standard of >90% accuracy for mushroom identification.**
   **As mentioned above, our model performs at around 50% accuracy, not too bad considering the classification problem at hand.**
2. **The app should not crash when in use and should always have access to the phone’s camera and camera roll.**
   **We were fairly set on implementing within a mobile application early on but never followed through with it, due to lack of app dev experience. However, our python code never crashed!**
3. **The app should display accurate additional information according to the identified mushroom species**
   **We got the python interface to display the top five mushroom species and the percentages of each one.**

4. **The app should provide an easy-to-use interface for managing images, as well as prompting the user for more information when appropriate; this can be evaluated via user-testing.**
   **Our python interface (DeepShroomHunter 5000) looked like this:**

```
Jeremys-MBP:tensorflow-for-poets-2 Jeremy$ ./mushroomHunter.sh
Welcome to DeepShroomHunter 5000!
PLEASE ENTER MUSHROOM OR DONE:
m1
2018-12-12 16:29:49.362133: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU s
upports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA

Evaluation time (1-image): 0.590s

trametes versicolor (score=0.44594)
plicatura crispa (score=0.18664)
stereum hirsutum (score=0.12425)
laccaria laccata (score=0.07898)
phlebia radiata (score=0.02740)
PLEASE ENTER MUSHROOM OR DONE:
```

   **A user would simply enter the name of the mushroom (based on filename) and get the expected species.**
5. **The app should provide accurate but distinct information that will help the user identify the mushroom through non-physical aspects.**
   **Something we really wanted to work on, if given more time, was incorporating other aspects of the mushroom. If the user could not guess the species in one try, we could prompt him or her for more information about the certain mushroom.**

# *Conclusion*

In conclusion, we learned a lot about the overall structure of working with and setting up a RNN model. Since a lot of our code was from the tensorflow-for-poets tutorial, we learned more about setting up and manipulating the data. Rather than stress about creating a neural network from scratch, we could build on top of already fine tuned models. It was interesting to see which parameters changed what. For example, by messing with the learning rate parameter built into the tensorflow model, we increased accuracy while only slightly increasing runtime. One of the biggest issues we ran into was the inability to capture other features about the mushroom through the images.

Mushroom identification is difficult, it requires knowledge about the texture, gills, smell, etc. Gathering that data through an image is next to impossible. What our project does, however, is give the mushroom hunter a jumping off point. Through a single image, we could determine, with fifty percent, accuracy the species of a given fungi. That is quite impressive with over 113 species to go off of. Overall, we were happy with our result and thoroughly enjoyed the process. In the future, there are certain areas we could build out and improve the idea.