Edible/ Poisonous Mushroom Image Classifier

Solution Design

Philip Reed w18011988

Solution Motivation

Mushroom foraging is the act of picking mushrooms (the fruiting body of a fungus), primarily to use for food or medicine. Mushrooms grow across the globe in a variety of environments and have been consumed by humans since ancient times, with reliable evidence of mushroom consumption dating back to several hundred years BC in China [1].
Although most mushrooms these days are cultivated, there is a growing interest in foraging as people become more interested in self-sufficiency and natural food (see figure 1).
Foraging for mushrooms can be a precarious activity for a novice to undertake, as there are many mushrooms that contain toxins which can be poisonous and potentially fatal to humans, making mushroom the main cause of natural poisoning [2].
The goal of this project is to create an Image classifier which takes an input of a link to an image of any mushroom or fungi, and classify the image as “edible” or “poisonous” and give a percentage confidence rating.
This binary-classification is hoped to provide the most reliable solution for mushroom foragers, who just want to be sure what they are picking is edible.
To achieve this goal, we will use a convoluted neural network, using TensorFlow and Keras through python, more details will be given in the Environment Setup section of this document.

Although there are other mushroom Identification applications and services, miss-classification can occur, largely due to these classifiers being multi-class (identifying specific species), which overall means there is a smaller amount of training data for each class available. Our goal is to provide the most accurate classifier just to establish whether or not a mushroom is edible.

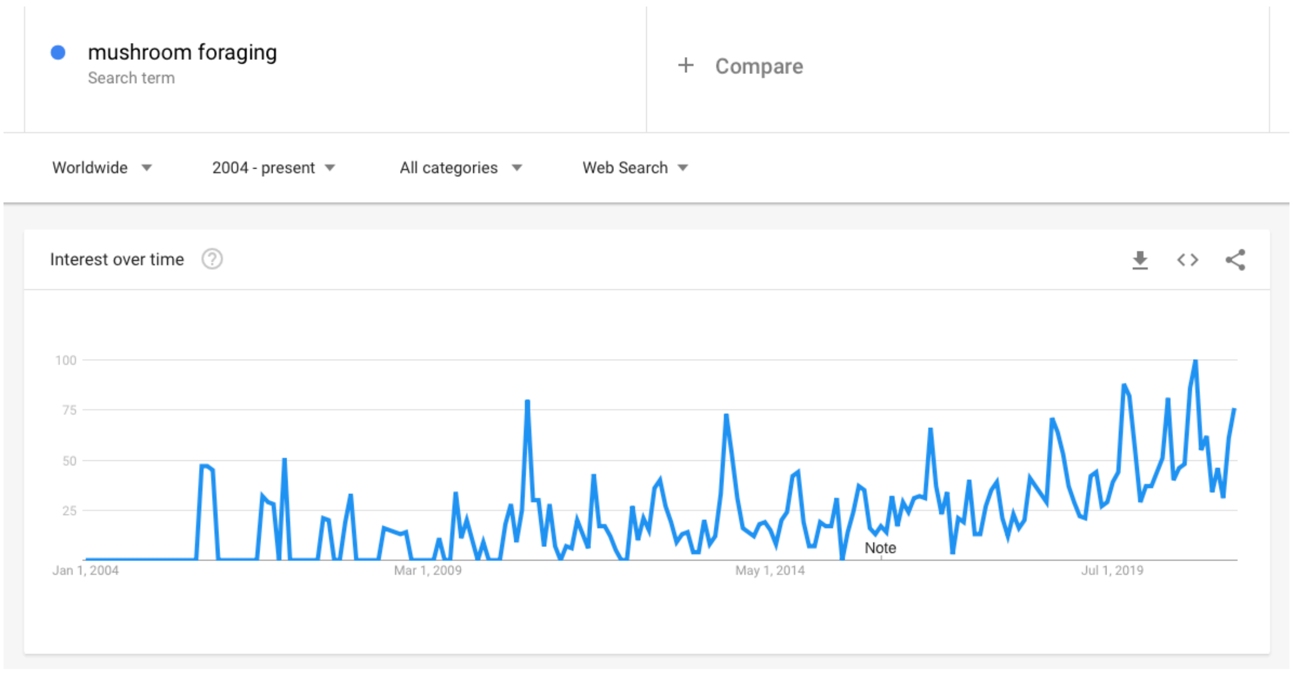


Figure 1: Google Trends graph showing worldwide interest in mushroom foraging increasing.

Proposed Solution

The initial proposed solution was to train a classifier that would be trained using a pre-made dataset of images of popular mushroom species. A dataset of 8 major mushroom species family from Kaggle user "CatoDogo" [3] was identified as a means to a potential solution. The dataset included 9 folder each containing 300-1500 images of mushrooms of the most common Northern European mushrooms, provided by the mycologist's society of Northern Europe. The Kaggle dataset was included an early prototype baseline implementation, however it became apparent that mushrooms from the same species can be either edible or poisonous, so identifying the general species of mushroom was not of any real use in terms of classifying a mushrooms edibility.

Alternative Solutions

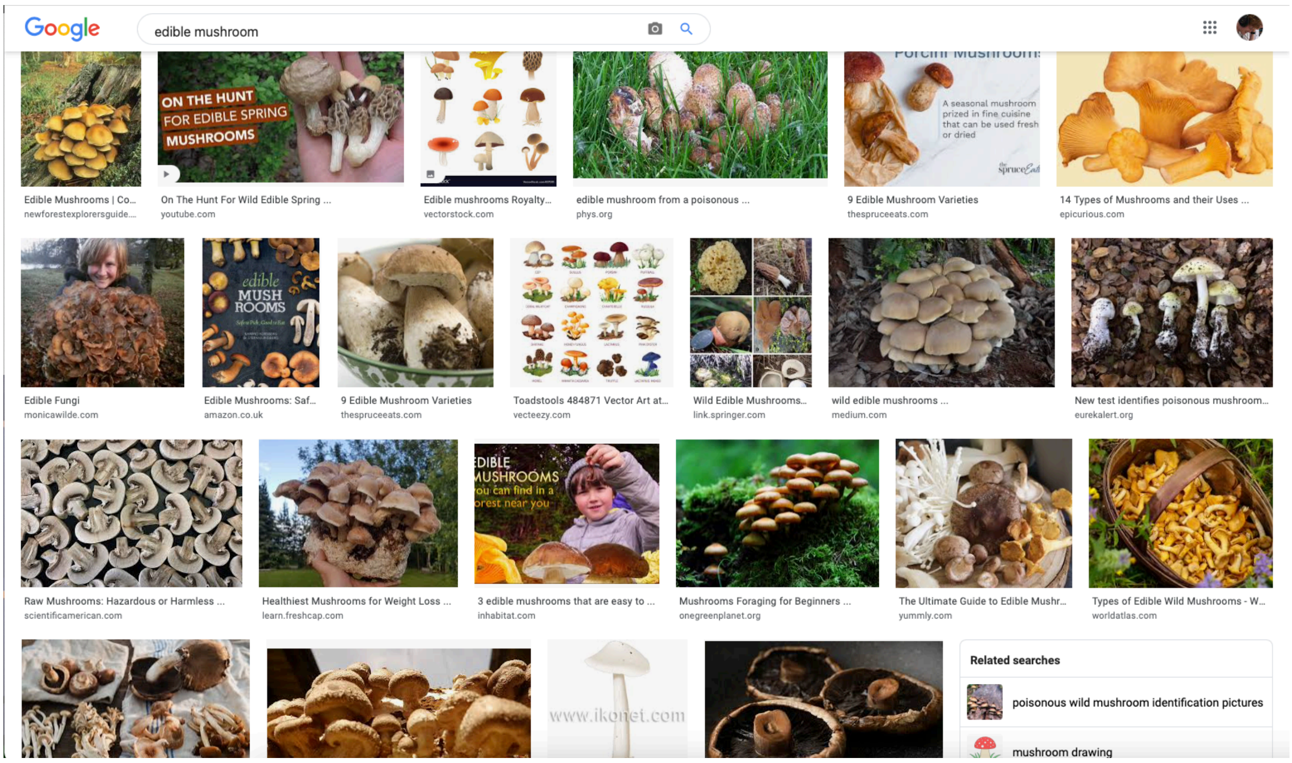

Google
edible mushroom
Edible Mushrooms | Co...
On The Hunt For Wild Edible Spring ...
Edible mushrooms Royalty...
edible mushroom from a poisonous ...
9 Edible Mushroom Varieties
14 Types of Mushrooms and their Uses ...
Edible Fungi
Edible Mushrooms: Saf...
9 Edible Mushroom Varieties
Toadstools 484871 Vector Art at...
Wild Edible Mushrooms...
wild edible mushrooms ...
New test identifies poisonous mushroom...
Raw Mushrooms: Hazardous or Harmless ...
Healthiest Mushrooms for Weight Loss ...
3 edible mushrooms that are easy to ...
Mushrooms Foraging for Beginners ...
The Ultimate Guide to Edible Mushr...
Types of Edible Wild Mushrooms - W...
Related searches
poisonous wild mushroom identification pictures
mushroom drawing

Figure 2: Search for "edible mushroom" on Google Images.

During research for how to solve our goal, we researched the possibility of classifying edibility using tabular data instead of image-data. Tabular data does have the benefit of being extremely effective in this use of binary classification, achieving an accuracy of 99.25% in a comparative 2019 study by Eyad Sameh Alkronz, et al. [4]; However, this did not meet our goal of taking an image input, and would therefore require an additional CNN to be trained to infer feature data from an input image, to then be used in the tabular data classifier.
This seemed like a risky approach, as it is easy to see from sample images of mushrooms that the majority of tabular data would be extremely difficult to predict accurately enough for classification to achieve an acceptable level of accuracy within the given time-frame.
Another potential way to identify a mushrooms edibility is to classify the specific name of the given mushroom and use the specific name to lookup edibility info. An example of this approach would be the mobile app Picture Mushroom, which uses an extremely large database of mushroom images and external servers to process classifications, this is an excellent service and provides detailed information about the species of mushroom, however, from analysis of app store reviews and personal experience, it has a tendency to miss-classify mushrooms, this is likely due

to the fact it is trying to classify by species, and obviously some mushrooms have very similar features to others. This might be okay for people who are just interested in learning about mushrooms, however our mission is to provide accurate edibility classification, so mis-classification is a serious problem. The main issue with this approach is the amount of time and resources it would take to implement. Training an accurate classifier of that size, with such a huge number of potential categories would take far too much time than the scope of this project allows for.

Final Solution

The chosen solution is to train a classifier using a split dataset of two folder categories, one containing a variety of photos of edible mushrooms and the other containing photos of poisonous mushrooms. This solution provides the greatest variety of training data for edible and poisonous classification.
For our classifier, it was decided the obvious choice was a CNN, CNNs are the most popular choice of solution for image classification due to their accuracy, being the most popular choice also means there is a great deal of support, literature and examples available to help with the production of the solution.
Although there are different classes of edibility in mushrooms, such as “edible when cooked” and “edible, but wouldn’t recommend”, we decided to go with a binary classifier due to the small amount of available data to be used in the datasets for classification of each sub-category of edible, as the primary objective of this classifier is harm prevention, meaning obtaining the highest accuracy possible is the most important aspect of our project.
Although there are readily available image datasets for identification of mushroom species to be found online, there are no datasets classifying general pictures of various species as“edible” and “poisonous”. To solve this problem, It is necessary to build an image scraper that would take a source url containing pictures of edible or poisonous mushrooms and save them into corresponding folders to be used as a dataset.
The easiest option would be to gather photos from google images using off the shelf image-downloader plugins for the browser, this approach was used by Cyrill Glockner in a towaredsdatascience.com article showcasing an outdated TensorFlow for poets model of mushroom edibility classification [5], however upon investigation, the range of images that come from google images can vary in quality and relevance, for example when searching “edible mushroom” or “edible mushroom images”, there are many illustrations and book covers returned and generally, the images would contain far too much noise and unwanted aspects to be used in the training of a classification model (see Figure 2.), which could potentially lead to issues with accuracy, furthermore there is no real way of trusting the mushrooms returned in the image search are actually going to be correctly classified, which due to the nature our goal, could be problematic and potentially dangerous.

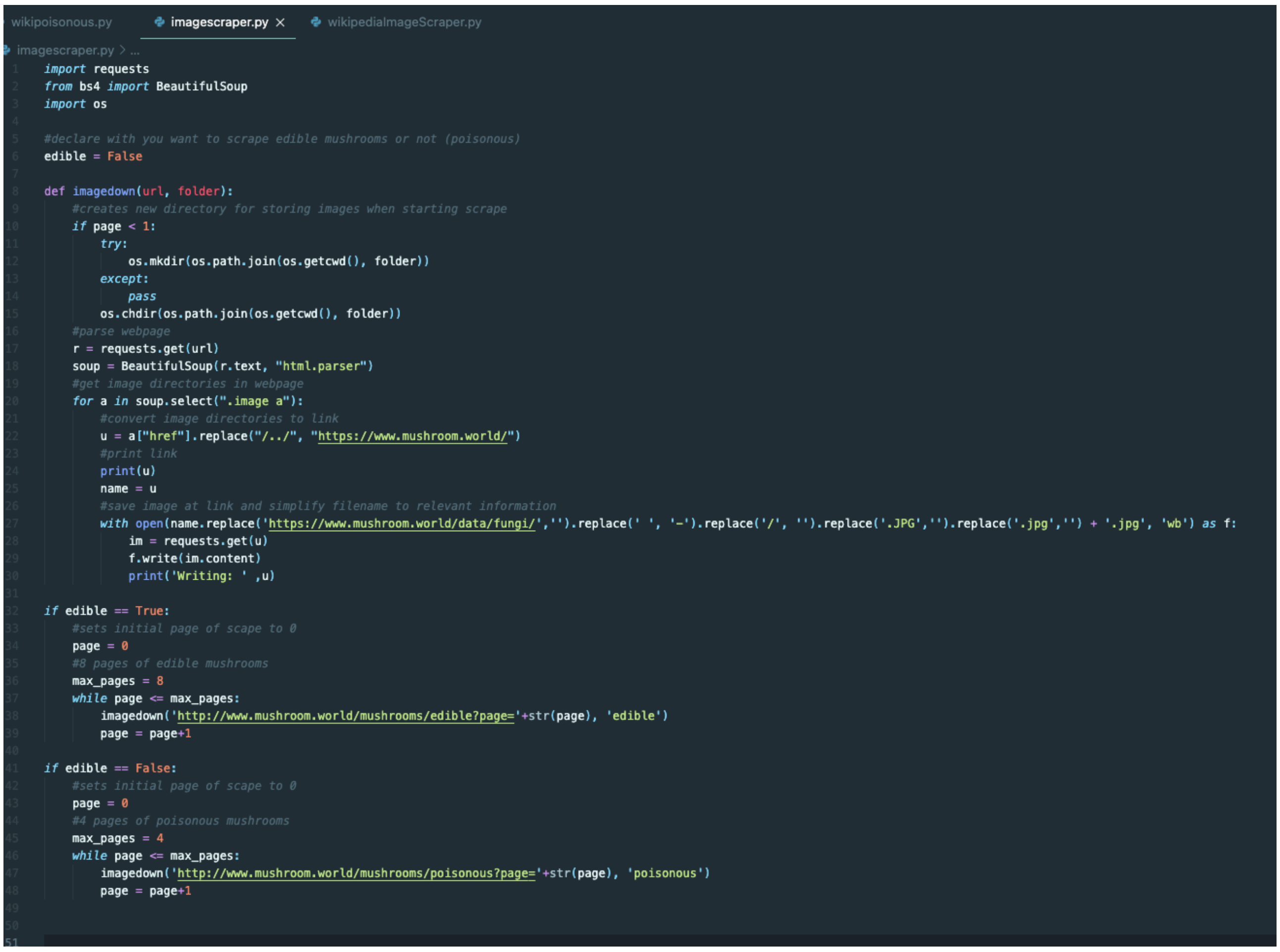
The website “www.mushroom.world" provides a large database of mushroom species, which are categorised by their edibility on separate URLs, making it perfect for the source of our images for a custom made dataset.
Using Python to scrape the web pages appears to be the most efficient method of downloading and categorising the images into their respective “edible” and “poisonous” folders required for the training of our model.

Dataset Creation

The Python library “BeautifulSoup” provides a simple way of scraping source HTML from mushroom.world, which would then provide links to webpages for each edible and poisonous mushroom species, which could then be followed by Python’s requests library and all images of the mushroom could be easily saved into a specified folder.

The scraper works by having the user declare the status of the Boolean edible.
If the boolean is set to true, the variable max_pages is set to the correct number of pages of edible mushrooms, which is then used to loop through the pages.
Inside the while loop, the function “imagedown” is called, with the correct URL for the edible mushrooms, plus the page number set to the current iteration of the loop, this way the function “imagedown” is called on every relevant page of edible mushrooms. Imagedown is also fed the folder name “edible” so the photos are saved in that folder.

The function imagedown takes the URL and the folder name as parameters.
The function then sees if the page number is < 1, to see if it is the first time the function has been ran, if so, it will create a new folder with the name “edible” or join an existing one if it already exists. The function then requests the URL of the current page and creates a BeautifulSoup text parse of the page then loops through all relevant links to full-size images, creates an absolute link from the relative path. The function then opens the images and saves them within the previously created “edible” folder, with a simplified filename.
When the variable edible is set equal to true, the imagedown function is called in the exact same way, except it is passed the URL of the poisonous mushroom directory of “mushroom.world” and images are saved into a folder called “poisonous”.



```python
import requests
from bs4 import BeautifulSoup
import os

#declare with you want to scrape edible mushrooms or not (poisonous)
edible = False

def imagedown(url, folder):
    #creates new directory for storing images when starting scrape
    if page < 1:
        try:
            os.mkdir(os.path.join(os.getcwd(), folder))
        except:
            pass
        os.chdir(os.path.join(os.getcwd(), folder))
    #parse webpage
    r = requests.get(url)
    soup = BeautifulSoup(r.text, "html.parser")
    #get image directories in webpage
    for a in soup.select(".image a"):
        #convert image directories to link
        u = a["href"].replace("/../", "https://www.mushroom.world/")
        #print link
        print(u)
        name = u
        #save image at link and simplify filename to relevant information
        with open(name.replace('https://www.mushroom.world/data/fungi/','').replace(' ', '-').replace('/', '').replace('.JPG','').replace('.jpg','') + '.jpg', 'wb') as f:
            im = requests.get(u)
            f.write(im.content)
            print('Writing: ' ,u)

if edible == True:
    #sets initial page of scape to 0
    page = 0
    #8 pages of edible mushrooms
    max_pages = 8
    while page <= max_pages:
        imagedown('http://www.mushroom.world/mushrooms/edible?page='+str(page), 'edible')
        page = page+1

if edible == False:
    #sets initial page of scape to 0
    page = 0
    #4 pages of poisonous mushrooms
    max_pages = 4
    while page <= max_pages:
        imagedown('http://www.mushroom.world/mushrooms/poisonous?page='+str(page), 'poisonous')
        page = page+1
```

Figure 3: baseline implementation image scraper code

Environment Setup

After reviewing many sources and tutorials on machine learning, it is clear that Python provides the most powerful and easy to use libraries in relation to building a CNN image classifier.
In order to easily use these libraries, a virtual environment is recommended.
Anaconda has been chosen for our baseline implementation as it provided a powerful enough virtual environment for this application.
The baseline implementation has been developed within a Python Jupyter notebook within a TensorFlow environment, Jupyter allows clear explanations for code which can be easily read and understood by people viewing the project on GitHub and more importantly for other team members during Iterative Development.

To develop our CNN model, it is necessary to import the TensorFlow library.
TensorFlow allows easy and up to date methods for loading and pre-processing data, building and training models and finally deploying the models to be used to classify new images.
Keras API has been imported from TensorFlow, which is the recommended API by TensorFlow as it allows high-level convenience and low-level flexibility, which will ultimately speed up development time and allow for better optimisation during Iterative Development.

Other packages are installed into the TensorFlow environment and imported to aid with displaying data:

- NumPy allows mathematical operations to be performed on output data from the model to make the output understandable and relevant.
- MatPlotLib allows the output data to be displayed as graphs, making it easier to compare accuracy of models throughout training.
- OpenCV provides computer vision and machine learning algorithms to be used within the training of models.
- Pillow provides image file support and processing capabilities to the Interpreter.

Pre-processing

In order to optimise the data to be used in our machine learning model, some res-scaling of the images has to be performed to make sure all images are the same size for the model.

Keras does a good job of taking care of this for us without having to manually pre-process images, a preprocessing.rescaling layer allows us to declare an input shape, so a standard set of 180x180 for dimensions will suffice for a baseline implementation, as we are more focused on speed of training than accuracy until the iterative development takes place, although research shows that decreasing resolution by a factor of two in both dimensions consistently lowers accuracy (by 15.88% on average) [6], and as most of our mushroom images found online are high resolution, it will be worth exploring increasing the image dimensions further on to try and improve accuracy.

CNN Model Description

Our classification model is a Convoluted Neural Network (CNN), which uses discrete data (mushroom image dataset) for training to classify edibility.
A Neural Network with convolution layers is preferred to a regular Neural Network for the application of image classification as it reduces the number of input nodes by essentially detecting edges of objects. The mushroom images in the dataset are highly dimensional, so a CNN's ability to reduce the number of parameters without losing quality suits the problem.
Our CNN classifier consists of:

- A rescaling layer
- 3 convolutional layers, each followed by a pooling layer
- A fully connected (dense layer)

For our model, we rescale the input to the recommended image dimensions by TensorFlow (180x180) and rescale input range from [0, 255] to [0,1], we do this to decrease the parameters and therefore reduce the requirement of computation power required by the model.
After the input has been rescaled, it is then passed into the first 2D convolutional layer. The first layer uses 16 filters, each sequential layer uses double the amount of filters as the previous one, going from 16 up to 64 filters for the 3rd convolutional layer, as noise is eliminated and the training data is filtered down to more useful patterns through each layer, more filters can be used to identify useful patterns. The kernel size is set to 3, meaning a 3x3 pixel filter will be used to scan the features and the padding on each layer is set to "same", which means the filter will not miss any of the outer features.
After each 2d convolution layer, we will use a pooling layer in order to speed up computation and make the features more robust. TensorFlow's default pool (filter) size is 2x2 pixels, which means the array of features is essentially compressed to half the size.
We will use a Rectified linear unit (ReLU) activation function to add nonlinearity for the baseline implementation as it is quick to compute and therefore ideal for the first solution just to test the concept and measure baseline performance before iterative development takes place to find more potentially effective and more mathematically intensive activation functions.
A standard flatten function layer will then be used to reshape the features into a vector.

A dense layer then takes the dot product of the input and the kernel and adds bias, and then applies the ReLU activation.
A final dense layer then outputs an array in the shape of the number of classes (2, edible and poisonous)

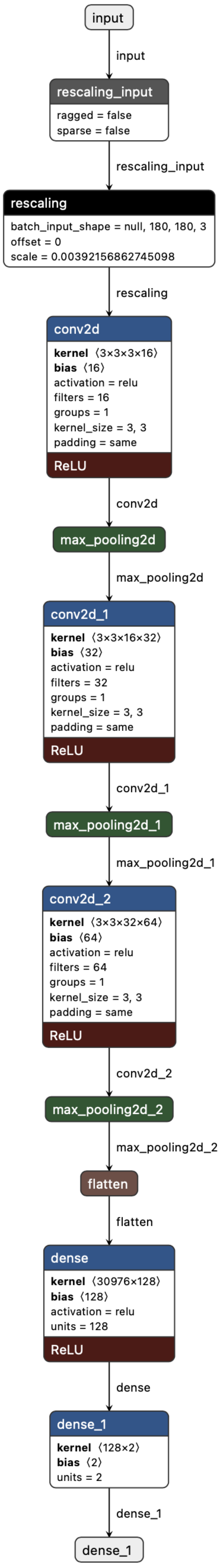

input
input
rescaling_input
ragged = false
sparse = false
rescaling_input
rescaling
batch_input_shape = null, 180, 180, 3
offset = 0
scale = 0.00392156862745098
rescaling
conv2d
kernel ⟨3×3×3×16⟩
bias ⟨16⟩
activation = relu
filters = 16
groups = 1
kernel_size = 3, 3
padding = same
ReLU
conv2d
max_pooling2d
max_pooling2d
conv2d_1
kernel ⟨3×3×16×32⟩
bias ⟨32⟩
activation = relu
filters = 32
groups = 1
kernel_size = 3, 3
padding = same
ReLU
conv2d_1
max_pooling2d_1
max_pooling2d_1
conv2d_2
kernel ⟨3×3×32×64⟩
bias ⟨64⟩
activation = relu
filters = 64
groups = 1
kernel_size = 3, 3
padding = same
ReLU
conv2d_2
max_pooling2d_2
max_pooling2d_2
flatten
flatten
dense
kernel ⟨30976×128⟩
bias ⟨128⟩
activation = relu
units = 128
ReLU
dense
dense_1
kernel ⟨128×2⟩
bias ⟨2⟩
units = 2
dense_1
dense_1

Figure 4: CNN detailed view

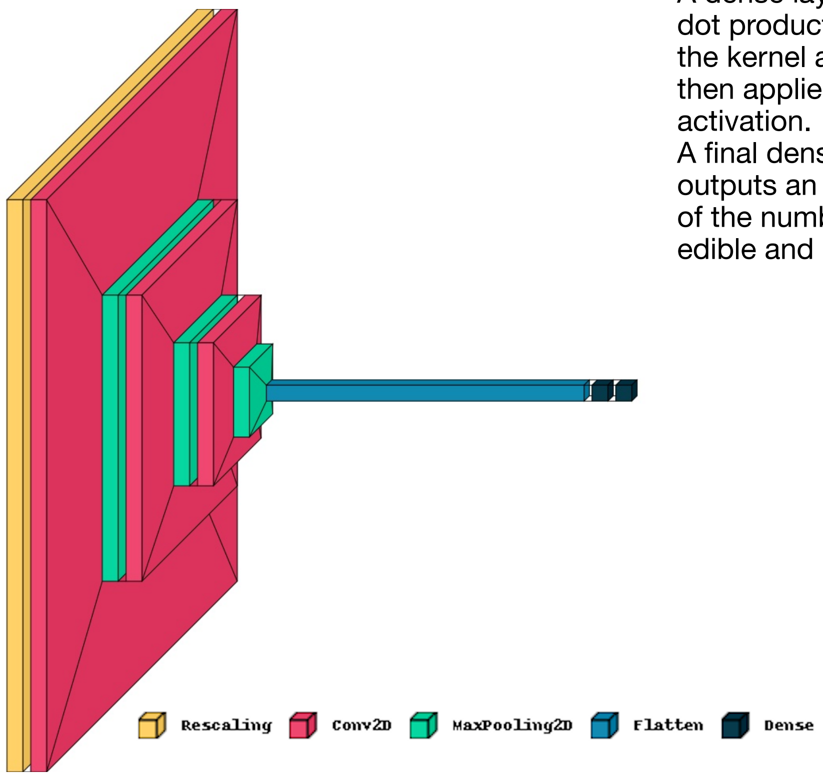


Figure 5: CNN overview

References

[1] Adegbenjo, A.E., Adedokun, M.O. and Oluwalana, S.A., 2020. Factors Influencing the Consumption of Wild and Cultivated Mushroom Species in Southwestern Nigeria. *Journal of Forest and Environmental Science*, *36*(4), pp.311-317.

[2] Patowary, B.S., 2010. Mushroom Poisoning-an overview. *Journal of college of Medical Sciences-Nepal*, *6*(2), pp.56-61.

[3] Dogo, C., 2021. Mushrooms classification - Common genus's images. [online] Kaggle.com. Available at: <https://www.kaggle.com/maysee/mushrooms-classification-common-genuss-images> [Accessed 9 February 2021].

[4] Alkronz, E.S., Moghayer, K.A., Meimeh, M., Gazzaz, M., Abu-Nasser, B.S. and Abu-Naser, S.S., 2019. Prediction of whether mushroom is edible or poisonous using back-propagation neural network.

[5] Cyrill Glockner, 2017, *Deep Learning and Poisonous Mushrooms*. [online]
Towards Data Science. 2021.
Available at: <https://towardsdatascience.com/deep-learning-and-poisonous-mushrooms-4377ea4c9b80> [Accessed 8 May 2021].

[6] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S. and Murphy, K., 2017. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7310-7311).