Image Recognition Software for a Web App

By: Heena Khan, James Phillips, Luis Chunga, Elijah Barbour, Mason Thieman, Matthew Radice, Steven Sheffey (Team ALCOVE)

Introduction

- We developed a free food-finding app during HackMT, where users could donate food by filling in relevant information about the food they are offering.
- Previously, the food donor had to manually fill this information every time they hosted food, which became tedious.
- To eliminate the form filling process, we decided to create image recognition for the app using convolutional neural networks.

Methods: Mathematical Executions

- A common issue when creating neural networks, is the amount of time it takes to train neural networks.
- We used Rectified Linear Units (ReLU) as our activation function for a few of the hidden layers in our CNN, which can be represented by the equation.
- Because of its simplicity in computations, we were able to achieve a faster training time for our neural network.

$$f(x) = \max(0, x)$$

where x is an input value

Methods: Practical Execution

- In the first attempt, we developed a model using the pretrain model MobileNet as our simple base net.
- On a small dataset, the accuracy was good, but when we implemented it with the entire dataset, it did not give us good results.
- Then we tried a couple of MobileNet models such as MobileNetV2 and NASNetMobile.
- The reason that we stick to a Mobile network is that it’s very resource consuming to train a whole massive Imagenet model.

Methods: Model Architecture

- Modern image recognition can require heavy amounts of memory and processing power. Additionally, the training of these neural networks requires large amounts of data.
- To solve this issue, for our models, we used Keras's ImageDataGenerator.
- The data generator provides methods for image pre-processing, such as resizing images, normalizing pixel values, and augmenting the dataset with transformations such as translation or rotation.
- This reduced the strain on the system from trying to load what could be thousands or hundreds of thousands of images all at once.

Results

- ➔ Due to Model 1's 22% accuracy as top 1, Models 2 and 3 had to be created.
- ➔ Although Models 2 and 3 have a higher accuracy than Model 1, it was still not identifying most of the food images.
- ➔ Since Model 1 could identify most of the images within the top 5 and 10, we deployed Model 1 into the Web app.

TABLE I
ACCURACY OF OUR MODELS, AVERAGED OVER 5 FOLDS

	Classes	Top 1	Top 5	Top 10
Model 1	101	22.23%	42.31%	52.56%
Model 2	50	34.30%	59.52%	70.99%
Model 3	25	41.37%	69.17%	80.68%

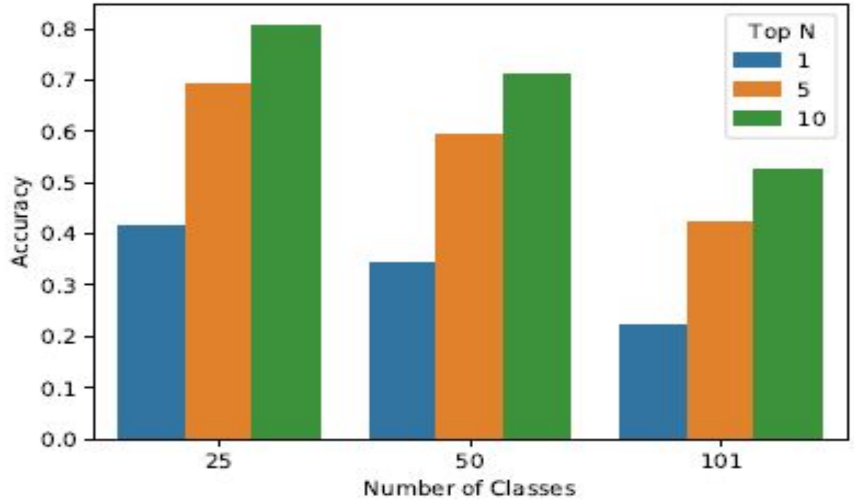

Fig. 2. Top N accuracy values across all 3 models

Results

- After generating our CNN model we trained on 101 food classes.
- The front-end web application was created using HTML, CSS, and JavaScript that accepts an image and sends it to the Keras model being hosted by Flask.
- The Keras model then returns the top 1, top 5, and top 10 predictions for the image.
- Although the models do not predict the correct food 100% of the time, the program is still able to save the user time by providing the user a list of either the top 5 or the top 10 food choices to choose from.

References

[1] Z. Dou, J. D. Ferguson, D. T. Galligan, A. M. Kelly, S. M. Finn, and R. Giegengack, "Assessing us food wastage and opportunities for reduction," Global Food Security, vol. 8, pp. 19–26, 2016.

[2] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101–mining discriminative components with random forests," in European conference on computer vision. Springer, 2014, pp. 446–461.

[3] S. Mader, "Food images (food-101)," May 2018. [Online]. Available: https://www.kaggle.com/kmader/food41

[4] Y. Zhu, Y. Chen, Z. Lu, S. J. Pan, G.-R. Xue, Y. Yu, and Q. Yang, "Heterogeneous transfer learning for image classification," in Twenty-Fifth AAAI Conference on Artificial Intelligence, 2011.

References

[5] P. Asikanius, “Predicting image social tags using a convolutional neural network,” Master’s thesis, 2018.

[6] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.

[7] “Dagnetwork.” [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/nasnetmobile.html

[8]B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning transferable architectures for scalable image recognition,” 2017.