Hotdog/not hotdog

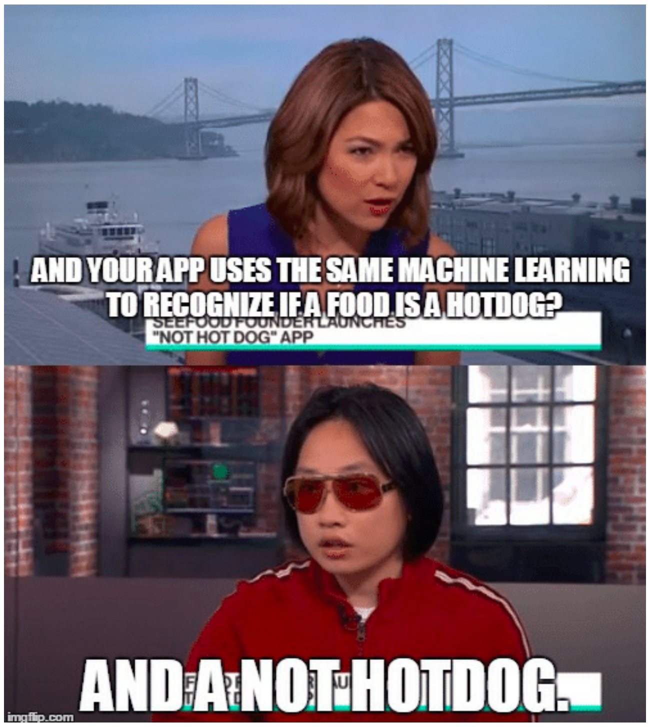
Project 1.1

Deep Learning in Computer Vision

June 2024

In this exercise you will recreate Jian-Yangs (from Silicon Valley on HBO) infamous hotdog/not hotdog algorithm.

https://www.youtube.com/watch?v=pqTntG1RXSY


AND YOUR APP USES THE SAME MACHINE LEARNING TO RECOGNIZE IF A FOOD IS A HOTDOG?
SEEFOOD FOUNDER LAUNCHES "NOT HOT DOG" APP
AND A NOT HOTDOG.
imgflip.com

The sole purpose of the algorithm is to classify images into two classes: hotdog or not hotdog.

Your task is simple; you shall create a convolutional neural network that can this do classification.

DataLoader

We have supplied you with a dataset of images containing either hotdogs or something that is not a hotdog. The images come from the ImageNet categories: pets, furniture, people, food, frankfurter, chili-dog, hotdog. The images can be found in the file `hotdog_nothotdog.zip` on Learn.

Attached to the exercise on Learn we have also given you an Jupyter notebook that contains a DataLoader that can load the images in the dataset.

The images all have different resolution and aspect ratios. Be aware that in a forward pass all images must have the same resolution. We supply you with a line of code which is the simplest way of achieving this - resizing the images to a square image (squishing the image if it is not already square). Note that this is not necessarily the best way of achieving this goal and you are free to do it however you please.

The task

Design and train a CNN to do the classification task, evaluate its performance, and document the process. For inspiration here are some questions you could answer in your report:

- How did you decide on your architecture? Did you start out with something else? How/why did you decide to change it?
- How did you train it? Which optimizer did you use? Did you compare using different optimizers? Did you do other things to improve your training?
- Did you use any data augmentation? Did you check if the data augmentation improved performance?
- Did you use batch normalization? Does it improve the performance?
- What is the accuracy of your network? Which test images are classified wrong? Any of them for obvious reasons?
- Did you use transfer learning? Does it improve the performance?
- Do you have a model that performs well? Try computing the saliency maps for some images of hotdogs.

- Compute the smoothgrad saliency map and plot it for some example images. Make sure you can you explain how adding Gaussian noise to an image is equivalent to drawing a sample from a normal distribution centered at that image. Do your saliency maps make sense? Remember that the variance of the normal distribution is a hyperparameter that can be tweaked.
- Did you use ChatGPT or similar generative AI tools? If yes, please include a paragraph where you briefly describe how you used it and how they were useful.

Hand-in

Your process, performance evaluation and results should be documented and discussed in a PDF poster to be uploaded on DTU Learn.