Endgame

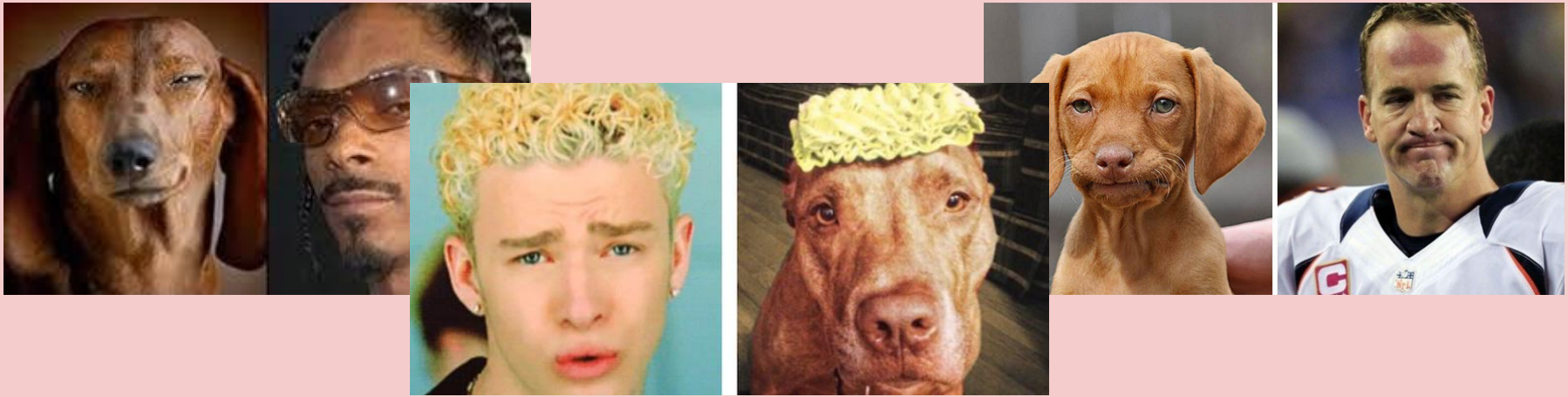
Exploration of Machine Learning Image Recognition by:

Claudia, Grant, Lars, Mrudula, and Nisha

2. Purpose

Long discussion; one common idea.

Match Celebrities with their Dog Breed Doppelganger through a Dog-Person face recognizer



3. Purpose Discussion

What is the difference between a well-made dog-person analyzer and a random number selector?

A dog-identifier can successfully identify dog breeds

We need a well trained dog image identifier that will only return dog breeds and then apply to images of humans.

4. Similar projects

Consensus accuracy goal: 40% correctness on dog breed is good

Higher levels of correctness for complex image recognition might require more complex models.

It turns out that Xception is a well trained image classifier...

“**Xception** is a convolutional neural network that is 71 layers deep. You can load a pretrained version of the network trained on more than a million **images** from the ImageNet database [1]. The pretrained network can classify **images** into 1000 object categories, such as keyboard, mouse, pencil, and many animals.”

4.5 Similar project discussion

So if we make a model that is very accurate (75%+), it's probably overtrained…

...or really good

Let's try xception

predict(image_path)
Predicted: [('n096048021', 'rad_shades', 0.99999999),
('n03452741', 'grand_piano', 0.8251691),
('n02804610', 'bassoon', 0.028958963),
('n03838899', 'oboe', 0.017797884)]

Stanford Dog dataset

The Stanford Dogs dataset contains images of 120 breeds of dogs from around the world. This dataset has been built using images and annotation from ImageNet for the task of fine-grained image categorization. Contents of this dataset:

- **Number of categories:** 120
- **Number of images:** 20,580
- **Annotations:** Class labels, Bounding boxes

Which celebrity are you? Face Recognition using Deep Learning

In this project we will show how can you use deep Learning convolutional Neural Network to scan the face features of celebrities and compare it with the dog database

Steps followed

1. Load in an image
2. Extract the face out of the image
3. Perform preprocessing so that the face image can be recognized
4. Load the image in deep neural network and extract the results

Face Recognition is a concept of identifying people(faces) from Images

Convolutional Neural Network-

The process goes as follows, we feed the network a face image. This face image will then be fed to multiple layers (the convolutional base). In the first layers, the basic features are detected: edges, corners,…. The middle layers detects parts of objects. In our case, they might learn to detect: eyes, noses, ears,… . In the last layers, they learn to recognize full objects, in different shapes and positions. Then based on all these detected features, the classifier makes a prediction.

5. Data selection

NO Dress→

NO Dress != Anne Hathaway

The data you select is fundamentally your project

Agility is diminished in big-data apps due to data quality and quantity concerns

We have a dataset of celebrity photos from kagle. This is Anne Hathaway

Dress

Dress

Dress

Dress

Dress
6/7 total

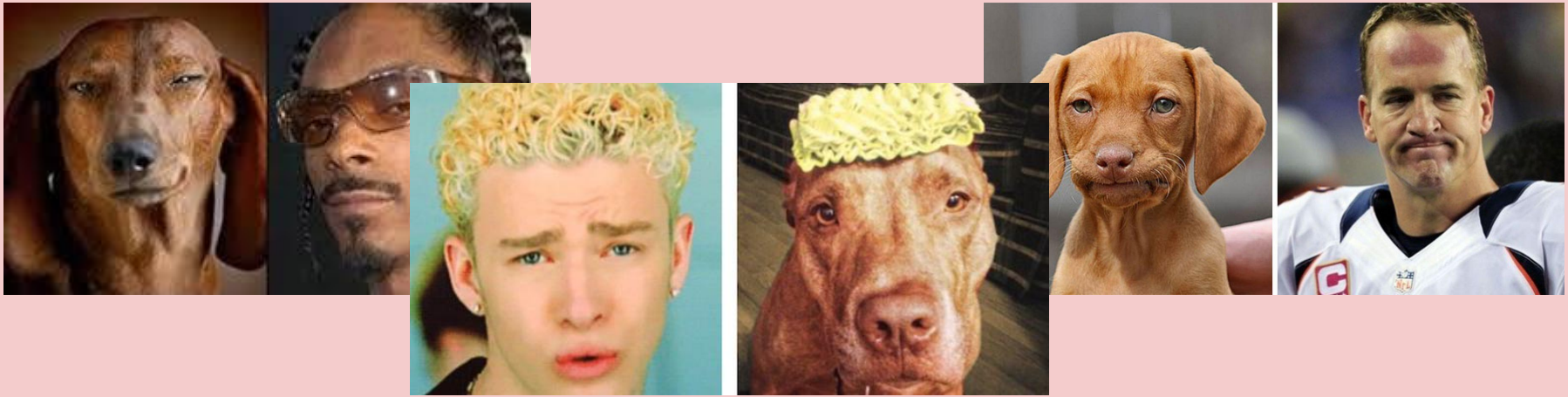
Anne Hathaway does not have legs.

So that is an entire celebrity entry in our dataset. It is nowhere near large enough to train a image recognition model that would be effective.

Model Selection: Does this fulfill our purpose?

Perhaps a model trained from a small set of funny dog photos would be best?

Without this specialty dataset, we had no option but to move forward in creation of our own model that ONLY identifies dog breeds when presented an image.



Training our own model

Before we show our notebook, take some time to appreciate the balance between pretrained-model selection and the from-scratch novel training of a few model for an application.

That’s when we found a third option in *transfer learning*. This was a better balance for us. We created our own cnn and ran it through one of the pretrained models.

→ → → Explore our Jupyter Notebook → → →