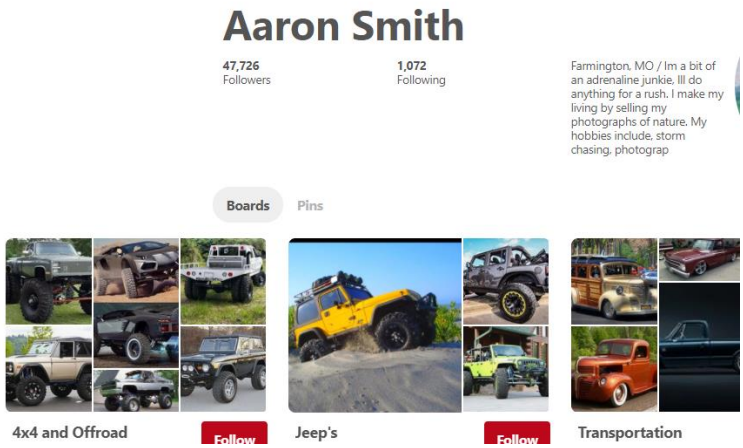
Information Retrieval using Deep Learning and Natural Language Processing on Pinterest Dataset

Introduction:

The proposed system will recommend the user and boards with similar images using transfer learning and fine tuning. It also recommends user/boards based on text classification implemented using nltk and TensorFlow. The experiment is performed on the data collected from Pinterest website. Pinterest is a web and mobile application company that operates a software system designed to discover information on the World Wide Web, mainly using images and on a shorter scale, GIFs and videos. Pinterest has reached 200 million monthly active users as of September of 2017.

Typical User profile would look as shown below:



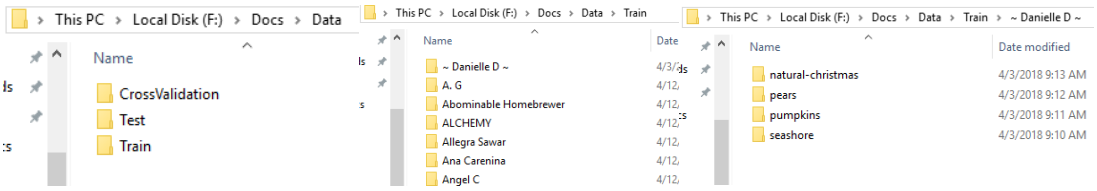
Boards are the folders/categories under which user pins his images.

Dataset:

The original idea was to collect the images user-board wise from the Pinterest website. Data collection was one of the major challenges of the project. The reason being, the board names and the images in the board are unrelated most of the time and it was humanly not possible to validate and download each required image from the website. Most of the data was downloading by the crawler based on intuitions and basic verifications. User list was crawled from the Pinterest website. Users were chosen such that, in total dataset more than 50% of the data matched to ImageNet dataset. Interactive program was designed such that, for each user all the board names were displayed and up to 4 board names was given as input. Below Screenshot shows how the program displays user boards for a user in the user queue awaiting input.

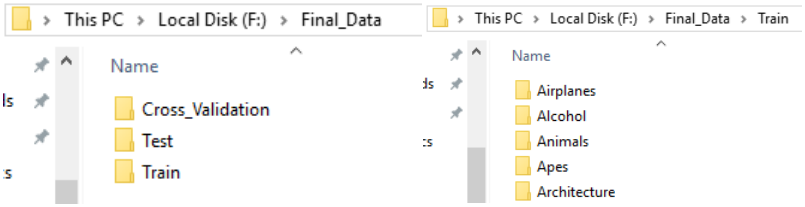
```
F:\Documents\Academics\ADL\Project>python crawl.py
More Tap the link for an awesome selection cat and kitten products for your feline companion!
{1: 'big-cats', 2: 'tattoo-ideas', 3: 'minion-quotes', 4: 'our-countryside', 5: 'diy-ideas', 6: 'recipes', 7: 'funny-sayings', 8: 'books', 9: 'hunky-men', 10: 'other-an
imals', 11: 'horses', 12: 'gardening', 13: 'crafts', 14: 'cats-and-dogs'}
Select Board Names seperated with ','
```

The original dataset had 48794 Images extracted from 200 user profiles with total of 791(classes)boards. The data was divided into Train, validation and test tests. Below is the structure.



The initial idea was to classify images among each of the user boards, for e.g.: from above figures, '~ Danielle D ~\pears' would be one class, so there were total of 791 classes. The initial assumption was that each of the user board would contain considerable number of unique images. However, since the images are not very specific to the boards and number of samples per class was evidently less, the whole data was restructured after testing classification accuracy which was less than 30%.

The new structure of the modified dataset is as below:



To modify the data, I used all the unique board names from the original dataset and assigned manually to the newly created classes based on the observable relationship. Consider below example: For the newly created class "Camping", the related boards that were assigned were, "camping-pillows, sleeping-bag-accessories, sleeping-bags-bedding, sleeping-bags" present in the original dataset. This modified data was structurally skewed as some of the classes had 700 – 1000 images and some of the classes had lesser than 100 images. To avoid high bias images were deprecated from the larger classes and some images were added to the classes having less number of images. So that, Final data has 135 classes and around 200 images for training per class.

Text classification dataset was created by adding 60 sentences for 55 chosen classes. The sentences were crawled from the "http://sentence.yourdictionary.com" website.

Second dataset for description-based classification was prepared by adding around 20 – 30 sentences for 8 classes. E.g. for class elephants, description would be, "has trunks and tusks", "loves water" etc.

Limitations of the Project due to dataset:

- The images do not generalize the classes well as there are irrelevant images.
- Image quality varies, as there are some HD images which has good features while others are low resolution.
- Limited relevant images for some classes.
- Text corpus was small and had more classes, due to this high-level classification models such as CNN, LSTM could not be experimented with.
- General descriptions for each class are very hard to come by.

Goals of the project:

- Retrieving similar images/users based on the image classification.

- Comparing model performance on image classification.
- Retrieving images/users based on text classification.
- Experiment on image recognition with the descriptions.

Expected Results:

- For any given image, retrieving user/board names and sample images from the related user/boards based on the classification task
- For any given sentence, classifying the sentence among trained classes and retrieving user name, board and sample images corresponding to the class.
- Comparing accuracy of the trained models. Considering the discrepancy and the size of the data, expected accuracy is somewhere between 60 %- 70%
- Validating the results based on the image being tested and retrieved.

Proposed Solution:

- Predictive model using resnet50 with ImageNet weights. To predict a test image for comparison with the trained model.
- Training a model using boards as categories, dividing the dataset into test train and validation sets. Using transfer learning comparing the accuracy with pretrained models (Inception, Xception and VGG 19)
- Selecting the classes from trained set, assigning each class with corpus, e.g.: “Running Shoe”:[“These shoes fit me well”, ”Nike manufactures some of the best running shoes”, ”I like red color shoes”]. Training this corpus using NLTK and TensorFlow

Methods Followed:

Image Classification

Transfer Learning: Transfer learning is a machine learning technique that focuses on repurposing learned classifiers for new tasks. In transfer learning for CNNs, a base network is trained on a base dataset to create weights and features. This classifier is then transferred to a new dataset by retraining a subset of the base network’s learned weights and features. The overall effect is a classifier that fits the new dataset with significantly less work than retraining a new network. When the target dataset is significantly smaller than the base dataset, transfer learning can be a powerful tool to enable training a large target network while minimizing overfitting. In certain tasks, transfer learning has been shown to achieve near state of the art results.

Fine Tuning: Fine-Tuning of pretrained networks to new tasks can be viewed as a special case of continuous learning with only two-time steps: one initial learning step and one update step. A variety of publications underlines the benefits which arise from pre-training deep networks on large datasets. As an example, Agarwal et al. stated that “pre-training significantly improves performance” for the task of object recognition. Similarly, Girshick et al. draw the conclusion that “We conjecture that the ‘supervised pre-training/domain specific fine-tuning’ paradigm will be highly effective for a variety of data-scarce vision problems.”. Further benefits have been reported for image retrieval, semantic segmentation, fine-grained recognition, or object localization. However, fine-tuning is only used with a fixed dataset of a new task.

Data Augmentation: Deep networks need large amount of training data to achieve reliable performance. To build a powerful image classifier using very little training data, image augmentation is usually required to boost the performance of deep networks**.** Image augmentation artificially creates training images through diverse ways of processing or combination of multiple processing, such as random rotation, shifts, shear and flips, etc.

Stemming: Stemming is a process applied to a single word to derive its root. Many words that are being used in a sentence are often inflected or derived. To standardize our process, we would like to stem such words and end up with only root words. For example, a stemmer will convert the following words "walking", "walked", "walker" to its root word "walk".

Tokenization: Tokens are basically words. This is a process of taking in a piece of text and find out all the unique words in the text. We would get a list of words in the text as the output of tokens. For example, for the sentence "Python NLP is just going great" we have the token list ["Python", "NLP", ïs", "just", "going", "great"]. So, as you can see, tokenization involves breaking up the text into words.

Bag of Words: The Bag of Words model in Text Processing is the process of creating a unique list of words. This model is used as a tool for feature generation. E.g.: consider two sentences:
Star Wars is better than Star Trek.
Star Trek isn't as good as Star Wars.
For the above two sentences, the bag of words will be: ["Star", "Wars", "Trek", "better", "good", "isn't", "is", "as"]. The position of each word in the list is hence fixed. Now, to construct a feature for classification from a sentence, we use a binary array (an array where each element can either be 1 or 0).

Experimented Networks:

- CNN: A plain vanilla CNN with 3 conv layers , maxpool ,dense and fully connected layer.
- Used pretrained Resnet 50 to predict the test image to validate and compare the image predicted by trained models.
- Trained data after augmenting using transfer learning and finetuning with pretrained Inception V3, Xception and VGG 16.

Network	Transfer Learning	Finetuning	Optimizers (Transfer Learning)	Optimizers(Fine Tuning)
Xception	Compiled the model by freezing all the layer and adding last layer to the convnet to fit the train data	Trained the data by on the last block by freezing all other blocks (Froze 172 layers for inception, 126 for Xception and 15 for VGG16)	nadam	nadam
Inception			rmsprop	SGD(with learning rate 0.0001 and momentum 0.9
VGG16			rmsprop	SGD(with learning rate 0.0001 and momentum 0.9

- For Text Classification, I used NLTK's text preprocessing steps. I used Lancaster stemmer, tokenization and bag of words for text pre-processing
- Used tflearn's DNN to train the created corpus. DNN uses the MLP mechanism which is suitable for the created corpus with 54 classes and 3200 sentences and description-based data.

Frameworks used:

- Beautiful soup 4 – To get webpage data and parse the images field and download images using urllib and selenium web drivers
- Keras – All the created networks and hyper parameters were imported from keras package running on tensorflow background
- Tflearn and Tensorflow along with nltk – text classification.

Results:

After training for considerable number of epochs, below table summarizes the result. (I used AWS EC2 instances to train the data and each epoch for all the networks took an average of 30 minutes)

Network	Training Loss	Training ACC	Validation Loss	Validation ACC	Epochs
CNN	3.8952	18%	3.95	17%	50
Xception(transfer learning)	1.2781	67.76%	1.5232	63%	2
Xception(fine tuning)	0.3089	92%	1.9246	67%	10
Inception (transfer leaning)	1.26	68%	1.8	56%	3
Inception (fine tuning)	0.8	79%	1.5	65%	15
VGG16(transfer learning)	2.6049	38%	2.9	34%	3
VGG16(fine learning)	1.6	60%	2.14	51%	3

(The screenshots of training accuracies are submitted in the result folder) Among all the above trained models, Inception (Finetuning) yielding test accuracy of around 65%. Below Screenshots shows the final out of the project, validating expected results:

Input:

For the input Resnet50 predicts the given image

```
Loaded Resnet50 predicted the input image as brown_bear with 97.81% confidence
```

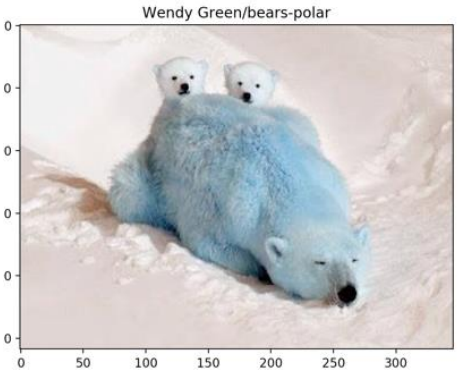
Fine tuned inception model predicts the given image

```
Fine Tuned Inception model predicted the input image as Bears with 98.93% confidence
```

For predicted class, related users and boards are displayed

```
Now displaying usernames and boards with similar images

=====================================================
Username                                Boards
=====================================================
Mybig Johnson                           da-bears
Pinterest Link https://www.pinterest.com/Mybig Johnson/da-bears

Oceanwide Expeditions                           polar-bears
Pinterest Link https://www.pinterest.com/Oceanwide Expeditions/polar-bears

Pat Barrows                             bear
Pinterest Link https://www.pinterest.com/Pat Barrows/bear

Sherri                          animals-bear
Pinterest Link https://www.pinterest.com/Sherri/animals-bear

Spirits Lemur                           south-america-kinkajou-honey-bear
Pinterest Link https://www.pinterest.com/Spirits Lemur/south-america-kinkajou-honey-bear

Tim Frock                               bears
Pinterest Link https://www.pinterest.com/Tim Frock/bears

Wendy Green                             bears-black-brown-grizzley
Pinterest Link https://www.pinterest.com/Wendy Green/bears-black-brown-grizzley

Wendy Green                             bears-polar
Pinterest Link https://www.pinterest.com/Wendy Green/bears-polar
```

Sample images are displayed from each of the user/boards



For any typed text, the model classifies the text and user/board names will be displayed as shown above.

```
Enter your text for classification

These butterflies are very colourful

 Now Loading model for text classification


Text Classification Model predicted the input Sentence as Butterflies with 71.68% confidence


Now displaying usernames and boards with similar images

=====================================================
Username                                Boards
=====================================================
B.J. Laird                              butterflies-and-dragonflies
Pinterest Link https://www.pinterest.com/B.J. Laird/butterflies-and-dragonflies
```

For entered description, the program displays the classes close to the entered description(here the classes are displayed with plus or minus 0.2 confidence of the highest confidence value)

```
Enter your description for classification

Mothers of these species are very caring and affectionate

 Now Loading model for decription classification


The description has been classified among below class(es)
Quadruped Mammal 17.2
Cats 11.65
Elephant 11.57
Lions 10.61
Bears 12.97
Monkeys 25.81

Do you want to enter more specific description?(yes/no)
```

If there Is more than one class, then the program asks for specific description.

```
yes

Enter your description for classification

These mammals have trunks and tusks

 Now Loading model for decription classification


The description has been classified among below class(es)
Elephant 81.58

Now displaying usernames and boards with similar images

=====================================================
Username                                Boards
=====================================================
Gigi Stoll                              elephant-envy
Pinterest Link https://www.pinterest.com/Gigi Stoll/elephant-envy

Lacey                           elephants
Pinterest Link https://www.pinterest.com/Lacey/elephants

Linda Rommelaere                                elephants-are-in-my-heart
Pinterest Link https://www.pinterest.com/Linda Rommelaere/elephants-are-in-my-heart

=====================================================

Do you Want to view the Sample images from above userboard(s)?(yes/no)
```

The output validates the expected results. The goals of the project are achieved with overall of 65% accuracy in image classification and high accuracy for limited texts in text classification.

Future Works:

- The system can be scaled to predict a part of whole of images

- Using substantial number of images and sentences to describe a class, system can be scaled to predict images according to the exact description, which can be used to identify people/places etc.
- Various network models can be validated.

References:

- https://machinelearningmastery.com/
- https://www.datacamp.com/community/tutorials/transfer-learning
- http://cs231n.github.io/transfer-learning/
- http://hera.inf-cv.uni-jena.de:6680/pdf/Kaeding16_FDN.pdf
- http://ceur-ws.org/Vol-1391/121-CR.pdf
- https://github.com/brightmart/text_classification
- http://www.wildml.com/2015/12/implementing-a-cnn-for-text-classification-in-tensorflow/