NDSC Report - Uncles.AI

1. Executive Summary / Abstract

The challenge in this project was to extract product-related information from a large volume of images and free text data. This project attempts to use product images to generate free text 'captions' that can be fed into an NLP model to predict product attributes. Using this model, we achieved a score of 0.51

2. Overview of Innovative Approach

We observed 2 main characteristics of the data:

1) Free text data had a higher signal-to-noise ratio than the image data
2) Free text data was short and potentially contained insufficient/erroneous information

To increase this signal, we attempted to extract additional free text data from image data using the following architecture:

- An *Object Detector* to improve the signal-to-noise ratio contained in the images
- A *Convolutional Neural Network (CNN)* as an encoder to extract image feature
- A *Long Short Term Memory (LSTM)* as a decoder to generate image captions
- The combination encoder/decoder will utilize *Transfer Learning* to generate the captions
- Captions generated will be combined together with the original free text data and run through another *LSTM Neural Network* to generate predictions

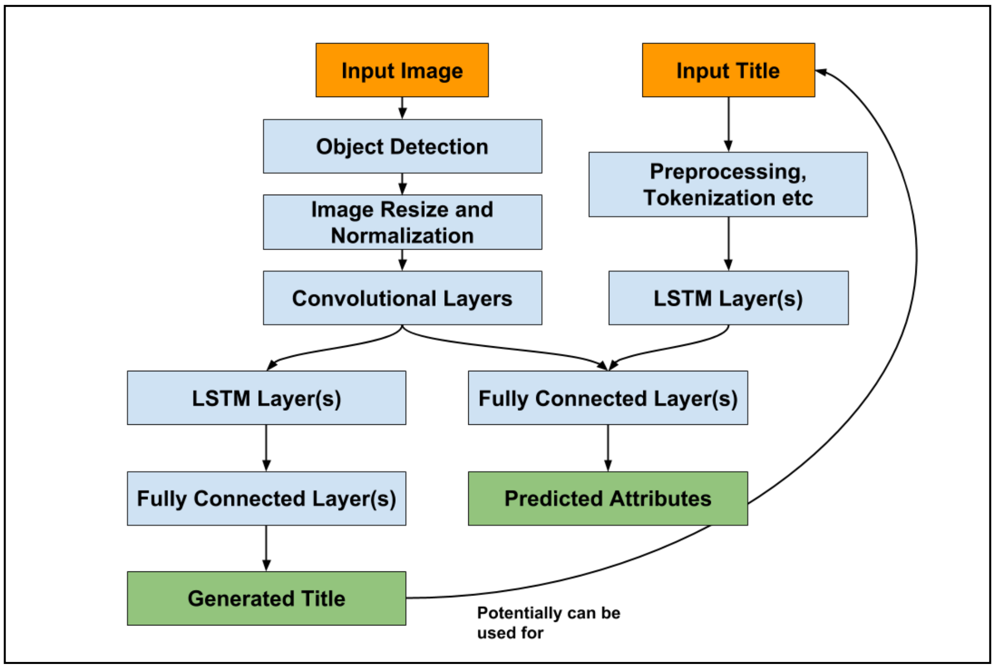


Diagram 1: Overview view of our proposed architecture

3. NLP

3.1. Formulation / Theory / Concept

It quickly became evident that the title text was one of the best indicators of a product attribute, as terms directly corresponding to a category could be found in the text. To utilize this indicator, we processed the text into features for a classification model. We also chose baseline models that would be scalable i.e. parallelizable for faster iteration.

3.2. Preprocessing

Relatively little cleaning was required because the data was already provided in a relatively clean format, e.g. all lowercase. However, a large amount of free text data as not in English. Here, we did not restrict vocabulary size in order to account for duplication of words in different languages.

The text was converted into a bag of words and normalized via tf-idf. We generated n-grams in the ranges 1 through 3 to capture terms like ‘3 4 sleeve’ that indicated ¾ sleeve length.

For each attribute in each category, the items with ‘nan’ as the attribute category were masked from the training set.

3.3. Algorithm

Once the text was processed into their tf-idf arrays, we applied baseline classification methods on the arrays, including logistic regression, decision trees and random forest models. We performed a 3-fold cross validation split as the validation strategy. The models were chosen for their success in previous text classification tasks, as well as the ease to which they parallelize.

3.4. Results and Discussion

Following these results, and a linear logistic classifier, we obtained a score of 0.508 (0.46082 on the new validation set).

The text classification tasks proved to be the most useful in terms of achieving a high MAP@2 on the leaderboard. Increasing the size of the vocabulary considered was very helpful in improving scores, which hinted that many of the rare words that appeared in the description were useful.

An interesting note was that the validation accuracy during training was much higher than the actual results on the leaderboard. This was indicative that many of the answers in the validation dataset could be on categories that rarely or did not appear in the training dataset.

4. CNN

4.1. Formulation / Theory / Concept

We used a ResNet18 model, pre-trained on the ImageNet dataset as the base model, to attempt to directly predict product attributes. This was done in order to utilize transfer learning to reduce the amount of time needed for training.

4.2. Preprocessing

The following processes were applied to input images for the CNN:

- Random cropping of images to size of 224 x 224 (The recommended image size for ResNet)
- Horizontal flipping done at random (some images flipped horizontally while others were not)
- Converting the image to a PyTorch tensor
- Normalizing the image tensor values

The random processes are executed during each epoch of training, which would give the effect of data augmentation.

4.3. Algorithm

A separate CNN model was trained for each category (beauty, fashion and mobile). Each model takes in an image tensor as the input, runs the tensor through the pre-trained base model, and gives multiple softmax outputs corresponding to the attributes associated with the category (e.g. colour group, skin type etc for beauty). Each model was trained for 3 epochs.

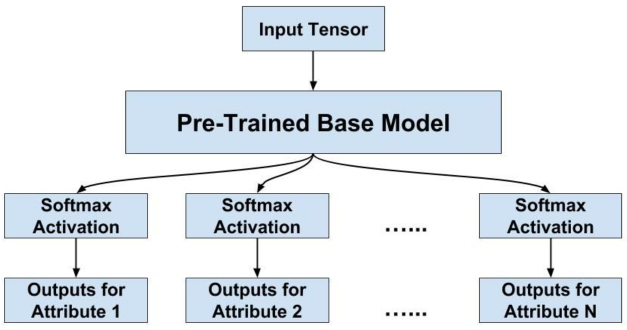


Diagram 2: Architecture of baseline CNN model

4.4. Results and Discussion

The submission score was about 11% for the baseline CNN models, which is understandable considering that the models were only trained for 3 epochs each. Training with more epochs could improve the score, but we hoped that decreasing the noise contained in the image data might give better results.

5. Object Detection

5.1. Motivation of doing object detection

We observed that there were images which contain objects/persons rather than just the product itself. We believed this was causing the CNN model to treat the extra objects/persons as features, which gives undesirable results during inference. We thus attempted to use object detection to filter out irrelevant objects in the images.

Eyeliner pertama dengan ink formula!* Garis hitam intens, kuas stabil, tahan hingga 24 jam.
BARU
HYPER ink
LIQUID LINER
MAYBELLINE
MAKE IT HAPPEN
NEW YORK

Diagram 3: Example of an image with many irrelevant objects

5.2. Formulation / Theory / Concept (if any)

Object detection consists of three main steps:

- Generation of multiple proposed bounding boxes (priors)
- Feature extraction within the priors
- Perform regression for bounding box coordinates and classification for object class

A Single Shot Detector (SSD) architecture was used for object detection to decrease compute time.

5.3. Preprocessing

A sample of 500 images from each category was randomly selected for training and validation of the object detection model. For each category of samples, sampling with replacement (bootstrapping) was done to get the training set, while the out-of-bag images (images not chosen during bootstrapping) served as the validation set.

The following processes were adapted during preprocessing, and again during each training epoch to augment the data:

- Random photometric distortion
- Random zooming out and cropping of image
- Random horizontal flipping of image
- Resizing image to 300 x 300
- Converting the image to PyTorch tensor
- Normalizing the image tensor values

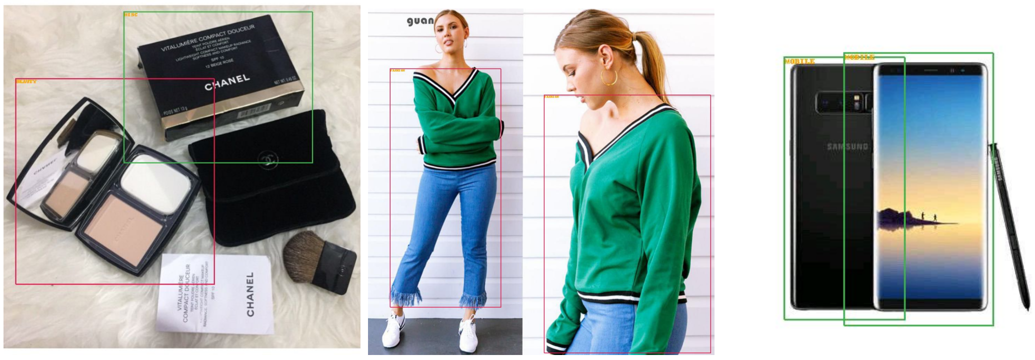
.

5.4. Algorithm

The SSD300 algorithm (Single Shot Detector using 300 x 300 images) was adapted to perform object detection. The loss function used for training the model, known as the MultiBox Loss, is a combination of a loss related to the predicted coordinates of the bounding boxes and the loss related to the predicted object class.

5.5. Results and Discussion

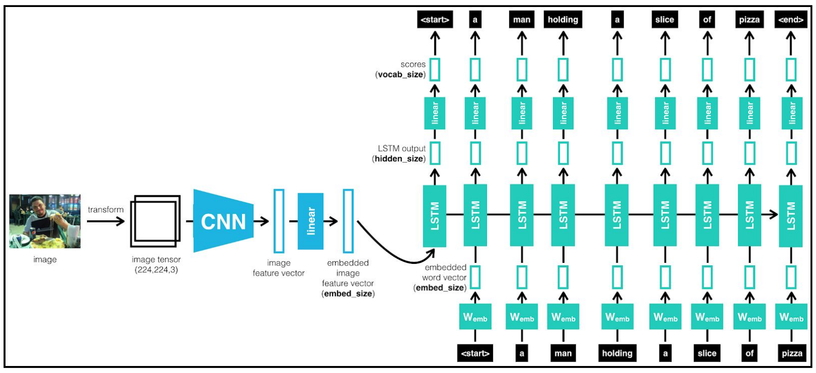
Visualizing some of the images with the predicted bounding boxes shows the potential in using object detection to filter out the irrelevant content (noise) in the images.



Diagrams 4a -4 c: Results of Object Detection

In terms of predicting the object class of each box, the average precision is 75% for beauty products, 90% for fashion products and 83% for mobile products. The bounding box coordinates obtained via inference could be used to perform image cropping, and the resultant image could be used for training or inference.

6. Image Captioning Architecture



The segmented images from earlier step are pre-processed by resizing into 224 x 224 and normalised. They are then fed into a ResNet151 model pre-trained on the Microsoft COCO dataset to extract image features. The output is then flattened to a vector and passed through a new fully-connected layer, and then an LSTM to generate image captions.

Diagram 5: Image Captioning Architecture

7. Conclusions / Reflections

In view of time constraint, the proposed approaches were performed and evaluated separately. We believe that providing a boosted random forest model or a deep learning model with a large volume of free text data provides the highest chance of success in predicting product attributes. Moving forward, we plan to combine all the models discussed together to produce an integrated system as illustrated in Diagram 1.

Beyond features extraction, we believe the same image captioning technique could have an important commercial application in *assisting sellers in describing their items*. Whenever sellers upload their photos, the captions generated could suggest keywords and item description for the item. This seller assistant feature could *improve sellers' performance by suggesting appropriate keywords to use in their item's title and description*, thereby ensuring their items are searchable by buyers, and contributing to an improvement in Shopee's Gross Merchandise Volume.