|  |
| --- |
|  |
| Algorithm to identify closest matching product |
|  |
| Based on wooplr problem statement |

**Bits pilani**

February 2, 2016

Authored by: Rishabh Agarwal Jain

Email: [f2013508@pilani.bits-pilani.ac.in](mailto:f2013508@pilani.bits-pilani.ac.in)

[Rishab.mps@gmail.com](mailto:Rishab.mps@gmail.com)

# Problem Statement:

Build a database of products from at least 3 different fashionable stores for women. Design an algorithm to analyse a look and identify the closest matching products from your pool of products.

# Solution (Approach used):

* Databases built:

1. Koovs.com – Dresses,tops,jeans
2. Forever21 – Jackets,jumpsuits,tops,dresses
3. Vajor.com – Tops,Dresses,leggings.
4. Fabally.com – Various party dresses and others.
5. Chumbak – Many trademark dresses of chumbak.

* Method used:

I decided to use Deep learning to extract features and identify the products from images.

* Testing:

Extensive testing on various Images and on Images provided by wooplr.

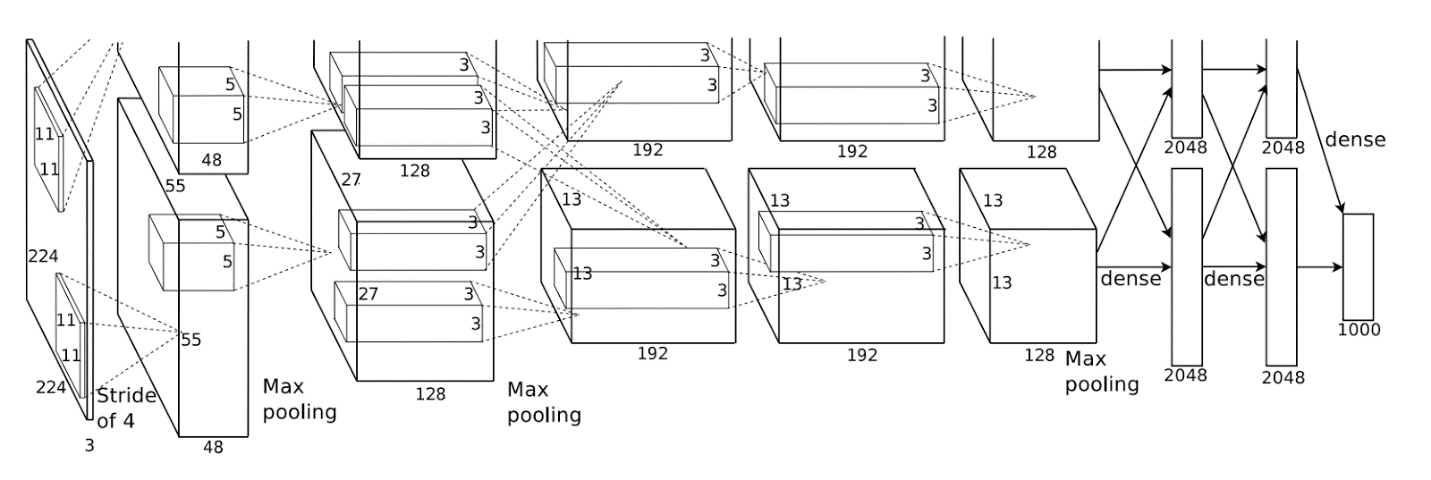
# Detail about deep learning model is provided from the next page

# Steps involved:

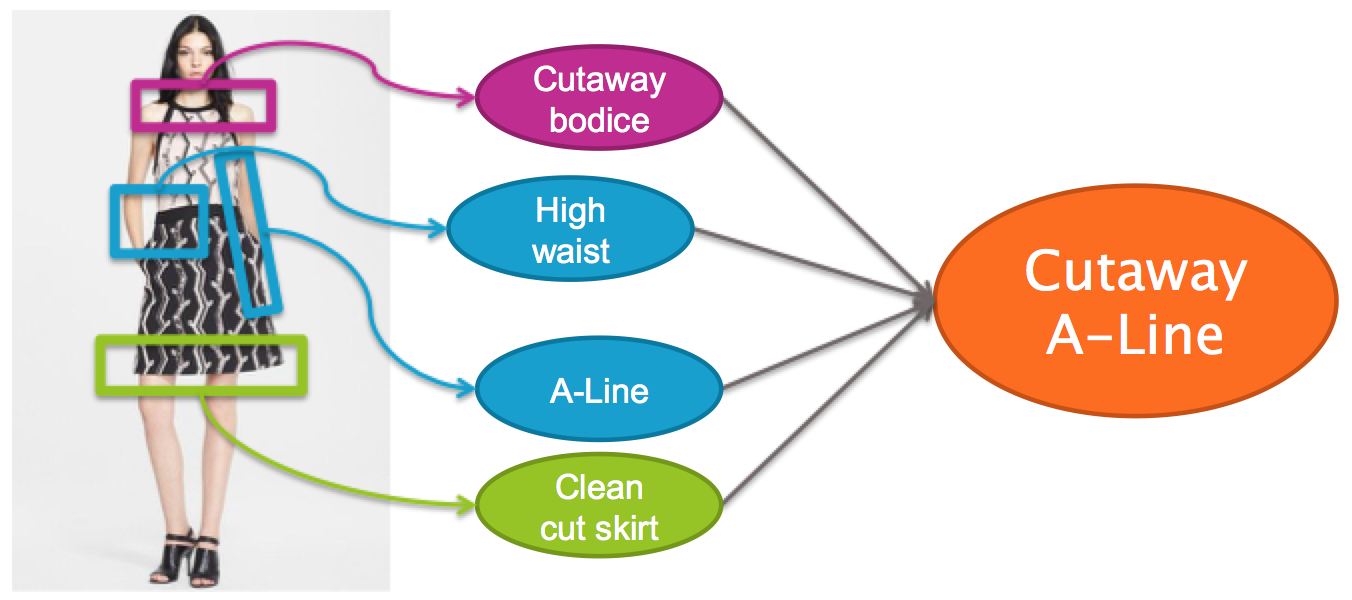
1. Loading the Data - loading existing data into a SFrame.
2. Extracting the features - loading an existing pre-trained model into a model object and using it to extract features.
3. Calculating the Distance - Creating and training a k-nearest neighbour’s classification model on the extracted features.
4. Finding Similar Items - Using the k-nearest neighbour’s model to help us find similar items.

# Deep Learning Model used:

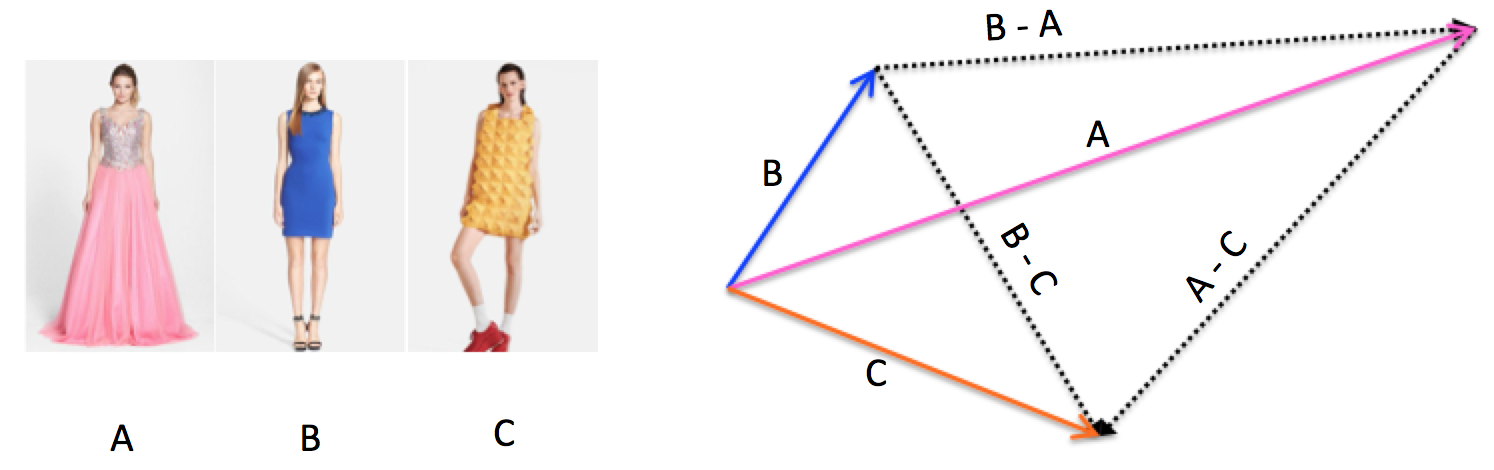
We load the model that will be used to extract visual features extracted via a trained deep neural network. This is an Alex Net architecture that has been trained on the ImageNet data set for 45 iterations.



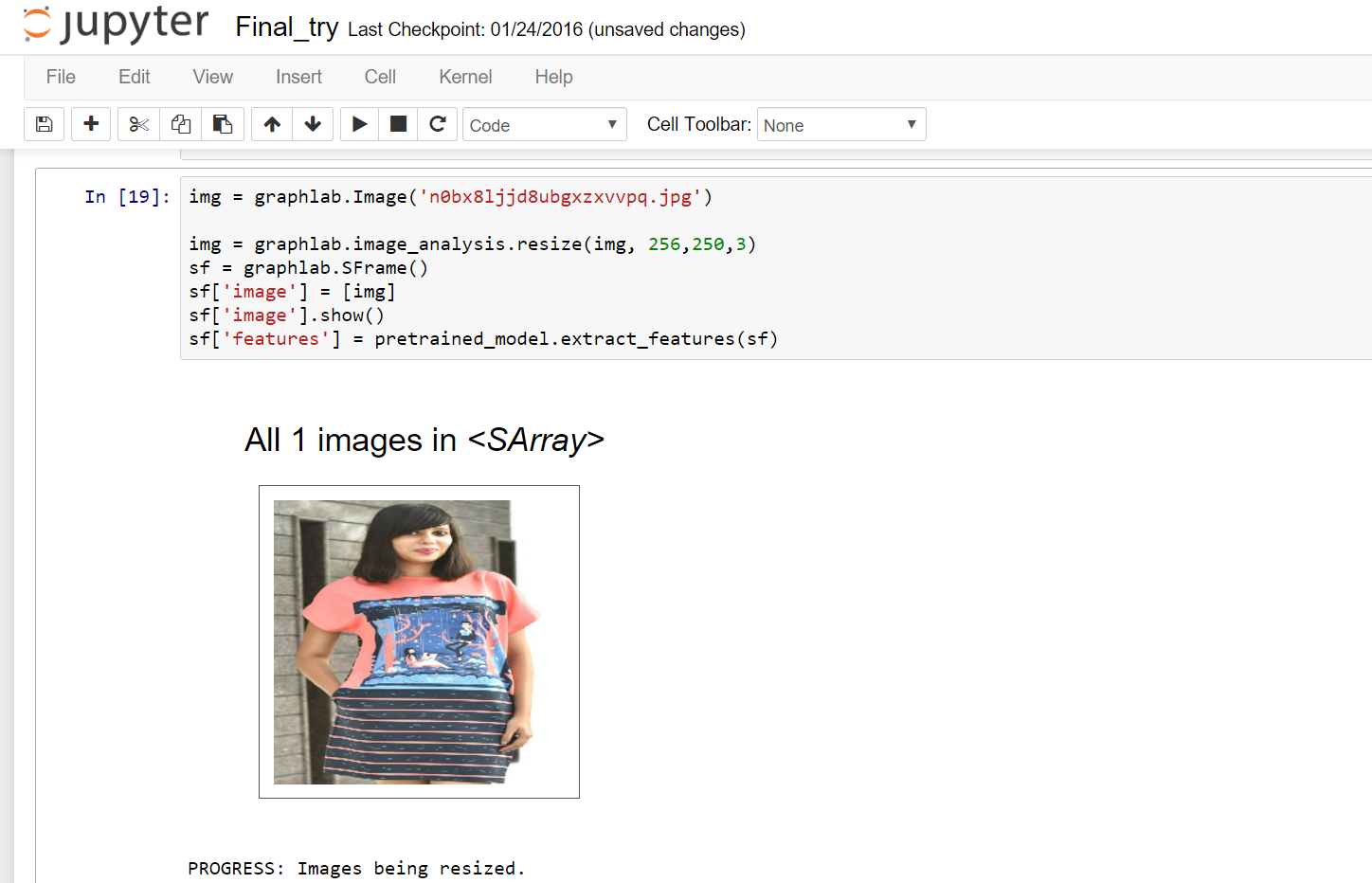
http://localhost:8888/notebooks/Desktop/pydata_seattle_2015/02-image_similarity_search/images/workflow2.png

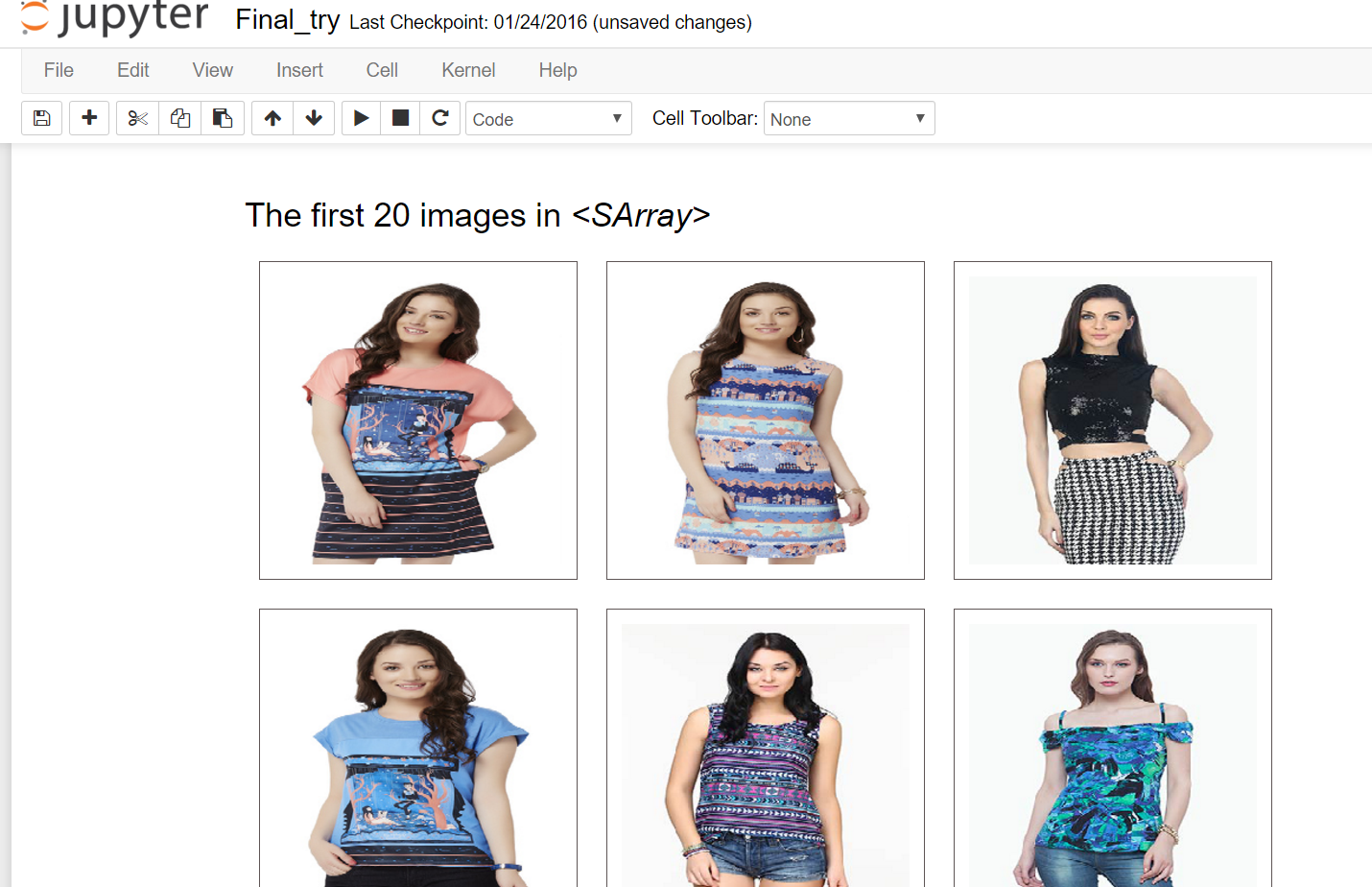


This is the last step in building the similar items recommdendation model. Using the features we extracted above, we are going to create a *k*-Nearest Neighbors model that measures the distance between all our features enables end users to find products whose images match most closely.



# Test:

1. Selecting the test image and resizing to a standard size in dataset.
2. Querying the nearest model and printing the resulted Images.



# Limitation:

* Lack of computation power to train my own deep learning neural network. Hence resulting in use of ALexNet.

# Bibliography:

* ImageNet Classification with Deep Convolutional Neural Networks by Alex Krizhevsky , Ilya Sutskever , Geoffrey E. Hinton -2012
* Python references.