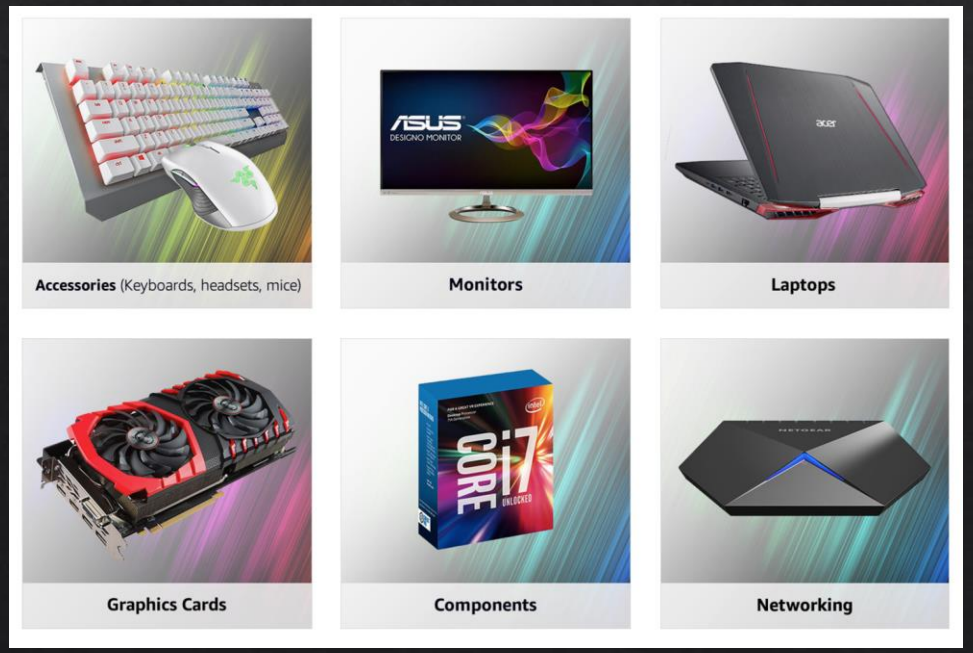
Project: Retail Products Classification

Team: Prashant
Xiaolong Li

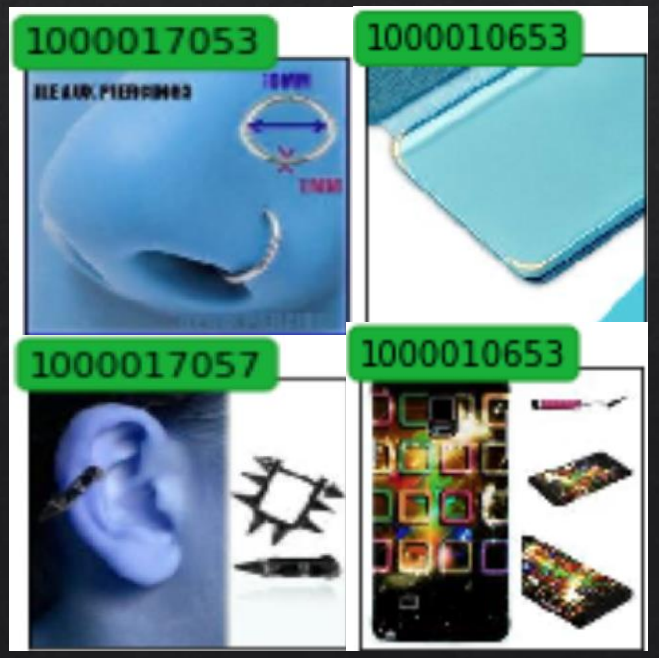
Challenges:

1. 30M products online on Cdiscount.com
2. find incorrectly classified products

Easy Classification



Difficult Classification



Project Aim:

Compare results from:

1. Traditional Computer Vision Techniques
2. Transfer Learning (Pre-trained model)
3. CNN
4. ResNet-50

Database used:

from kaggle.com

- 9M products
- 15M images 180x180 px
- 5000 product categories
- 73Gb of data

Techniques To Be Used

Traditional Computer Vision Techniques	Transfer Learning	CNN	Residual Network
SIFT + SVM	Use Pre-trained Google InceptionV3 Model	Layers: Convolution Convolution MaxPooling Dropout Convolution Convolution MaxPooling Dropout Flatten Dense Dropout Dense Dropout Dense	ResNet-50 Model

Computing Resources

- Local Machine (MacBook Pro)
  - 2.7GHz i5 CPU, 8Gb RAM
- Kaggle Website Platform
  - Python 3, Keras, TensorFlow & Jupyter Notebook
  - 32 CPUs, ~20 Gb RAM
- Google Cloud
  - We installed Python 3, Keras, TensorFlow & Jupyter Notebook
  - 2 K80 GPUs, 104Gb RAM, 16 CPUs
- Google DataLab
  - Preinstalled Keras, TensorFlow & Jupyter Notebook
  - Default CPU Machine
- Google Storage
  - to use Google Cloud and DataLab

Current Work

- CNN on Kaggle, Google Cloud, Google DataLab
- Model built & trained for 15 categories equivalent to 10,000 training images and 2,000 test images

Issues Faced

- Setting up online platforms takes too much time! Too many steps need to be followed
- Reading database batch by batch on Google Cloud
- Jupyter Notebook not working on Google Cloud Instance - No graphical interface for debugging

Results

- Success Rate: 42%
- Training time: 2.38 hr

```
Epoch 1/10
10000/10000 [=======] - 892s - loss: 2.5203 - acc: 0.3595
Epoch 2/10
10000/10000 [=======] - 865s - loss: 2.3215 - acc: 0.4055
Epoch 3/10
10000/10000 [=======] - 835s - loss: 2.1702 - acc: 0.4150
Epoch 4/10
10000/10000 [=======] - 857s - loss: 2.0077 - acc: 0.4156
Epoch 6/10
10000/10000 [=======] - 857s - loss: 1.9668 - acc: 0.4156
Epoch 7/10
10000/10000 [=======] - 867s - loss: 1.9435 - acc: 0.4156
Epoch 8/10
10000/10000 [=======] - 811s - loss: 1.9272 - acc: 0.4156
Epoch 9/10
10000/10000 [=======] - 850s - loss: 1.9195 - acc: 0.4156
Epoch 10/10
10000/10000 [=======] - 863s - loss: 1.9052 - acc: 0.4156
```

Future Work

- Finish product classification using the mentioned 4 techniques
- Use TensorBoard for debugging
- Improve results to a good extent
- Compare results from the 4 techniques

Thank You