MAY 30, 2020

SNEAKER ML

A STUDY ON HYPED SNEAKERS WITH MACHINE LEARNING

ANDREW ZHAO

Domain Background

This project is aimed to discover the differences in hyped sneakers and regular shoes as well as explore differences in *hyped* sneaker designs. *Hyped* sneakers are defined as sneakers that are listed on popular resale sites such as **Goat**, **Flight Club** and **StockX**. The resale market of these coveted sneakers has flourished in the last couple of years. While some people still view *hyped* sneakers has a past time hobby, others actually became professional resellers. Additionally, sneaker companies also interested to find out what “works” and what are some design practices that can make their products excel in resell markets.
Since *hyped* sneakers have great business value for sneaker resellers and shoe brands alike, it is then vital to try to identify design characteristics in these *hyped* sneakers that set them apart. The purpose of the project is to explore *hyped* sneaker designs, what differs them from “everyday” shoes, as well as what are some character defining designs that different brands are doing to distinguish from their counter parts.

Problem Statement

The most direct way to compare sneakers is to examine their design. It is therefore logical to examine shoe designs with their images. It is important to note that this view is going to ignore the background story of a shoe (eg. whether they are backed by popular celebrity). This project is going to scrutinize the images of shoes that are categorized as both *hyped* and *non-hyped*. Some approaches will involve clustering [1], classification as well as image generation [2]. Clustering will attempt to find natural groupings in input data and make sure the dataset is explored thoroughly. For classification, two different tasks will be performed:

1. Binary classification to distinguish *hyped* and *non-hyped* shoes.
2. Multiclass classification to see how well machine learning algorithms can identify design differences in brands.

For both of these tasks, evaluation metrics like test accuracy and F1 scores can be used to ensure validity of results if there exist class imbalance issues. Finally, GANs can be used to "generate" novel *hyped* sneaker designs to find some more additional characteristics of *hyped* sneakers.

Data Collection

The dataset consists of two major parts: *hyped* sneakers and *non-hyped* shoes. For hyped shoes, all images are obtained from **Goat**, **Flight Club** and **StockX**. A *Python* script was employed to gather all of the image links and download them automatically. The name of the shoe was also saved along with the file. Conveniently, all the shoes’ names that were listed in the above site had the brand of the shoe in the name so further scraping was not necessary. The

[1]: Chang, Jianlong & Wang, Lingfeng & Meng, Gaofeng & Xiang, Shiming & Pan, Chunhong. (2017). Deep Adaptive Image Clustering. 5880-5888. 10.1109/ICCV.2017.626.

[2]: Goodfellow, Ian & Pouget-Abadie, Jean & Mirza, Mehdi & Xu, Bing & Warde-Farley, David & Ozair, Sherjil & Courville, Aaron & Bengio, Y.. (2014). Generative Adversarial Nets. ArXiv.

initial count of the number of images from **Goat** was 972, 1075 for **Flight Club** and 866 from **StockX**.

For the non-hyped sneaker images, they were all gathered from **Amazon** by querying *"Men's Shoes"*. The assumption here is that shoes from **Amazon** are not considered *hype* most of the time. Further hand selection was needed because there indeed were some *hyped* sneakers found in the scraped **Amazon** data. At this stage, the total number of images for the *non-hyped* dataset was 4293. Notice here that the *non-hyped* outnumber the *hyped* category because some later removal is needed for the *non-hyped* dataset.

Preprocessing

After scraping the data, preprocessing was needed for these images. First, some images like the ones found in Appendix A were deleted. Most of the removed photos were from the **Amazon** dataset and these images were not images of shoes or are very awkwardly oriented. As for the *hyped* dataset, most of the images of shoes were satisfactory.

A huge hurdle at this stage is that most of the images of shoes belong to the *hyped* dataset were shoes pointing to the right, but the *non-hyped* data set had shoes angled at many different kinds of orientations. Even though most classification training tasks usually perform transformation on the training data when fed into models, by having a different initial transformation will have a different probability of the kinds of transformations that the *non-hyped* and the hyped shoes will have when fed into classification models therefore, the in theory these models might learn that instead of actual design differences which is the intended task.

A possible automated solution could be using RotNet [1] which detects a rotated angle of an image using CNNs. However, it is very computationally expensive to employ a CNN model for this small task. Additionally, techniques such as getting the contours of the shoe then using a cv2 function *fitElipse* to calculate the angle was not successful.

The eventual technique was to first use Canny Edge detector to detect the edges of the grayscaled photo. Then the edges are used by Hough Transformations to detect lines of the image. The detected lines are very short lines that are along the edges of the shoe. Since shoes have a natural angle in the front, this angle will either incur a positive or negative angle. Positive angle will therefore likely to be a right facing shoe while negative angles will probably be a left facing shoe. Therefore, using the gathered Hough Transformation detected lines, the orientation and rotation of the shoes is also estimated. Below are figures demonstrating this series of transformations employed by the described algorithm

[1] Gidaris, Spyros & Singh, Praveer & Komodakis, Nikos. (2018). Unsupervised Representation Learning by Predicting Image Rotations.

Figure 1 Canny Edge detected shoe image

Figure 2 Hough Transformation detected lines

Figure 3 Final after mirror & erase hough lines

Next, after standardizing the orientation and rotation of *non-hyped* shoes, a manual selection process was used to clean out additional outlier images that were either

- unsuccessfully transformed
- not image of a shoe
- image of a hyped shoe

After removal of the above images, the *non-hyped* dataset was left with 3845 images. Then, all of the remaining images were rescaled to 224 x 224 because the subsequent classification training and other image explorations need all the images to be of the same size.

Exploratory Data Analysis

In order to proceed to more advanced topics, it is vital to first get familiar with the data that we are working with. After preprocessing and removal, the dataset is left with 2913 *hyped* sneaker images and 3845 images of *non-hyped* sneakers. We can see a montage of *hyped* and *non-hyped* sneakers in the figure below.

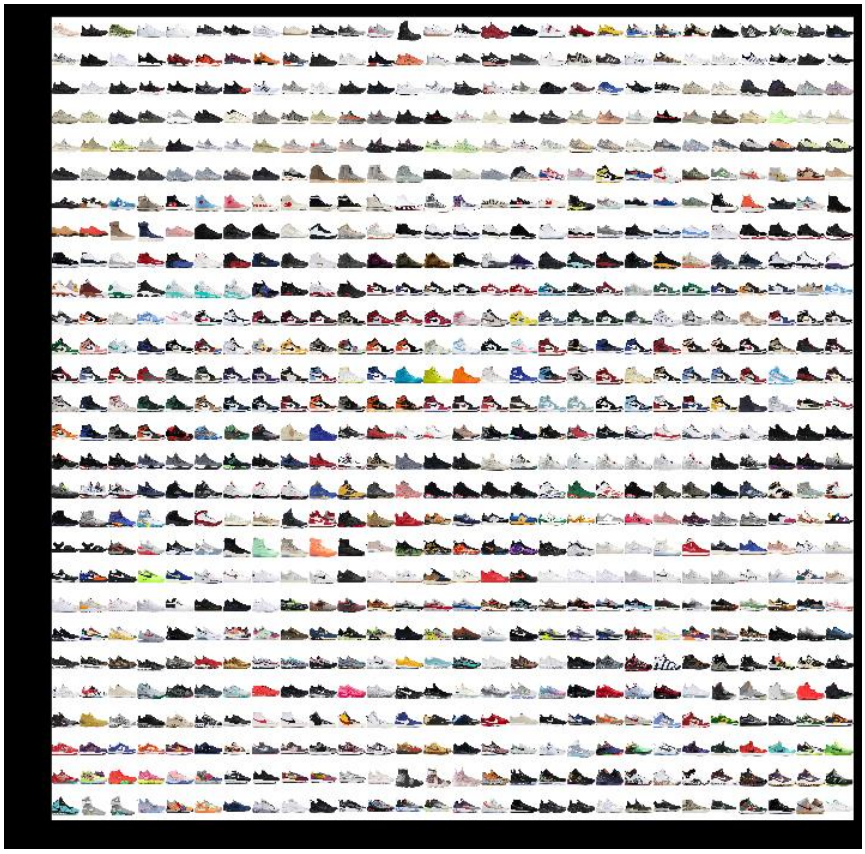


Figure 4 Hyped Sneakers

Figure 5 Non-hyped sneakers

In addition to the image patterns itself, pairplots of raw image height and width are also useful to see how our data varied during the collection stage.

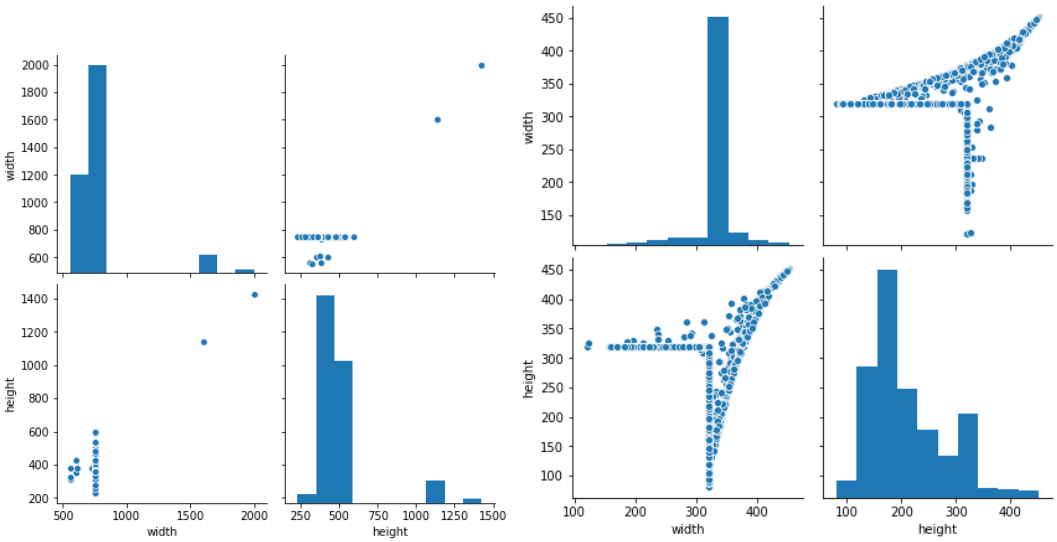
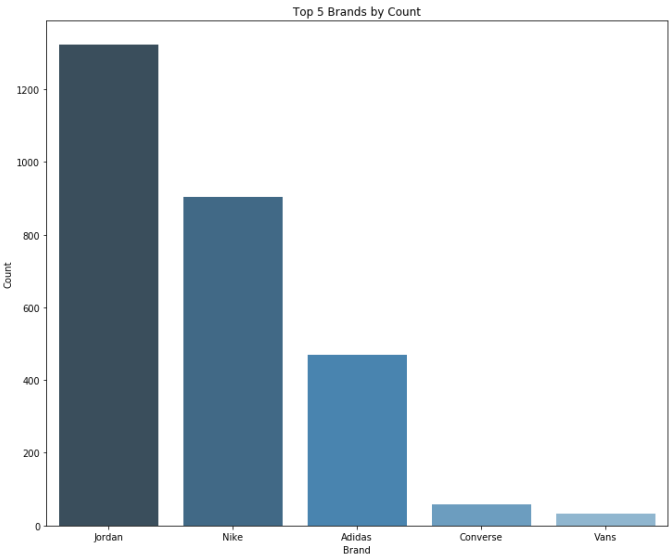
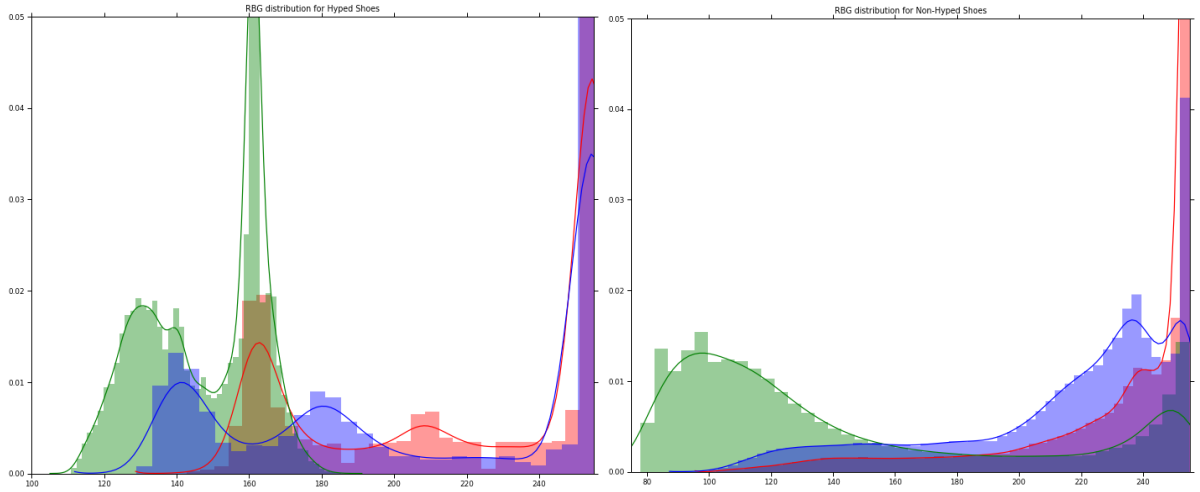


Figure 6 (left) pairplot of raw dimensions of images of hyped sneakers. (right) pairplot of raw dimensions of images of non-hyped sneakers.

Another useful information is the class distribution of brands for *hyped* sneakers.



Color RGB images have three channels of Red Green and Blue. The range of values are from 0 to 255 for each channel. For our dataset, we can plot the average pixel intensity for each channel with distribution plots for each of the distinct part of the dataset.



We can see the distributions for RGB channels for *hyped* and *non-hyped* are quite different. This difference in pixel intensities along the three channels can be calculated by using mutual information. Additionally, to test if mutual information is a plausible tool for determining the differences between *hyped* and *non-hyped* shoes, we got the pixel distribution of a recently released *hyped* shoe: **the Nike Ben & Jerry dunks** and compared the pixel intensities to each of the average pixel intensity for *hyped* and *non-hyped* shoes from our database. The calculated mutual information between *hyped* and *non-hyped* shoes was 8.09 while the mutual information between *hyped* shoes and the Ben and Jerry was 1.80 and the mutual information between *non-hyped* and the Ben and Jerry was 1.75. This proves that there could be indeed some similarities in RGB intensities for *hyped* shoes.

Additionally, in order to further explore the designs of *hyped* shoes, a “mean” shoe can be calculated by averaging every image tensor. The result is depicted below.

We can see this “mean” shoe very much resemble the shoe model **Jordan 1s**, which make up a decent portion of our *hyped* shoes part of the dataset. Here are some more “mean” shoes with different variations of database subsets:

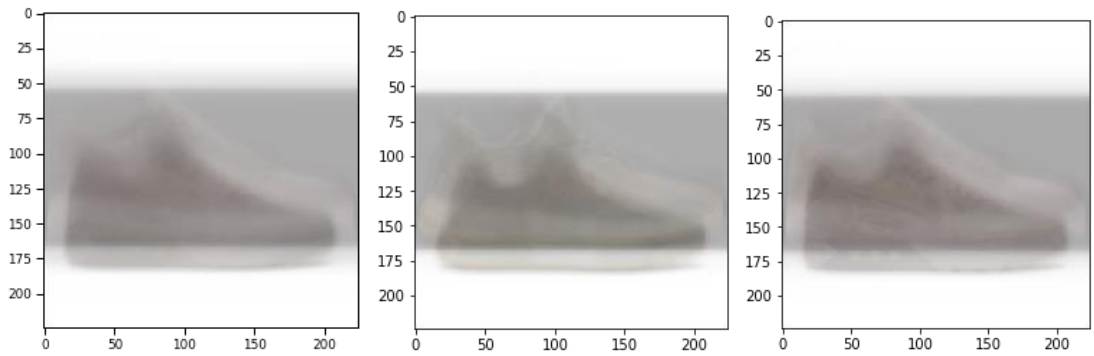


Figure 7 mean shoes made from subsets of “no Jordan 1s”, “adidas” and “nike” respectively

Furthermore, we can also get the *eigenimage* of the *hyped* shoes to take a look at summary representation of the dataset in a visual way.

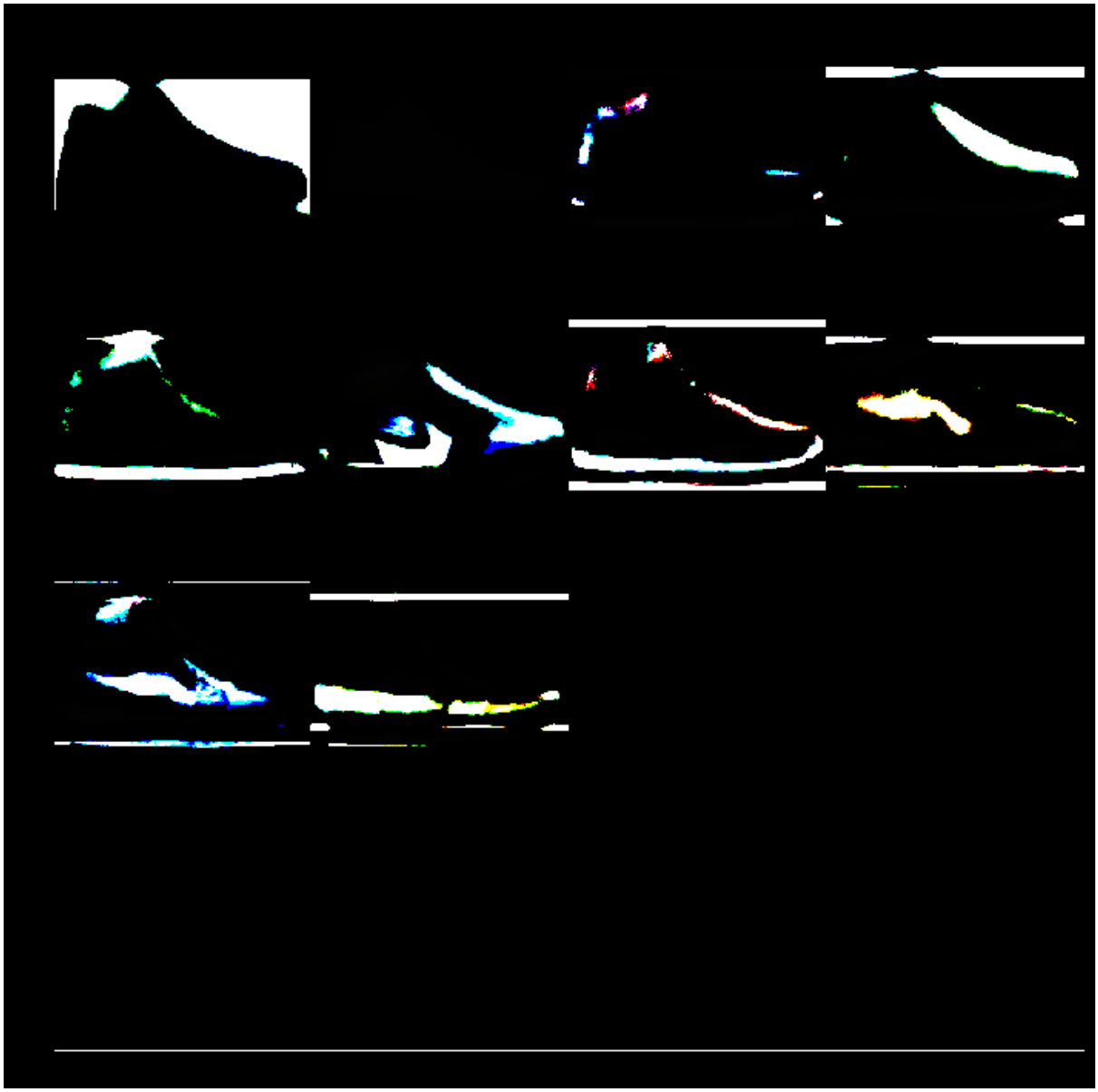
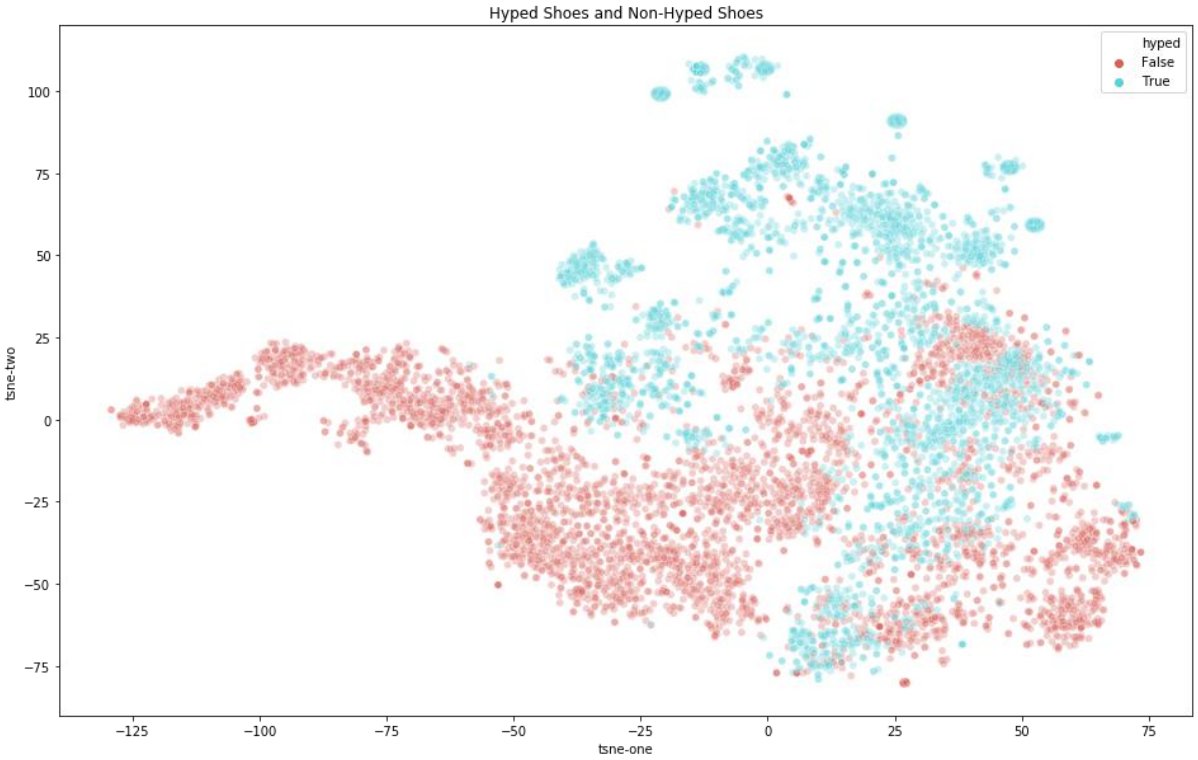
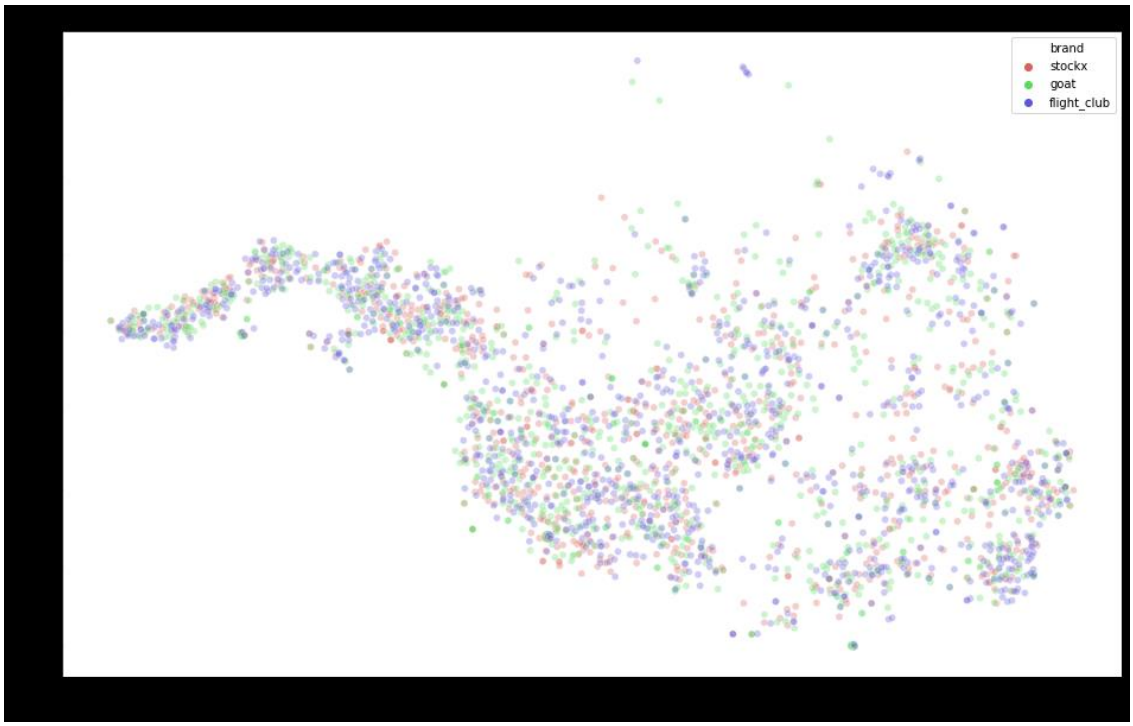


Image Clustering

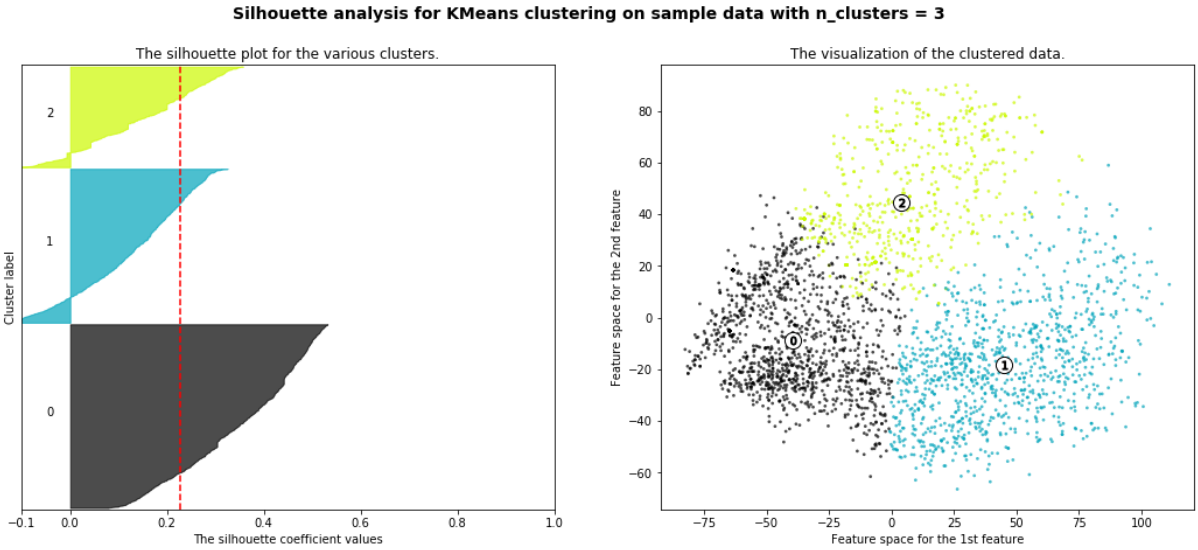
To further explore the dataset, techniques such as TSNE and KMeans were used. Below we can find the TSNE plot for *hyped vs. non-hyped* shoe images of their RGB tensors. The dataset seem to not be that well separated in these two dimensions but does indeed form some smaller individual clusters.



The RBG vectors of *hyped* shoe separated by brand was also plotted but no clear pattern for brands were visible from this plot.



Additionally, by first reducing the RGB image tensors dimensions using PCA, KMeans could be employed to find better clusters in our dataset. By conducting silhouette analysis and using the elbow method, 3 clusters was the best K to choose for the *hyped* shoes part of the dataset.



Besides using PCA, trained CNN models are also excellent feature extractors. We first download pretrained models and take out the dense layers. The output of the CNN layers can be then flattened and fed into TSNE to represent our data in two dimensions.

Image Classification

After exploring the dataset using various EDA and clustering techniques, we are finally set to build classification models.

Classification can be vital tools that can be used by sneaker reseller our sneaker designers to use the output of models to see how *hyped* a shoe design is. Additionally, shoe brands can seek to maintain their “identity” when designing for future shoe models by using a brand classifier.

Since our data is taken from resell sites and that duplicate shoe models probably exist, we do not want the same shoe to be split into both train and test sets. Instead, we want a particular shoe model, even if there are many of them, to belong either in the train or test set exclusively. This is achieved by using fuzzy name matching and grouping the similar names into the same groups. Then we can proceed to split the train and test set randomly with these groups.

The chosen test size is 0.1, which is common. Finally, the train and test images are copied into two different main directories:

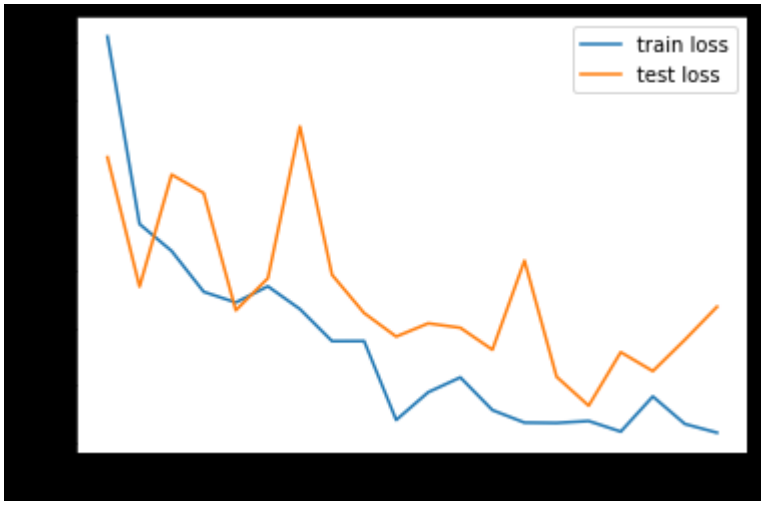
- hyped classification
- brand classification

for the purpose of two distinct class. Furthermore, for each of these task directories, train and test folders are made to make subsequent modelling easier with Pytorch. Before the images are fed into the model, standard random transformations are done to the train set to ensure the robustness of our models.

For the first task, the *hyped* or *non-hyped* classification, three main CNN pretrained model architectures were showing promise:

1. VGG16
2. MobileNet V2
3. ShuffleNet

The standard Cross Entropy Loss was used for classification and Adam optimizer was used throughout with learning rate ranging from 0.001 to 0.01. Additionally, batch size of 64 seem to perform the best. Below is a sample loss per iteration for ShuffleNet architecture:



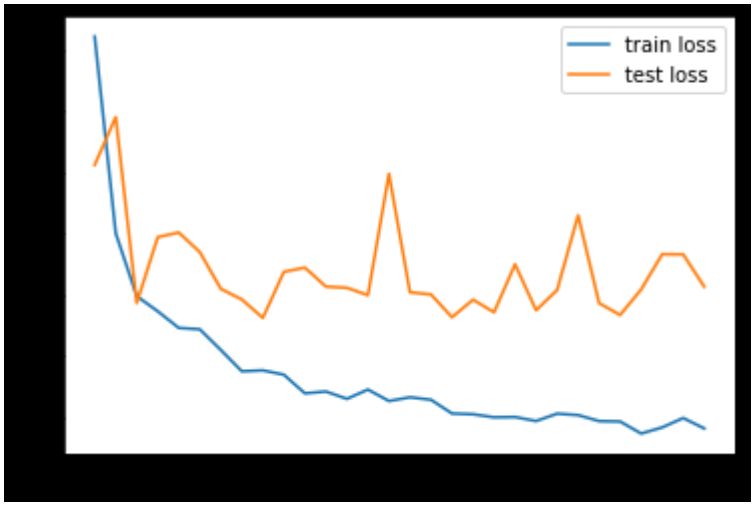
The final chosen model was ShuffleNet with a test accuracy of 0.96, with batch size 64, learning rate of 0.001 and 20 epochs.

Next, we used Ray Tune to find a better learning rate for the model. Unfortunately, Tune was not able to further improve the model drastically, therefore the above model can be finalized.

For brand classification, since the class imbalance for brands could be a major problem, the training data is separated into only 4 classes:

1. Jordans
2. Nike
3. Adidas
4. Other

Even with this split, the “other” class only had 197 samples in the training set while Jordans had 1190. For pretrained models, VGG16, MobileNet V2 and ShuffleNet were used again because they performed decently. Below is the loss per iteration for VGG16 model which is the finalized model architecture before hyperparameter tuning.



And a classification report for the VGG16 model:

Classification Report for VGG

	precision	recall	f1-score	support
0	0.88	0.81	0.84	47
1	0.98	0.97	0.98	132
2	0.84	0.90	0.87	90
3	0.71	0.71	0.71	21
accuracy			0.90	290
macro avg	0.86	0.85	0.85	290
weighted avg	0.90	0.90	0.90	290

Because of the class imbalance, the use of f1-score was a better metric when determining the best model. We tested out again on the newly release Ben and Jerry Dunks which the final model correctly classified as **Nike**.

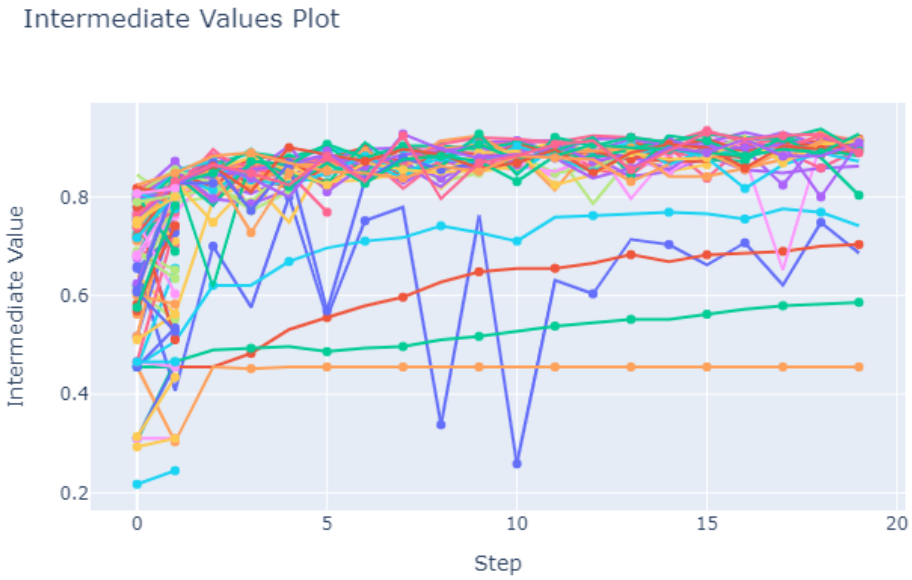
Next, hyperparameters were tuned for the initial VGG model. The search space and their respective best values are shown below:

```
Params:
    n_layers: 2
    n_units_l0: 7933
    dropout_l0: 0.21148857210893734
    n_units_l1: 12020
    dropout_l1: 0.46695470143984114
    optimizer: RMSprop
    lr: 1.0455375032668124e-05
```

And the final classification report shows a decent improvement from our previous initial model:

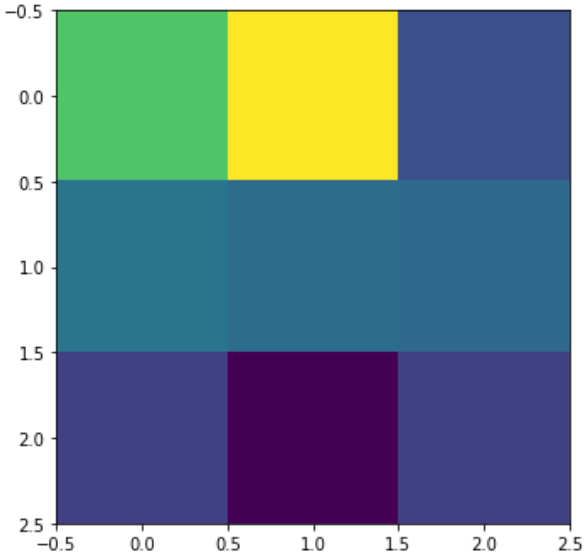
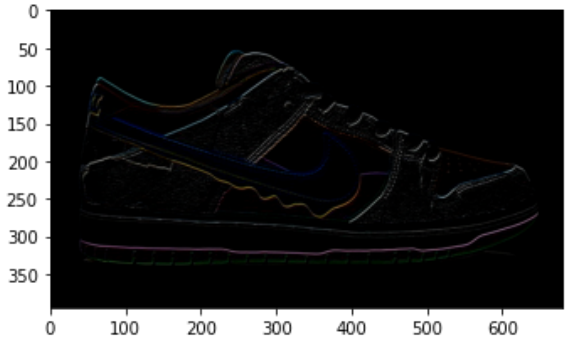
```
Classification Report for VGG_optimized
Train acc: 0.9657, Test acc: 0.9138
              precision    recall  f1-score   support

           0       0.89      0.89      0.89        47
           1       0.93      0.98      0.96       132
           2       0.92      0.84      0.88        90
           3       0.85      0.81      0.83        21

    accuracy                           0.91       290
   macro avg       0.90      0.88      0.89       290
weighted avg       0.91      0.91      0.91       290
```

And here is an intermediate test accuracy plot for all the trails Optuna conducted during the tuning stage.

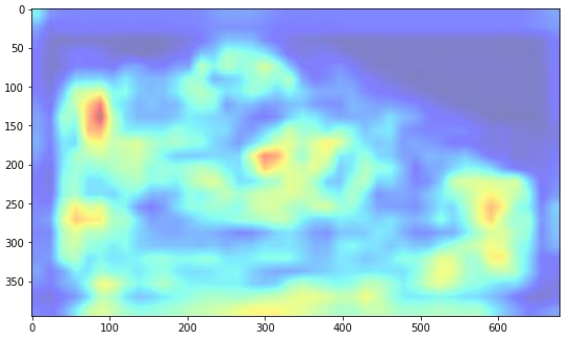
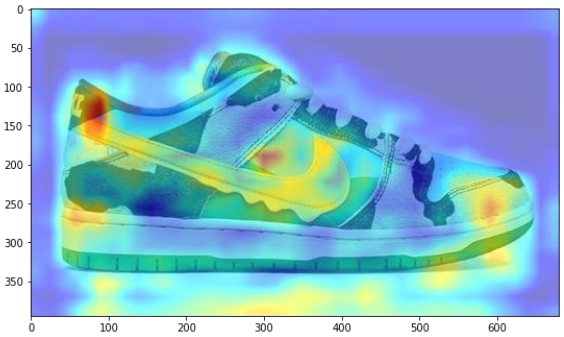
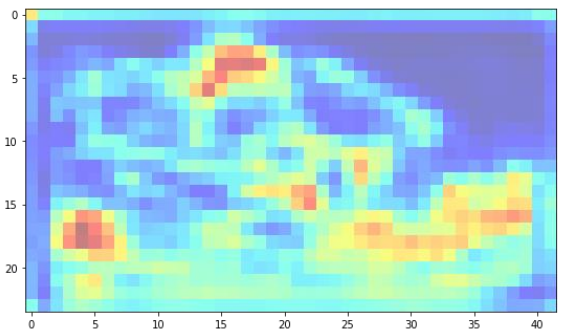


CNN Layers Visualization

In order to better understand our models, we can use visualization techniques to see why our models are behaving in a certain way. By using taking our individual filters that are trained in our model, we can apply them to images and inspect closely on what they are doing. An example can be seen below. This 3 x 3 kernel seem to detect some important edges of the input shoe image.

Additionally, Score-CAM and Grad-CAM[1] are other ways to see where our model's gradients are most 'exited'.

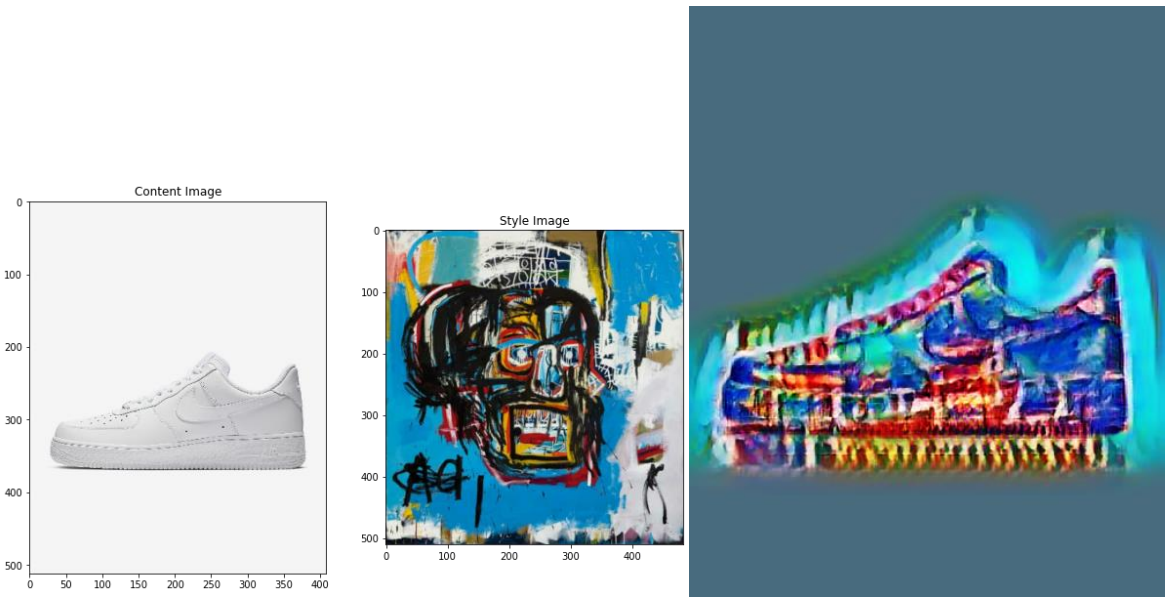

Some edges are detected more than others when our image is going through conv layers and it is a great way to see this with Score-CAM and Grad-CAM visualizations.

[1] Jing, Yongcheng & Yang, Yezhou & Feng, Zunlei & Ye, Jingwen & Yu, Yizhou & Song, Mingli. (2019). Neural Style Transfer: A Review. IEEE Transactions on Visualization and Computer Graphics. 10.1109/TVCG.2019.2921336.

Generative Models

Beside classification models which could potentially bring values to the people in the sneaker resell industry, generative models could also have an impact in the industry. Generative models could inspire shoe makers to have seek for inspiration as well as prototype quickly.

The first generative technique was used was neural style transfer [1]. This technique involves a content image (in our case a shoe) and a style image. Using CNNs, the style can then be transferred to the content image which creates a combined image. A demonstration is show below which has a content of the classic *Air Force 1s* and a style image from *Basquiat*.



Additionally, unsupervised techniques like deep dream [2] can also be very inspirational that generate patterns onto existing images from pretrained Inception v3 model.

[1] Jing, Yongcheng & Yang, Yezhou & Feng, Zunlei & Ye, Jingwen & Yu, Yizhou & Song, Mingli. (2019). Neural Style Transfer: A Review. IEEE Transactions on Visualization and Computer Graphics. 10.1109/TVCG.2019.2921336.

Besides these stylistic techniques to “enhance” an image, generative adversarial networks [1] can actually come up with novel designs. In particular, Wasserstein GAN-GP are used because of its use of the *Earth Mover’s Distance* as its value function [2]. This results in better behavior in its gradients compared to normal GANs. Additionally, WGAN-GP also introduced gradient penalty which penalizes the model if the gradient norm is shifted to not 1 instead of clipping it like regular WGAN [2]. Below is a sample output from the model with 4.6k iterations. The generated shoe images are still not on par with some of the GAN outputs like BiGAN or ProGANs, but this is a decent result with small dataset and limited compute power.

Conclusion

In conclusion, this project explored the various styles of *hyped* sneakers and their comparisons to *non-hyped* shoes. These classification models can be valuable to both reseller and shoe makers. Furthermore, a brand comparison model can be valuable to brands to further strengthen their brand image. Additionally, generative models can be inspirations for shoe makers to make novel designs or rapid prototyping of designs.

[1] Goodfellow, Ian & Pouget-Abadie, Jean & Mirza, Mehdi & Xu, Bing & Warde-Farley, David & Ozair, Sherjil & Courville, Aaron & Bengio, Y.. (2014). Generative Adversarial Nets. ArXiv.
[2] Gulrajani, Ishaan & Ahmed, Faruk & Arjovsky, Martin & Dumoulin, Vincent & Courville, Aaron. (2017). Improved Training of Wasserstein GANs.

Deep learning techniques are proven to be great on image classification tasks using CNNs as well as for generative models. It is then up to people in various industries to use them to their advantage to further set them apart from their competitions.

APPENDIX A Random Photos

At WGAN-GP at 5.6k