-- Summer Internship Report --

July-August 2020

Generation of Artworks from face wash samples obtained from Nykaa.

Prepared By :

Komal Agnihotri,IIT Bombay .

Gokul Ram ,IIT Bombay .

Aniket Sadashiva ,IIT Bombay .

Guide :

Sreenivas Narasimha Murali

Description of the work :
To generate artworks from face wash samples obtained from Nykaa website using Detecto (for Object Detection) and Google Cloud Vision API(for OCR).

Collection of face wash samples :
Facewash samples were obtained from Nykaa website using a python script (Nykaa Image Downloader.py) that uses Selenium framework to scrape through the site to collect images.

Cleaning of obtained samples :
Sample with facewash of tube type were selected and rest were rejected for the generation of artwork.Some other products like oils,scrubs,powders etc were also removed.
We finally obtained **303** samples.

OCR using CloudVision API :
We modified the script **'sample_ocr.py'** to 'OCR for Multiple Files CloudVision API.ipynb' and used it for generating OCR text from images into a python dictionary (which was subsequently made into an excel file) for further processing.

- Process followed for correction of text after Text detection using Google Cloud Vision API.

1. The Text blocks were transferred into a table with each column corresponding to a text block
2. Comparison of detected text to the text in artwork, checking for the following:

Sr no	Type of Text	Corrective action
i	Missing texts	Corrected by entering -1 in empty blocks, marked in red
ii	Wrongly detected spellings	Correction done by comparing to actual spellings
iii	Two different information merged in a block	Separation of text by their original groupings in the artwork
iv	Vertically oriented text missed	Corrected by entering -1 in empty blocks, marked in red
v	Poorly detected text from circular logo	Corrected by entering -1 in empty blocks, marked in red

- Process followed for categorisation of corrected text to retrieve useful data for artwork generation:

1. Analysing the detected text boxes to find the common categories in all face-wash artworks
2. Finalising four categories of text:
    a. **Brand name**
    b. **Key Ingredients**
    c. **Special claims** – based on the effect it has on the face and skin
    d. **Others**- type of face-wash, made in country , majorly highlighted, miscellaneous information
3. Any missing category of information was substituted by entering -1 in empty blocks

Also text-coordinates and area were extracted from cloud vision to another excel sheet.

Creating the database of samples:
Next stage was labeling (Rectangular Boxes) the samples obtained into two categories using LabelIMg tool and generating .xml files for the same so that they could be compatible with Detecto :

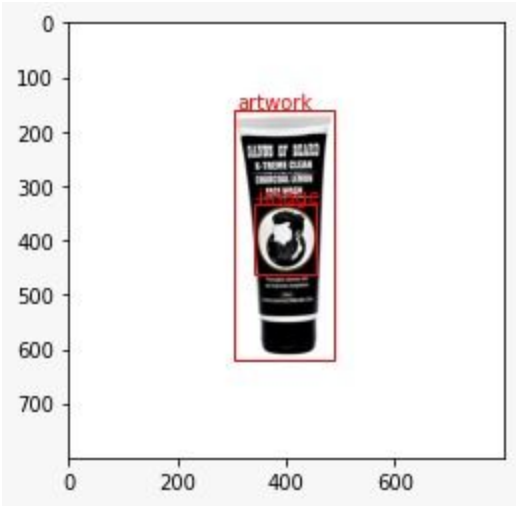
- Artwork -The outline of the entire facewash sample.
- Image - The image contained inside the artwork like fruits,special designs,human faces etc.

Training the Detecto framework :
We trained the entire dataset on Detecto and obtained the following results :

- Learning rate=0.001
- EPOCH: 80
- LOSS: 2.56 %
- Average Time taken per epoch: 129 seconds

The weights were saved in **'Nykaa-Image-Analysis-Weights.pth'.**

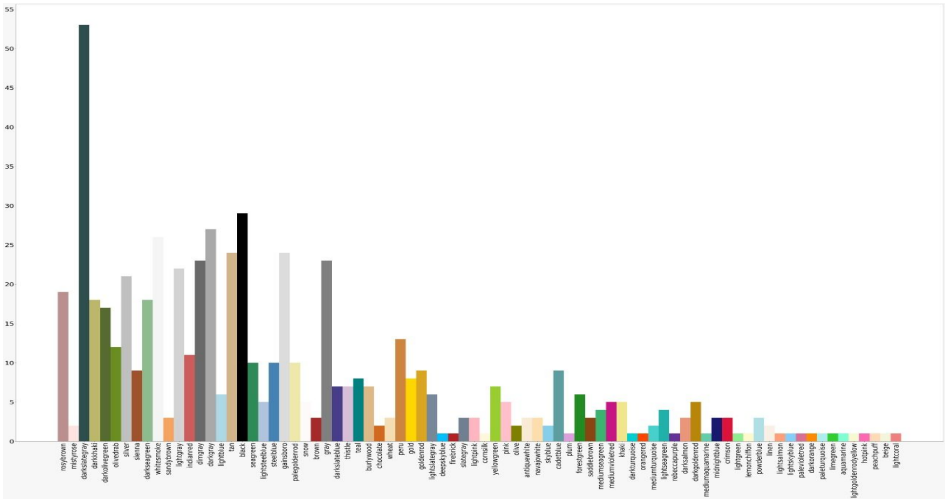
We stopped after 80 epochs as there was not a significant drop in loss.



Sample Result obtained after training the model

Generation of Color Histograms :

A color histogram was generated using the frequency of top 3 colors using k-means clustering algorithm and their hex values were recorded in an excel sheet after it.The script is 'colorHistogram.py'.The colors were compared with CSS4 standard colors.



Color Histogram of Facewash samples.

Generation of Artworks:

An excel sheet was made containing data about artworks ,images, texts and their coordinates with respect to the artwork .The sheet also contained relative area occupied by images .Link to the sheet :
https://docs.google.com/spreadsheets/u/2/d/1lDOOQPm8TNSUmT9ommZRGX8XEQ4FPnXGl1XtCPUsJ88/edit?usp=sharing

It has 4 sheets.

- -1 entry represents image/artwork/text not found or corrupt input.
- First sheet (Image Analysis) : Contains coordinates of artwork and co-ordinates of images relative to artwork .We have subtracted the coordinates of top left coordinate of artwork to get the relative co-ordinates.It also contains the top 3 colors for each image.
- Second sheet (Relative Coordinates of Text) : Contains Relative coordinates of Text with respect to artwork.
- Third sheet(Area Analysis) : Contains absolute and relative areas occupied by texts
- Fourth sheet (OCR text) : Contains corrected-text generated by Cloud Vision API.(Red color represents corrected text)

The script is ‘Artwork Generation.py’.

Then we used OpenCV and PIL to generate artworks by putting the most dominant color followed by creation of rectangle and addition of texts horizontally.These have been placed in the folder ‘Generated Artworks’.

The following are some of the samples :

Images not to scale !

Classification of texts : We also tried to classify texts by using one-hot representation of texts and decision trees and random forests .However it resulted in overfitting and hence we rejected it .

Additionally there were **three talks** given by each one of us :

The First talk was given by Aniket ,based on Computer Vision techniques primarily focussed on Deep Learning methods .It included a brief overview about classic CV techniques like HOG,SIFT,Segmentation etc.Then it moved on to RCNNs and then to YOLO.Dry runs and brief overview of how each of these methods work was also presented.

The second talk was given by Gokul .The primary focus of the second talk was on **Image processing techniques and computer vision** using classical CV techniques.

Morphological transformations are simple nonlinear operations carried out on images pixel data. They are done to remove imperfections by accounting for their form and structure. The elementary morphological transformations on images are erosion and dilation respectively. The various other operations such as opening and closing are merely a combination of them. When performed on a binary image or a color image, they effectively lead to noise removal. The other important transformations include the morphological gradient, top hat transform and black hat transform. Each with a specific application of enhancing a bright or dark object of interest in the image. Furthermore, the Hit & Miss transform and skeletonisation operations can find ridge lines and triple points.

Watershed Segmentation is a transformation carried out on grayscale images. The segmentation leads to partitioning of a digital image. This simplifies its representation into something more meaningful and easier to analyse. The concept involves flooding of the image from its minima (catchment basins), until the peaks (watershed lines) get submerged. To avoid over-flooding barriers are built at specific locations, resulting in segmentation. The problem of over-segmentation occurs due to the local irregularities in the gradient of the image. This is controlled by flooding the topography from a defined set of markers. These markers divide the image into foreground, background and uncertain types of regions. Marked-controlled watershed segmentation coupled with distance transform can be employed to segment mutually touching objects such as coins placed together.

Feature descriptors are a simplified representation of image, encoding only the important information about the image into a numerical footprint. This involves pixel dimension reduction to large magnitudes. The feature vectors thus obtained are useful in object detection and image classification. HOG descriptor counts the occurrences of gradient orientation in the localised portion of an image.

The distribution of the magnitude and direction of gradient thus obtained, can be viewed as a histogram. In effect, an image of 8 x 16 (128-pixel size), is converted to a 9-bin histogram(array). Block normalisation of this descriptor is then recommended to avoid image sensitivity to lighting variations. The Histogram of oriented gradients can capture details such as the shape of a person in image.

The third talk was given by Komal on RNNs and LSTMs. It primarily focused on understanding the working of RNN, GRU and LSTM.

Recurrent Neural Networks (RNN) is a generalization of feedforward neural networks that has an internal memory. RNN is recurrent in nature as it performs the same function for every input of data while the output of the current input depends on the past computations. After producing output, it is copied and fed back to the recurrent network.

Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) are the modified version of recurrent neural networks, which makes it easier to remember past data in memory. The Vanishing Gradient problem of RNN is resolved here.

Conclusion :

The objective of this project was to generate artwork using deep learning models and computer vision. This was implemented in five phases:

The first phase involved Data mining, done by web scraping of facewash images from Nykaa website using Selenium framework. An artwork dataset with 303 images, were thus obtained after data cleaning for ‘tube-type face-wash’ samples.

Optical character Recognition was performed in the subsequent phase using Google's open source platform, the Google Cloud Vision. The text in the artwork was detected in different blocks. Upon documentation of the text detected, it was found that vertically oriented text, characters inside round logos and blurred images gave irregular or missing outputs. These were corrected manually and marked separately in the excel sheet.

The Third phase involved image detection using PyTorch-based Detecto. It enabled a fully-functioning computer vision and object detection model, with just 5 lines of code. The annotation of the dataset was done into two classes, ‘artwork’ and ‘image’. The training was done for 80 epochs as there was not a significant drop in loss. Testing of the model yielded good results, as most images were detected accurately,for the test set. Though exceptions occurred for samples with multiple images inside the artwork.

A Color histogram was obtained in the fourth phase using the k-means algorithm. The top 3 colors were separated and identified. Analysis of the histogram gave insights on the majorly utilized colors in face wash artworks. The hex values of the colors were recorded in an excel sheet.

The final phase involved the collection and consolidation of all the data from the previous phases. The relative area of an image in any artwork and its positioning with respect to a common reference point was calculated. The text data was studied and characterized into 4 broad categories. The next step was to combine all this data to generate a face wash artwork from scratch using open CV. This was achieved successfully with each artwork having a single color background, 3 lines of description of the facewash and an image positioned inside it.

To conclude, this project was a very enriching experience for the three of us. We, being students and new to the field of CV, AI and Deep learning, explored many facets of how things work in a industry based project. We would sincerely like to thank Mr Sreenivas for his constant guidance, support and encouragement throughout this project. It was a genuinely wonderful learning experience.

Links to various folders and files :

Link-to-the-report : https://docs.google.com/document/d/1R3qb8lqxp8u0DtpUpAMtTv21L6NfSC0ai-c_8nF9beY/edit

Link-to-the-main-drive-Nykaa_Analysis : https://drive.google.com/drive/folders/1nfESUZF1zNT4c8kZVXjWUIjPhjGepuNx

Generated Artworks : https://drive.google.com/drive/folders/15U_bStthtl5aWT4Td9hXt_w_rlO1QkYA?usp=sharing

Link-to-all-the-scripts-folder : https://drive.google.com/drive/folders/1w9XXm1tLwo0ulwShfbnznk0nHHEZx9Mv?usp=sharing

Script-for-Generating-artworks-(Artwork-Generation.py) : https://drive.google.com/file/d/12hnzm6Z3lYmYmuKwYFnK1dvUxH4qBVpR/view?usp=sharing

Script-for-making-Color-Histograms: : https://drive.google.com/file/d/1kE48pNTt5i99ehOfw3bJyLscgu_DKQo3/view?usp=sharing

Script-for-Downloading-Images-from-Nykaa: : https://drive.google.com/file/d/1kE48pNTt5i99ehOfw3bJyLscgu_DKQo3/view?usp=sharing

Notebook-for-generating-excel-sheet-from-OCR-files: : https://drive.google.com/file/d/1sZnxEaQuAGMA5iHDiu5cAUZIuXd_S0Qd/view?usp=sharing

Link to the excel sheet :

https://docs.google.com/spreadsheets/u/2/d/1lDOOQPm8TNSUmT9ommZRGX8XEQ4FPnXGl1XtCPUsJ88/edit?usp=sharing