Abstract

The majority of people in today's generation use technology to manage their life and take care of their basic necessities. The majority of us in this generation shop for clothing, groceries, and electronics via e-commerce websites.

We must conduct a search to determine whether a specific product is offered on e-commerce websites before purchasing it. The issue is that the text-based search demands the precise product names. And everytime we don't know the precise name. Therefore, we require a search method that goes beyond text. Users would find it convenient if e-commerce websites offered image-based search.

We attempted to integrate this image-based search on an e-commerce website in this project.

Acknowledgement

We are thankful to our supervisor, Nazia Jahan Khan Chowdhury to give us proper guidelines to complete our project successfully. We have completed our “CSE 3200 System Development Project” titled ”E-commerce website based on reverse image search technology” with help of our supervisor.

Ahsan Habib , Tashin Ahamed

Contents

1. Introduction

Reverse image search is a technology that helps consumers search for products through images instead of keywords. Reverse image search is a Search Engine Optimization (SEO) technique that uses an image as a query and retrieves information about related products. Visual search is one of the latest advances that is making a mark on e-commerce.

3 big advantages that visual search has over text-based search when it comes to on-page and off-page Search Engine Optimization:

1.Reduction of extra steps for customer: During the buying process, if a customer has to scroll through 10 pages to find a particular product, it can be a buzz kill for them. The chances of them buying a product after going through so many pages are almost NIL. Visual search eliminates the tedious task of searching by bringing users straight to the products that match they are 'visualizing' what. It generally yields a more targeted result and translates to better usability

2.Refined on-site search: Qualified leads that are looking for a specific product usually want more control over their search parameters. Visual search allows them to filter their search beyond keywords. The product recommendations are based on similar visuals, colors and complementary styles.

3.Eliminate roadblocks: The roadblock occurs on-page when users can't find what they are looking for on your page. Assuming that product is out of stock, a carefully implemented visual search program can show results for a similar color, style or size preference. This way you can ensure your customer always finds something that is closest to their search.

Thats why we tried to implement this new innovative tool that will become the future of e-commerce.

2. Background

All the searches on e-commerce websites are text-based search.In general, people don't know the precise name of most of their desired products. As image-based search came into picture. But it took so long time to process images because the processor was so slow back then. However, the Nvidia includes some powerful GPUs. GPU architecture allows parallel processing of image pixels, which, in turn, leads to a reduction of the processing time for a single image (latency) Now gpu can process thousands of images in one second whereas cpu 15-20 images per sec.Now reverse image search can be revolutionary.

3. Features

The main features of this website are:

• Customer can buy their desired product with- out any hardship.
• Save time of customers.
• Increases sales for the business because customers can quickly purchase the things they want.
• This can manipulate customers choice when the products isn't available on stock.It will increase customers' area of interest.

4. Methodology

Reverse image search (or simply called search by image) is a task where given an image as a query, then we need to give some other images that are similar and have correlation with the query image. This task is very useful when we have a very huge image database, and our users want to get some items that look similar with the one that they see. In the other side, this task is also challenging since our machine needs to be able to distinguish and find correlation between one image and the others.

Convolutional Neural Network: the machine's eye

CNN is one of the basic building block when we want to deal with image processing. It works by mimicking the way our brain process image signal. Our brain recognize an object by identifying some semantic patterns —line, edges, curve, texture, colors, etc. — and integrate them into meaningful idea. This is also what CNN wants to achieve.
CNN is basically a series of convolutional layers that is stacked upon each other. The output of a layer will become the input of the next layer, and so on. Each layer will try to capture a spatial feature from the in-

put.
In order to do that, each layer will try to apply some filters to the input and output a “new image” representation. This is done by convolutional filters.

By doing this strategy, we will be able to capture spatial correlation between the neighboring pixels in the input image. Passing the result to the next layers will make the network able to capture even more complex feature.It can be illustrated in Fig 1. The first layers can extract low-level features, like lines and edges. Then, the middle layers will be able to extract curves. The last layers can distinguish more high-level features, like the shape of tire from a car, or the wing of a bird.

The high-level features is then flattened to make 1D-vector. The vector is fed into fully-connected network to make a class prediction. This way, our machine will be able to distinguish the cats and the dogs image!

Extract Image’s Feature Using Deep Neural Network: Deep neural network is a stack of many convolutional layers. This way, the net-

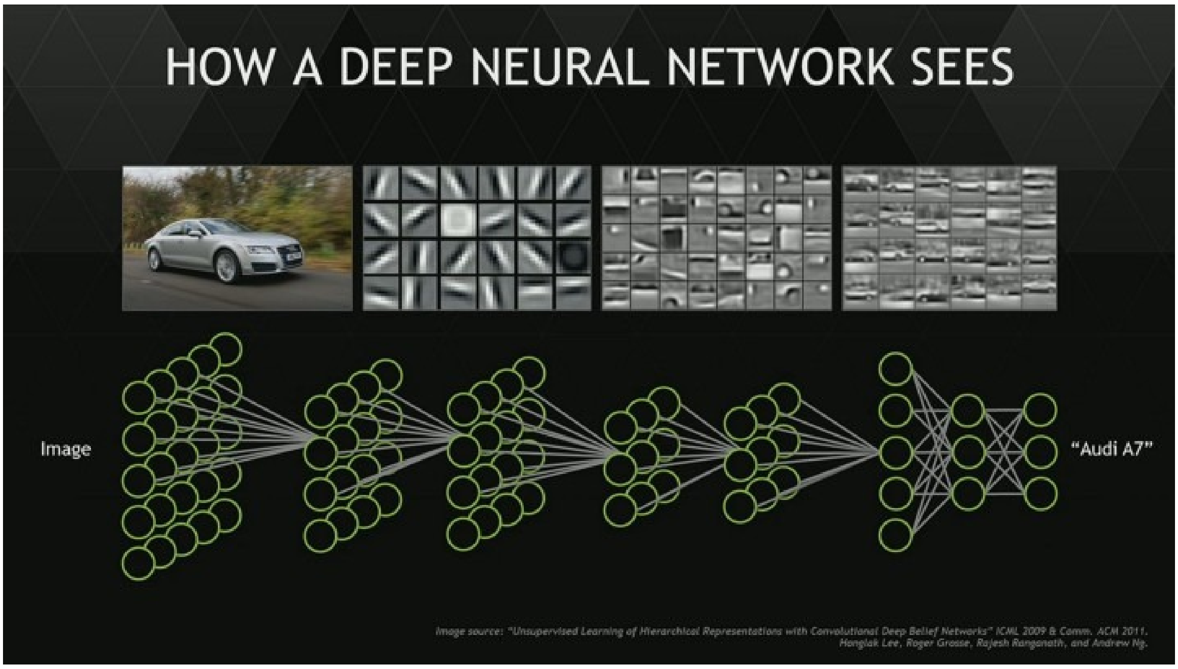

Figure 1: How deep neural network works.

work will be able to recognize very complex pattern, so that the model can get a very high accuracy. The deep CNN-based architecture has been proven as the most accurate model so far. Starting from its bloom at 2012, CNN has been the base of every winning model in the ImageNet, the largest image recognition contest in the world. One of the widely-used deep-CNN model is ResNet-50.

Why resnet ?

ResNet owes its name to its residual blocks with skip connections that enable the model to be extremely deep. Even though including skip connections is a common idea in the community now, it was a revo-

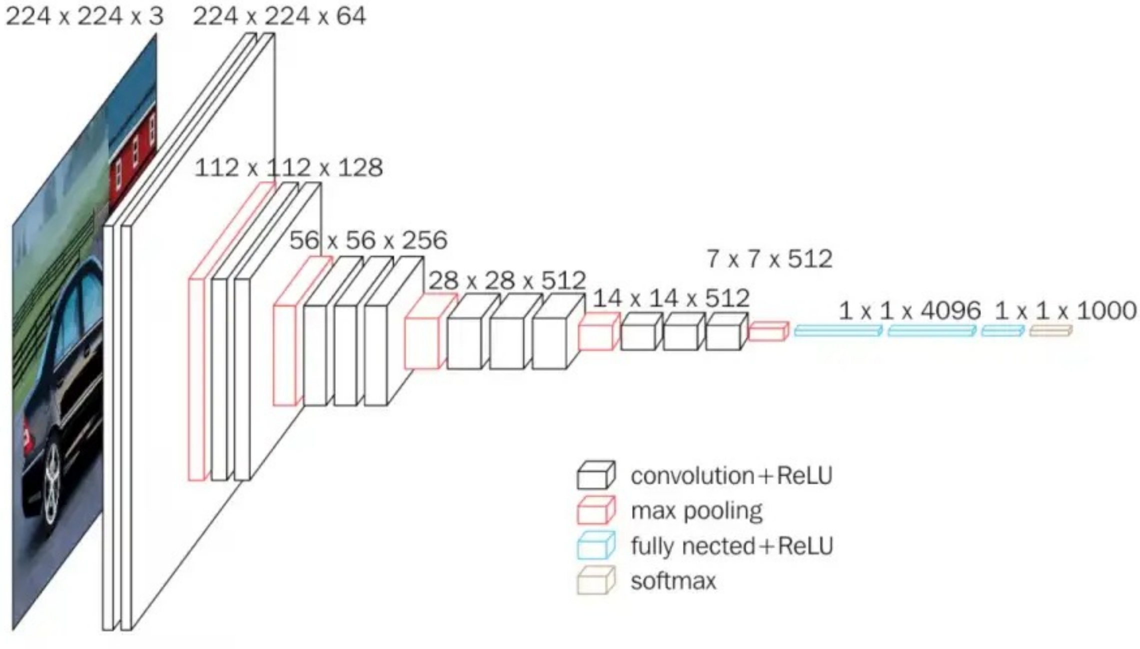

Figure 2: How CNN extract features.

lutionary architectural choice and allowed ResNet to reach up to 152 layers with no vanishing or exploding gradient problems during training.

With the developments in hardware technology and the variety of design techniques in deep learning deeper and deeper, models became popular in ImageNet competition. Unlike LeNet and AlexNet, VGG and GoogLeNet managed to deal with larger structures. However, training deeper networks requires

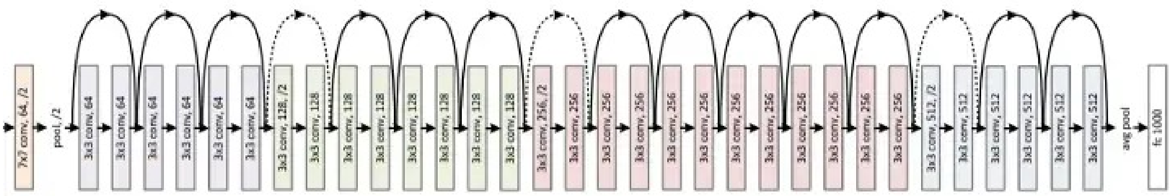

Figure 3: ResNet 34 Architecture (Illustration is taken from the original paper [1]).

some kind of intuition of how gradients flow in the models and a heuristic about how to train models. As models get deeper, research groups needed to push their imaginations and come up with more creative designs. After the Inception module in GoogLeNet, another interesting breakthrough came with the residual learning mechanism of ResNet. Other notable improvements were 1x1 convolutions, dropout layers, and ReLU, yet none of them were as daring as these two.

While it is mentioned in detail in the earlier posts, it is still needed to go through the problem of vanishing or exploding gradients. In gradient-based learning procedures, gradients are calculated in terms of the final loss and the weight space. “In machine learning, an artificial neural network is a model that consists of a directed graph, with weights (real numbers) on the edges of the graph. The parameter space is known as a weight space, and learning consists of updating the parameters, most often by gradient descent or some variant. In the final layer, the loss is partially differentiated with respect to each of the weights, and the weights are updated in the reverse direction of the differentiation in order to decrease the loss. The step size of this update can be constant or adaptive and it is controlled by the learning rate of the training procedure. For the prior layers, the gradient flowing backward from the following layer is multiplied by the input of the current layer. As the gradient flows backward it is multiplied with weight matrices over and over again, which may result in vanishing or exploding gradients. The risk grows as the number of layers increases since the gradient traverses a longer path. Although exploding gradients can be controlled by batch normalization[6] and gradient clipping, vanishing gradients is much bigger of a problem.

ResNet introduces bypass connections in the network and allows the gradient to flow without getting multiplied with weight matrices several times.

In a branched network structure, if a layer leads to multiple modules, the gradients coming from all the modules are summed up and backpropagation continues with the chain rule. In the figure above, if the weight layers tend to have very small numbers (in the order of 10e-5 or smaller) at least some of the gradients would be decreased to a millionth of the gradi-

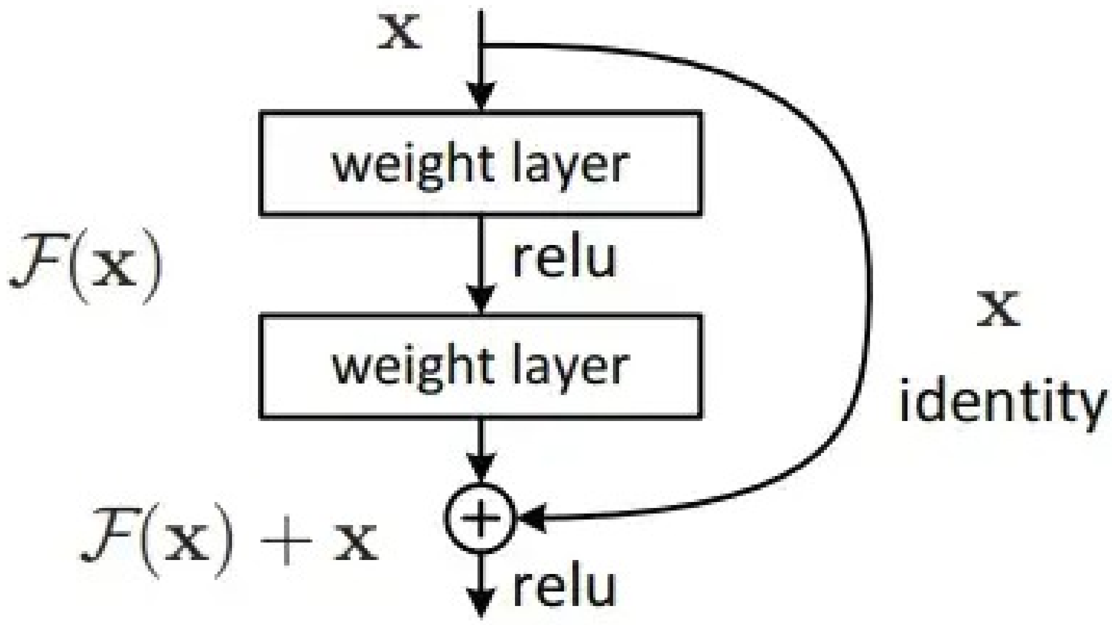

Figure 4: Skip Connections (Illustration is taken from the original paper [1])

ents in the upper layer. But the design of ResNet is providing identity connection by skipping some of the layers, thus making the gradients flow without being subject to any multiplications. That method is a very common architectural choice in recent networks having hundreds of layers.

ResNet with Tensorflow:

Even though skip connections make it possible to train extremely deep networks, it is still a tedious process to train these networks and it requires a huge amount of data. It is also covered in the VGGNet post that, trying to train these kinds of networks with MNIST data may not lead to convergence and acceptable accuracies. ResNet is originally trained on the ImageNet dataset and using transfer learning, it is possible to load pretrained convolutional weights and train a classifier on top of it.

First, needed libraries are imported.

```
import os
import numpy as np
from numpy.linalg import norm
import os
import time
import tensorflow as tf
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.applications.resnet50 import ResNet50, preprocess_input
import math
from sklearn.neighbors import NearestNeighbors
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from sklearn.decomposition import PCA
```

Figure 5: Importing Libraries

Then define the model for features extraction using resnet50:

```
img_size =250

model = ResNet50(weights='imagenet', include_top=False,input_shape=(img_size, img_size, 3),pooling='max')

batch_size = 128
```

Figure 6: Importing Libraries

The script will load the pretrained Resnet50 model, and then truncate the architecture so that we only include the feature extraction part by supplying the include top makes False on the argument. Then, we load each image in the database and preprocess it to the desired 224x224x3 input image. After that, we feed the image to the the Resnet50 network, and extract its feature. Then, the final feature matrix (7x7x2048 in dimension) is flattened to make a 1D-vector.

Building Reverse Image Search Service:

We have already had the vector representation of each image in our database. The vector representation contains some meaningful high-level description that can be used by our machine to capture correlation between one image and the other. So, we are ready to build the reverse image search service.

Now we can find related image using KNN:

K Nearest Neighbor:
K Nearest Neighbor is a supervised machine learning algorithm but it can be used for unsupervised tasks as well. When the testing data comes, its distance is calculated with all the training data points to find the k nearest neighbours which make this algorithm computationally very expensive.

We need a fast and optimized method to find the nearest neighbours. Using a tree data structure can speed up the search process. Three approaches to finding the nearest neighbours for a data point: 1.Brute Force Approach
2.KD Tree and Ball Tree
We've used Ball Tree find the neighbors.

Why Ball Tree?

In the Ball Tree the total space of training data is divided into multiple balls (circular blocks), and the distance of testing data is calculated only with the training points in that block instead of calculating with all the training data points. Ball Tree is built by splitting the complete space into multiple smaller circular blocks.

Consider training data with two-dimensional data (x,y) – (1,2), (2,6), (3,4), (5,6), (7,8), (8,3). Plot it on a graph. Below are the steps to divide the space into multiple parts using the Ball Tree.

1.Select a random data point (5,6)

2.Find a point farthest to that point (1,2).

3.Again find the farthest point to the current point (7,8).

4.Project all the points on the line joining the farthest points.

5.Find the median to divide the space into two halves.

6.Find the centroid in each halve. The centroids are denoted by black dots

7.In each halve, find the point farthest from the centroid and draw the circle. (1,2) is selected as the farthest point in the first ball and (8,3) is the farthest point in another ball. The radius of each circle is the distance from the centroid to the farthest point.

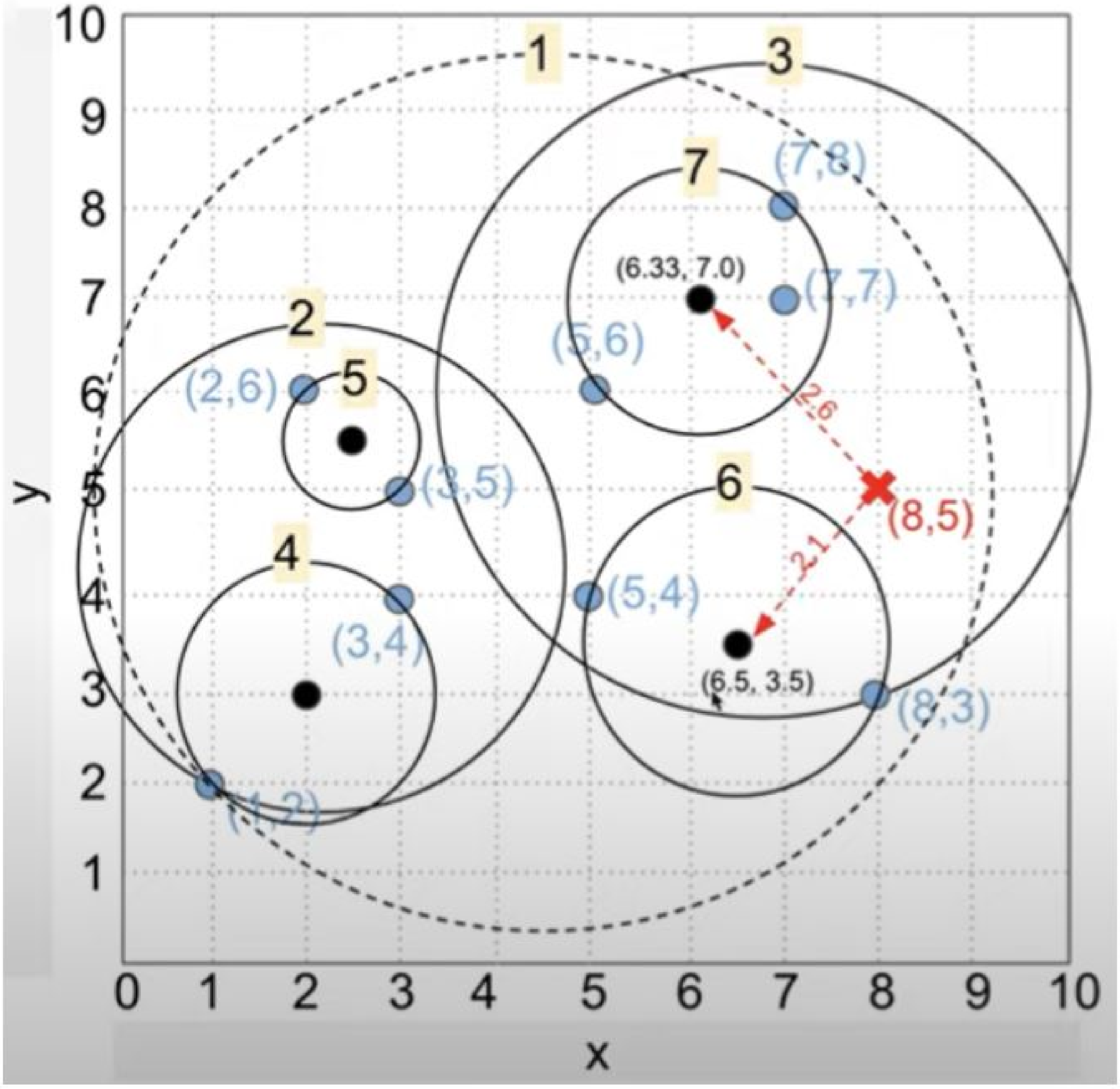

Figure 7: Ball tree features insertion

5. Platform

Frontend:

The front end of a website is everything the user either sees or interacts with when they visit the website. It is responsible for the total look and feel of an online experience. Put simply, the front end is a combination of two different elements: the graphic design (the look) and the user interface (the feel). Each of these is created independently. Users mostly interact with front end.We designed our website using - - HTML - CSS - Type Script

Backend:

5.3.2 Backend The backend, also called server-side, is the infrastructure that supports the front end and is made up of parts of a piece of software regular users can't see. The backend is basically a website's brain. The backend includes the server that provides data whenever requested, and the application that delivers that information about related products.

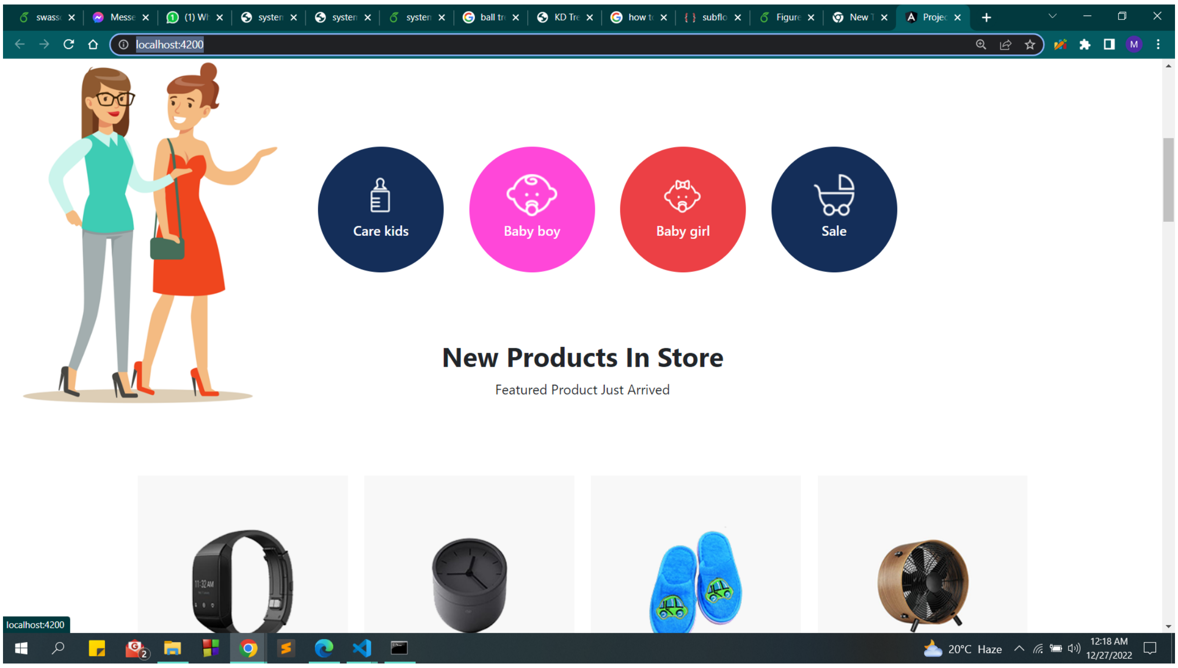

Figure 8: Visual search based website

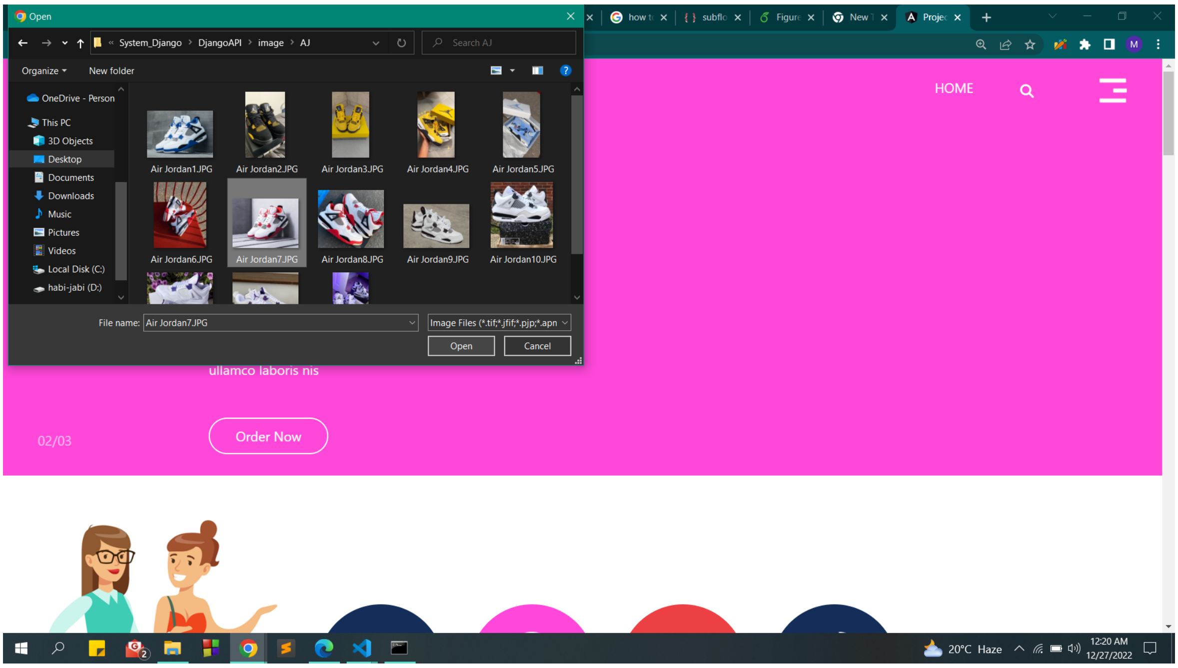

Figure 9: Select image for searching

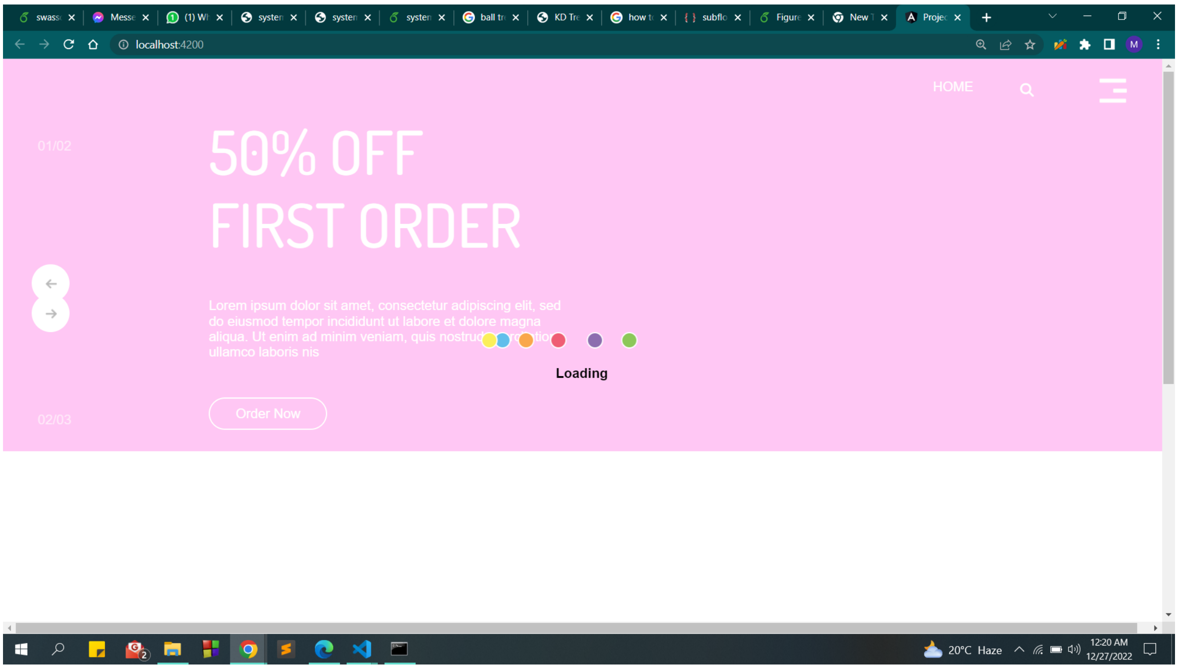

Figure 10: Waiting for feedback

Figure 11: Searched image

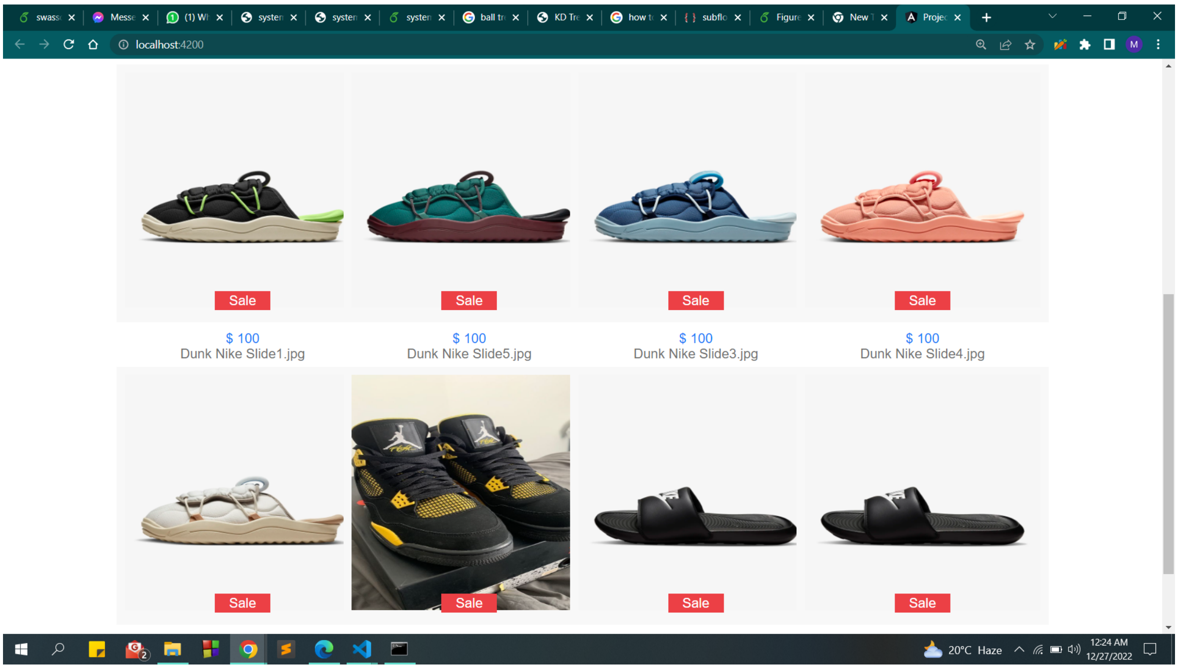

Figure 12: Most related images

6. Performance Evaluation

In case of our reverse image search if user upload a good quality image the result will be much accurate.However the model would give relevant results weather the quality of image are good or not. And user have to take picture of their desired product in a way that the model can understand what he/she is looking for as we have not implemented region detection.

7. References

1.Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun,Deep Residual Learning for Image Recognition,https://arxi

2.https://www.oreilly.com/library/view/practical-deep-learning/9781492034858/ch04.html

3.https://medium.com/swlh/reverse-image-search-using-pretrained-deep-learning-model-83f16ef4aec8

4.https://towardsdatascience.com/tree-algorithms-explained-ball-tree-algorithm-vs-kd-tree-vs-brute-force-9746debcd940