

Software As A Service Model For Computer Vision Based Product Recommender

Rajath C Aralikkatti ^[NITK Surathkal, India] and Sangeeth S V ^[NITK Surathkal, India]

Abstract. Most E-commerce search engines are still largely dependent on keyword matching and the user's knowledge base to find the product that fits most into the customer's requirement. This is inefficient in the sense that the description (keywords used) can vary a lot from the buyer's side to the seller's side. In this paper, we propose adding another layer to the search criteria. This smarter search engine would basically capture an image as the input and try to classify the image into a product description. This can also make searching for a product much faster and easier. Therefore, there is room for improvement in the search process and for making the customer experience smoother. This paper discusses the incorporation of Software as a Service architecture into the development of the computer vision based product recommender.

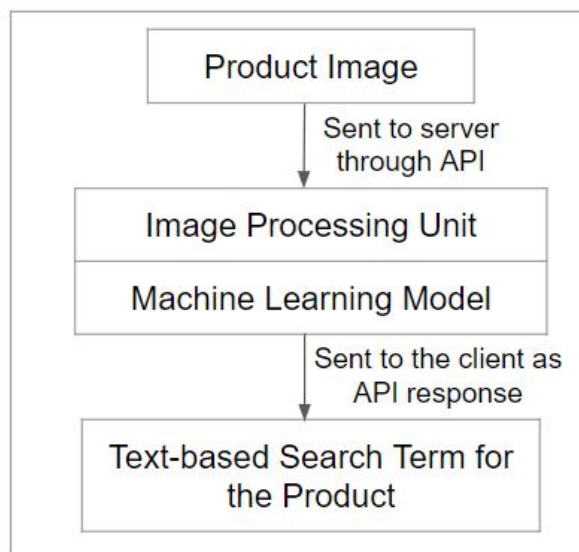
Keywords: Software as a Service (SaaS), Services, Machine Learning (ML), Computer Vision and Image Processing, Artificial Intelligence (AI).

1. Introduction

The E-commerce system is fast-evolving and online shopping is rapidly becoming an unavoidable part of our daily lives. The latest challenge that online shoppers face is the sheer amount of digital information. This explosive growth creates an information overload, which in turn, inhibits timely access to items of interest on the Internet. Herein lies the need for a recommendation system. Almost every E-commerce website has its own implementation of a recommendation system based on available data such as recent searches, prior purchases, etc. The aim of such a system is to give the customer an efficient and more personalized experience. However, most recommendation systems are text-based and usually rely on keyword matching systems and the user's knowledge base. Moreover, the text-based description of a product can vary a lot from the buyer's side to the seller's side. With the rapid development in image processing technologies owing to the improvements made by neural networks these recent years, we can now make a shift from traditional word-based searching methods to searching methods involving visual similarity.

Our service makes it possible to search for a product from its image. A machine learning model produces the search term for a product using its image as an input. We build an Application Programming Interface (API) that takes in an image as a request and produces the text-based classification as the response. This API can be used by our clients and can be incorporated into E-commerce applications as an additional feature for their searching tool.

A simple flowchart showcasing the major components of the computer vision based product recommendation system is shown below:



2. Software As A Service

Software as a service (SaaS) is a cloud-based service where instead of downloading software for your desktop PC or mobile device to run and update, you instead access an application that is deployed on a hosting service and is accessible via Internet. Here are some of the characteristics of SaaS service model:

- ❑ SaaS makes the software available over the internet.
- ❑ The software applications are maintained by the service provider.
- ❑ They are highly cost effective since they do not require any maintenance on the user's end.

- ❑ They can be automatically upgraded and updated.
- ❑ All users use a shared data model. Multiple users can therefore make use of a single instance of infrastructure.



Despite all its advantages, SaaS has some risks associated with it as well. The major reason why those considering SaaS are hesitant is because of the security concerns. Many people are not comfortable with having their data and applications stored off premise and not being in direct control of the security. Outages on the cloud service provider's side are also a big issue. The other concerning factor is that performance and efficiency is directly based on the user's internet connectivity.

3. The Image Based Product Recommender As A SaaS System

The Image based Product Recommender can be implemented as a SaaS model by incorporating the following ideas. The major functionality of the image based product recommender is to produce a product description or search term given the image of a product as input. This is achieved with the help of a machine learning (ML) model that maps the given image to its product class. While it is possible to run the ML model on a client's device it would be more practical as a SaaS system because of how cumbersome setup and updation are with an on-premises model. This is because the client may not have adequate hardware to efficiently run the ML model. The ML model is also constantly updated to improve performance as we collect more data and new groundbreaking ML architectures are developed.

One of the main advantages of a SaaS implementation is that the lack of computational resources on the client's side is no longer a restraint. For a better performing Image Processing Unit we may need to run heavy ML models which need to be run on powerful GPUs that may not be available on the client's side. A SaaS design overcomes this issue as the computation involved is done on our servers. This makes our service more accessible as we can now reach those clients who would otherwise not be able to use our service in the case of an on-premise model.

Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud platform. It has more services than any other cloud provider. It has services for artificial intelligence and offers a lot of functionality. For these reasons, AWS is the platform chosen to deploy our own Image Classification service.

A possible way to deploy the image classifier in AWS is to design an API that takes in an image as a request and produces the text-based classification as the response. The image can be sent to the API in the body of a PUT request to the image classifier instance. The API design must be able to properly decode the request data into the classifiers input format. The text based classification can be sent as part of the body of the response. Now, because the API does not depend on anything other than the input request, we can utilize it in multiple ways without having to change the classifier's implementation. For example, a website could be made that allows the user to upload an image which will then be used to call the API. A mobile or desktop application can be made just as easily just by accommodating an API call.

4. A Study Of SaaS Features Used

The image classifier can be modelled using a Software as a Service (SaaS) architecture as shown in the previous section. The following points talk about how we incorporate several important features provided by the SaaS architecture into the Computer Vision based Product Recommendation service.

1. Multitenant Architecture

All users and applications that depend on the image classifier interface share a single, common infrastructure and codebase that is centrally maintained by the cloud provider. We are able to save valuable development time, previously spent on maintaining numerous versions of outdated code.

2. Easy Customization

We can easily customize how the image classifier is invoked.

For example, we can very easily provide a website in which a user can upload an image to get its classification from the image classifier service. We can, just as easily, make an API that sends data as part of the HTTP request and gets the text based classification as the response. This also means that we can make changes to the image classifier very easily with much less customer risk and less cost.

3. Subscription / Pricing Model

We can easily implement a pricing strategy not unlike many APIs on the web currently. For example, the [openweathermap.org](https://openweathermap.org/api) is an API that has several different pricing packages available. 60 API calls per minute has no charge and they proceed to increment the price linearly with the number of API calls. This can be implemented in our model also by providing a unique API key to registered users. The SaaS architecture thereby provides better access to our application while making it easier to manage privileges and data use.

4. Security

AWS provides features like Security Groups wherein we can configure inbound and outbound rules. These can be used to restrict incoming and outgoing traffic. AWS provides asymmetric encryption for accessing the AWS instance for backend configuration. This ensures that only the people who have the required key will be able to access the AWS instance (the image classifier model). By using AWS as the cloud provider, we are instantly able to provide a significant amount of security with minimal effort on our part.

5. Scalability

AWS provides the ability to dynamically scale up and scale down the service. If it turns out that our service is being utilized more than we expected, it can very easily be scaled up. Rules can be configured to deploy more instances if the incoming traffic crosses a specified threshold.

5. Conclusion

In this report, we have presented our earlier approach for the image based product recommender in a Software as a Service architecture. Several key changes were made to the earlier approach, so that it could be provided as a service.

6. References

1. S. Amershi et al., “Software Engineering for Machine Learning: A Case Study,” Proc. - 2019 IEEE/ACM 41st Int. Conf. Softw. Eng. Softw. Eng. Pract. ICSE-SEIP 2019, pp. 291–300, 2019.
2. L. Chen, F. Yang, and H. Yang, “Image-based Product Recommendation System with CNN,” <http://cs231n.stanford.edu/reports/2017/pdfs/105.pdf>, p. 2015, 2015.
3. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., pp. 4510–4520, 2019.
4. D. Kang, D. Raghavan, P. Bailis, and M. Zaharia, “Model Assertions for Monitoring and Improving ML Models,” <http://arxiv.org/abs/2003.01668>, 2020.
5. S. Bell and K. Bala, “Learning visual similarity for product design with convolutional neural networks,” ACM Trans. Graph., vol. 34, no. 4, 2015.
6. T. Diethe, T. Borchert, E. Thereska, B. Balle, and N. Lawrence, “Continual Learning in Practice,” <http://arxiv.org/abs/1903.05202>, no. Nips, 2019.