Computer Vision based Product Recommender using a Model Driven Design

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 a model driven design approach to the development of the computer vision based product recommender.

Keywords: Model Driven Design (MDD), Machine Learning (ML), mobile application, 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 computer 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.

Similar to the way machine learning has impacted how recommendation systems are being built, advances in machine learning have also stimulated widespread interest in incorporating AI capabilities into software and services in other domains. To be able to effectively build such applications new development processes must be employed as traditional development processes cannot be used owing to the many differences we encounter upon incorporating machine learning into an application.

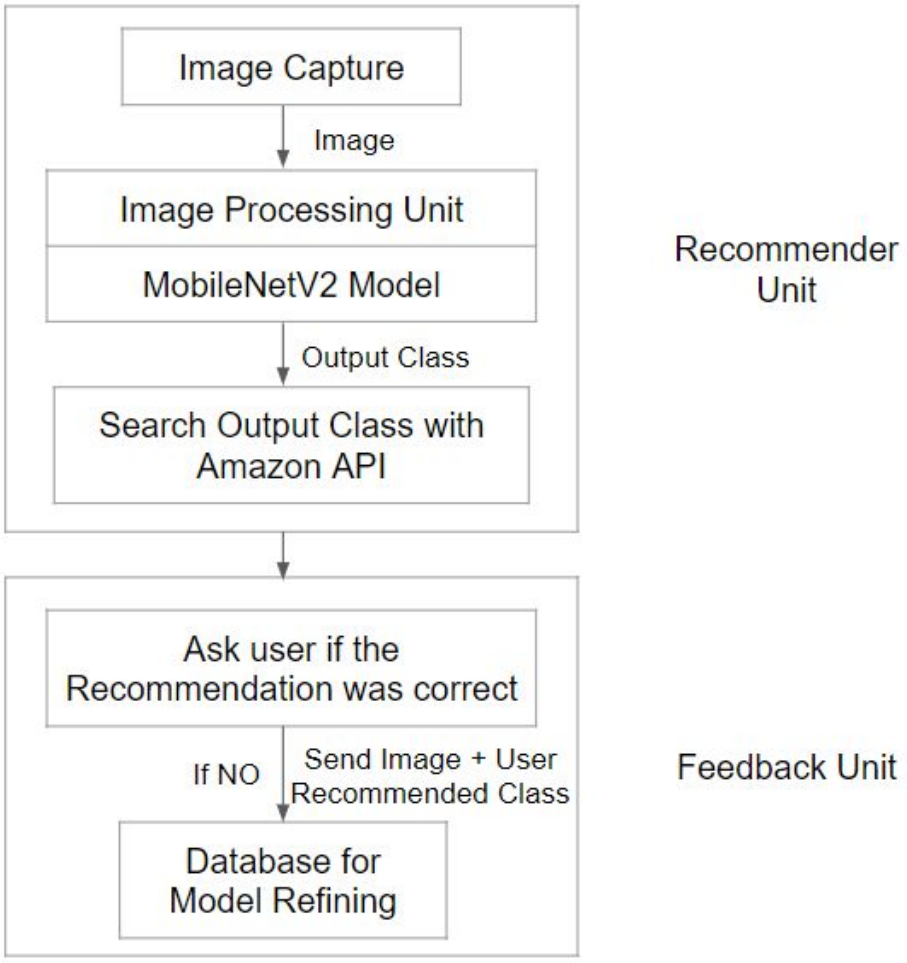
Our product is a mobile application with a fast and interactive UI that classifies an image into its corresponding product class, using machine learning, which is then used as a search query in Amazon to list the products. We achieve this computer vision task of mapping the image of a product to its product class by using MobileNetV2 as the architecture for the machine learning classification model. We run the model on device and hence the computational resources of the device becomes an important consideration. MobileNetV2 is able to retain similar accuracies as larger models while minimizing the number of operations making it the apt choice.

In machine learning the performance of a model is strongly tied to the kind of data it is trained on. Hence it is always good to have as much data as possible. This makes data collection both during the initial stages and user feedback stages very important. Our application is also built with making the data collection during the user feedback stage an important focus. We collect the image captured and the user-suggested product class as feedback from the user when our application makes a wrong recommendation.

The product does the following,

- ❏ It requires access to the camera of the mobile device used in order to get the image needed.
- ❏ It captures the image and feeds it to the Image Processing Unit (IPU) working in the backend.
- ❏ The IPU processes the image and outputs the text-based classification of the captured object.
- ❏ Then, feedback is collected and used to improve the model.

A flowchart showcasing the major components of the image based product recommendation system is shown below:



2. Model Driven Design

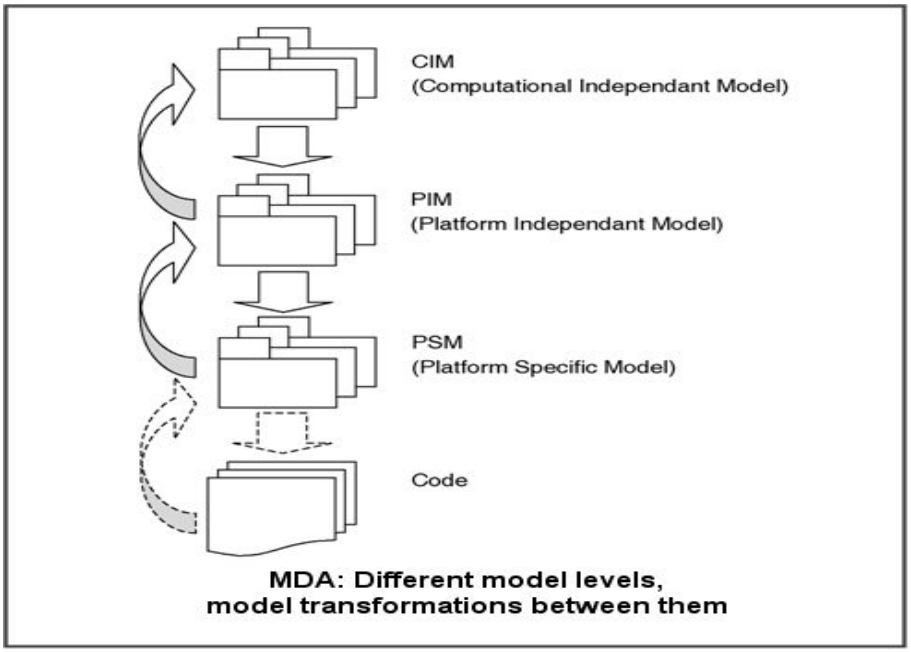
Model Driven Design (MDD) is described by the Object Management Group (OMG) as the separation of business decisions from the software oriented decisions by making use of the interoperability of formal models. Basically, MDD uses *models* (may be conceptual, logical, physical, etc) to describe the functionality of the *solution,* independent of any software-oriented decisions.

In Model Driven Architecture (MDA), platform independent models are initially expressed in a platform independent modelling language. The model is described according to the requirements specification, and this model is independent of the programming language which will eventually be used to implement the software. The result source code can be specified while the model is being developed or after the model is fully developed. Either way, the model always takes the priority.

Another important feature of the Model Driven Architecture is that changes are always bidirectional, ie, changes made to the model will have to be reflected in the implemented code and vice versa. The end product will be platform specific code that reflects the described model in terms of functionality as well as conceptual consistency.

According to the Object Management Group (OMG), the model driven architecture has four model levels:

- CIM: Computational Independent Model
- PIM: Platform Independent Model
- PSM: Platform Specific Model
- Code



MDA: Different model levels, model transformations between them

One of the benefits of model driven architecture is that the first two model levels are completely independent of the platform. So, if the need arises for our software to be migrated to another platform, we need only alter the platform specific model (PSM). The model-driven approach to application development is centered around the abstraction away from code to form a visual model. This makes the development and use of the application much easier.

Model driven design is also much less error-prone because we ensure that at every step, the model and the implementation are consistent. MDD ensures that each component in the application will be properly acceptance tested, ie, the functionality of the application will be in line with what is specified in the model. Thus, MDD leads to meaningful validation as well.

MDD ensures that the documentation of each and every component is up to date. In the current scenario, many products are facing the need to migrate to the cloud architecture. A model driven design is less sensitive to such changes. The only thing that could change during this step is the code generator/interpreter. After this change, all application models can directly be transformed into code under the cloud architecture also.

So, in conclusion, MDD is one of the best development processes out there. It completely ensures the validation and testing requirement of each and every component. It is also well documented and less sensitive to platform changes. By developing the models as a visual representation of the functionality, we are able to bridge the gap between business and IT.

3. The Image Based Product Recommender As A Model Driven Design

The image based product recommender can make use of the Model Driven Design architecture for its development. It enables rapid, collaborative application development. Because it uses visual modelling techniques to define data relationships, to build user interfaces and to process logic, model driven software development enables both developers and business users to rapidly deliver applications.

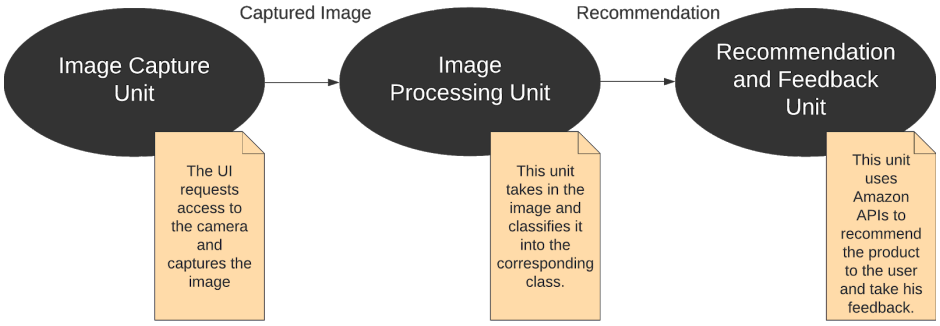
There are two core concepts associated with model-driven design. They are abstraction and automation. The software application model is defined on a higher level of abstraction and is then converted into a working application using automated transformations/interpretations. This removes the need for generating/writing code manually.

The three stages of MDD as specified above (by Object Management Group) are:

1. **Computation Independent Model (CIM)**

   The computation independent model is a model defined in the Object Management Group's Model-Driven Architecture as a primary model. This model reflects the requirements from a business perspective. It contains the basic roles and constraints of each unit of the application.

   The image based product recommender will have mainly 3 units. The image capture unit, the image processing unit and the recommendation and feedback unit.



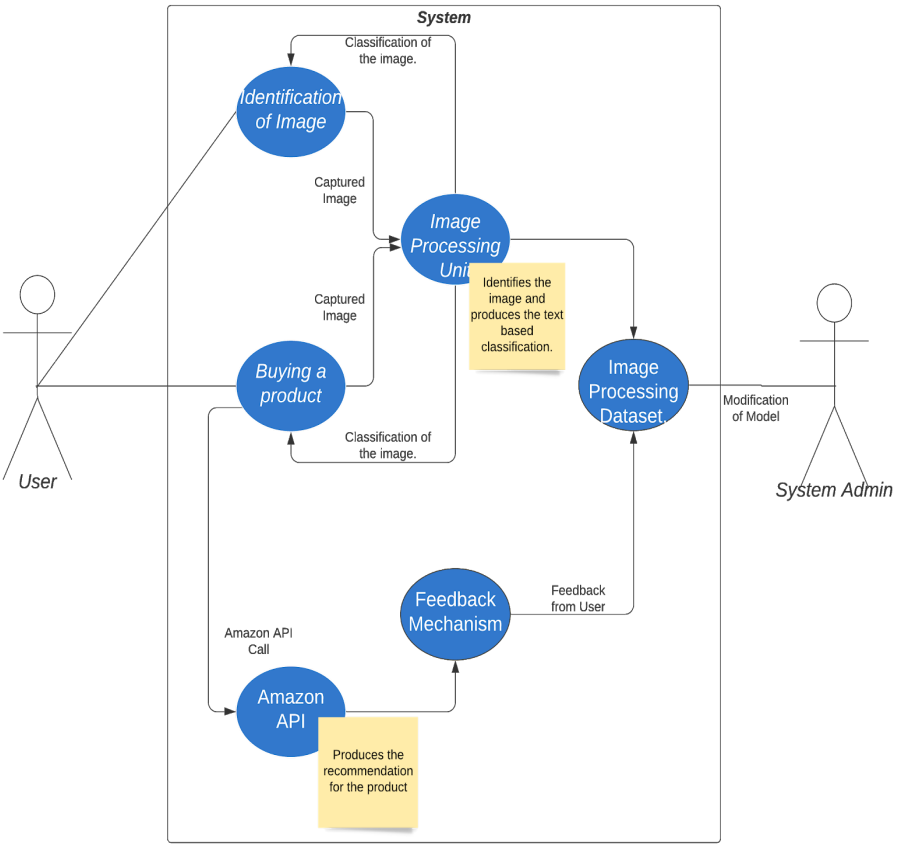
2. **Platform Independent Model (PIM)**

   The platform independent model focuses on the high level business logic without considering the features of the implementation technology of the system. It will contain no reference to the underlying technological platform.
   The image based product recommender performs the following tasks:

   - ❏ This is the unit that the user interacts with. It captures the image and feeds it to the image processing unit. This makes use of the camera of the device in order to capture the image.
   - ❏ The image classifier processes the image and outputs the classification of the captured image which is then used in conjunction with Amazon APIs to produce a list of products for the user.

- ❏ The application then collects the feedback which is then used to refine the model at the Admin's convenience.

USE CASE DIAGRAM



3. Platform Specific Model (PSM)

A platform specific model is a software or system model which defines the functionality in a manner specific to a particular implementation platform. For our purposes, it is possible to make the application platform independent, as shown below, using the Flutter SDK.

In order to upgrade the platform independent model to the platform specific model, google's Flutter SDK is going to be used. The advantage of using this tool is that it can build an application for any platform we specify, be it, android, ios, web or even a desktop (can be any operating system, Windows, Mac, Linux, etc. are all supported). The codebase is the same for all these platforms and it is written in the object oriented programming language, dart. With minimal coding on the developer's part, a fast and interactive application can be made.

The tensorflow-python model will be used for the Machine Learning (ML) part that deals with the image processing and recommendation. The model will be run on the device using the *tflite* package in flutter.

4. Conclusion

In this report, we have presented our earlier approach for the image based product recommender in a Model Driven Design architecture. Several key changes were made to the earlier approach, so that a model driven design could be adopted.

5. 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.