Software Requirements Specification

for

Computer Vision Based Product Recommender

Version 1.0 approved

Prepared by
Rajath C Aralikatti, 181CO241
Sangeeth S V, 181CO246

NITK Surathkal, Karnataka.

9th September, 2020.

Table of Contents

1. Introduction

Most online shopping search engines largely depend on knowledge base and proper keyword specification as their search strategy to find the product that the customer wants. While very efficient search engines exist, the inefficiency arises due to the difference in the description of the product on the buyer's and the seller's side. So, there is a need for visual similarity to be taken into account during the recommendation process. Hence, an image based product recommender will improve the customer's experience by making the task of searching for products much easier.
The software is a mobile application that provides a clean and user friendly interface to all its users.

1.1 Purpose

The computer vision based product recommender aims to adapt and if possible try to improve on some of the existing software engineering methodologies in the field of image processing and mobile application development. The android application recommends products based on the images taken and, using image processing technologies, it identifies the classification of the item and searches for it on amazon. It also seeks to establish a feedback mechanism that will be used to refine the model working in the backend. In our approach, the user can capture any image which contains the product.

1.2 Intended Audience and Reading Suggestions

The product aims to produce a fully functional prototype for the image based product recommender. It has no usage restrictions. It can be utilized by anyone who wishes to make use of it for e-commerce purposes. The users are requested to kindly provide feedback on the recommended items so that the Image Processing Unit (IPU) in the backend can be improved accordingly.

1.3 Product Scope

The software that this document specifies, aims to make the e-commerce experience convenient and more easy to use. The system is based on an underlying machine learning model that uses several image processing techniques to classify an object and then search for it on Amazon. The machine learning model would be able to search for consumer items that are commonly searched for on e-commerce websites from images that are taken. Above all, we hope to provide a smooth and comfortable user experience.

2. Overall Description

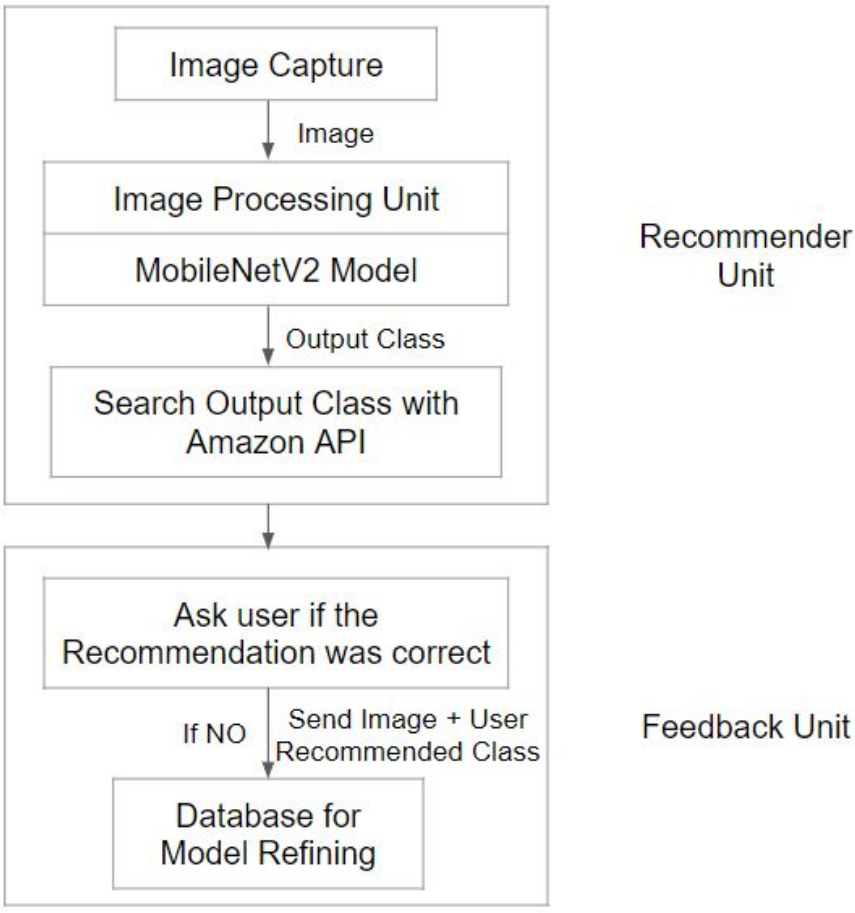
2.1 Product Perspective

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 uses the MobileNetV2 which uses convolutional neural networks optimized to run on mobile devices to process these images.
- ❏ The IPU processes the image and outputs the classification of the captured object.
- ❏ Then, feedback is collected and used to improve the model.

2.2 Product Operation Flowchart

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



2.3 Operating Environment

The operating environment for the image based product recommender is as listed below,

- ❏ Platform-independent mobile application - Android 8+ / IOS
- ❏ Frontend: Flutter/ Dart
- ❏ Backend: Firebase
- ❏ Image Processing Unit: Uses MobileNetV2 as the architecture for the ML model

Why flutter?
Flutter is a free and open source Google mobile UI framework that provides a fast and expressive way for developers to build native apps on both IOS and Android. It facilitates fast development, expressive and flexible UI and very high performance apps.

Why MobileNetV2?
The IPU is run on device and hence networks that require high computational resources cannot be used. MobileNetV2 pushes the state of the art for mobile tailored computer vision models,by significantly decreasing the number of operations and memory needed while retaining the same accuracy, making it an apt choice for the machine learning architecture used.

2.4 Design and Implementation Constraints

Language requirements include dart (for flutter development), python and tensorflow for training the machine learning model. The primary focus is to demonstrate the use of software engineering methodologies in the field of artificial intelligence and to adapt them to the design and development of this mobile application. While we have implemented a feedback system to collect new data for refining the ML model it may not be possible to practically demonstrate an improved model as we can only gather limited data due to the fact that we do not have enough users. Retraining the model with limited extra data will not make any significant changes to the performance of the IPU.

2.5 Assumptions and Dependencies

The image based product recommender needs camera access for capturing the input image. On-demand access to the amazon API is required for the successful working of this app. If the recommendation made by our application is incorrect, the user recommended class for the product along with the image taken is saved onto our database as part of the feedback mechanism. For this, internet connectivity is required for uploading the image and its suggested class. The product is designed for Android 8+ / IOS systems.

3. System Features

The functional requirements of the product are:

- ❑ Image Capture
- ❑ Image Processing Unit
- ❑ Recommendation and Feedback

3.1 Image Capture

3.1.1 Description and Priority

The image capture is the first step in the working of our application. The image captured is then fed to the image processor for further computation. It is of the highest priority because it is the first step in the working of our mobile application.

3.1.2 Stimulus/Response Sequences

The captured image is previewed after the picture is clicked and the option to retake the image or proceed is provided. This feature can be reused until the user is ready to proceed. Once the user opts to proceed, the captured image is fed into the image processing unit.

3.1.3 Functional Requirements

To carry out the services provided by this feature, a fully functional camera is required to capture the image and then feed it as input to the image classifier..

3.2 Image Processing Unit

3.1.1 Description and Priority

The image processing unit takes in the captured image and determines the class to which the object in the image belongs to. It uses MobileNetV2 as the architecture for the machine learning model which is run on device.

3.1.2 Functional Requirements

To return the product class to which the object in the image belongs to. This product class serves as the search keyword to the Amazon API used in the next stage.

3.3 Recommendation and Feedback

3.3.1 Description and Priority

Once the image processing is done, it will output the classification of the object based on the image taken. This recommendation is then used in conjunction with Amazon APIs to showcase the search results for the corresponding item.
Then, the user is asked for feedback in regards to whether the recommendation is correct or not, which is then used to refine the model working in the image processing unit.

3.3.2 Stimulus/Response Sequences

The image processing unit outputs a classification of the captured image which will redirect the app to an item catalog. The user can then proceed with the item if he is satisfied with the recommendation. He can return to the app and the feedback window will be available. If he is not satisfied with the recommendation, then an option to identify the actual product. In this case, an automated text-based suggestion is provided.

3.3.3 Functional Requirements

To carry out the services specified, extensive use of Amazon APIs is required. A dictionary of product classes is also maintained which is used to auto-complete the user-suggested product class when the feedback is asked. This user-suggested class along with the image taken is saved onto our database so that this new data can be used to refine our ML model later on. Moreover, internet connectivity is required for uploading the image and its suggested class in the feedback mechanism.

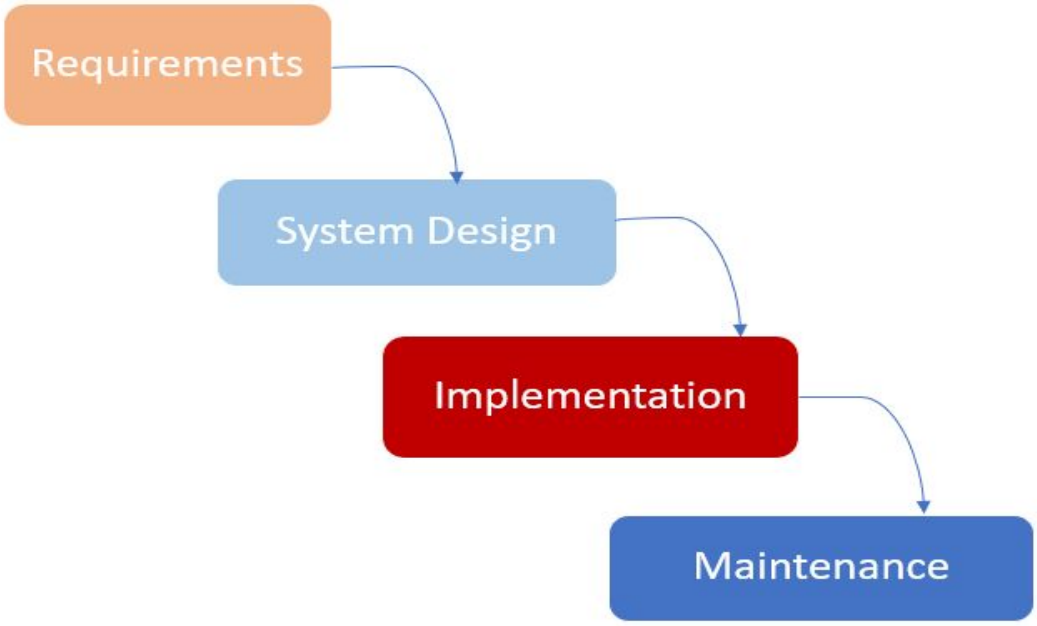
4. Software Development Life Cycle

We have chosen the “Waterfall model with Back-flow” as our Software Process Model. Waterfall model is a linear sequential model. Since our requirements are clear, concise and unambiguous, this model is the apt choice for our project description.

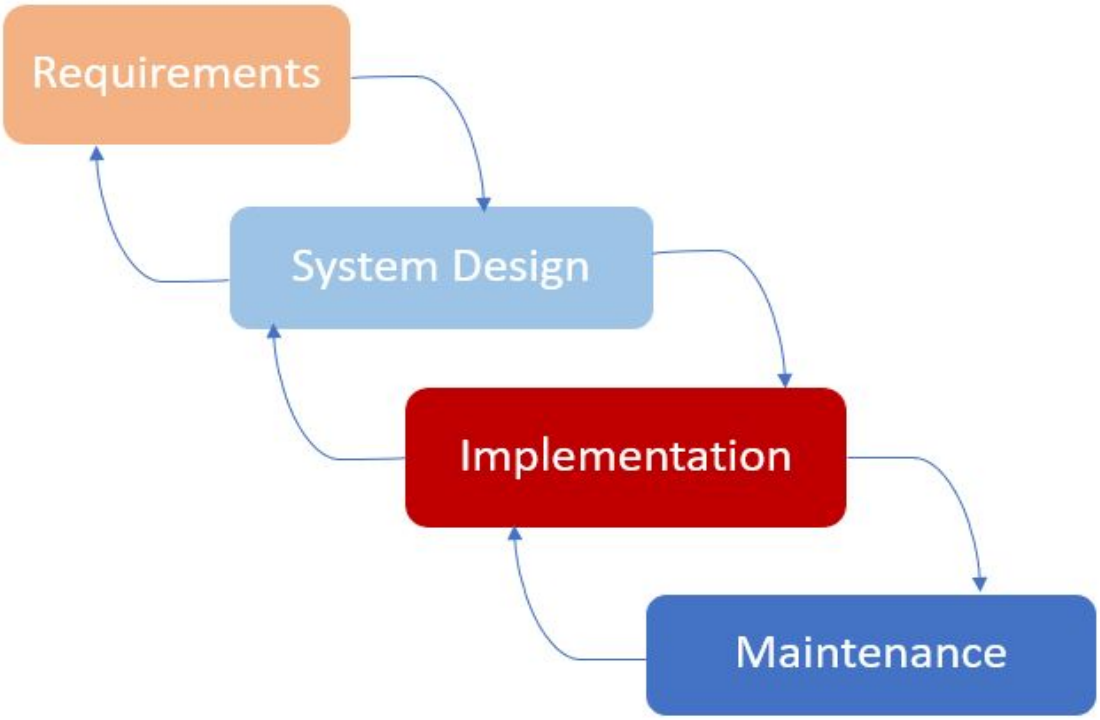
We have chosen the waterfall model with back flow for the following reasons:

- Waterfall model is intuitive and easy to implement in our project. This helps any users and developers understand each phase of the Software Development Life Cycle (SDLC), in neat independent phases. Since the phases do not overlap, we only need a backflow to correct any mistakes that may occur.
- Waterfall model is shown to perform exceedingly well for projects which are short and small in size and scope. Our problem statement fits the description. Therefore, this SDLC is a good way to proceed.
- Waterfall methods are shown to work for projects where requirements are well understood and stable, i.e, they don’t change over time. Our requirements are unambiguous and fixed. Therefore, we can proceed with the waterfall model.
- Feedback is an essential part of our project. Even if the requirements and planning phase is done exceptionally well, some sort of feedback is needed. For this reason, we have chosen a feedback system as “Backflow” which is a slight modification to the original Waterfall model. This gives us the advantage and convenience of visiting any previous stage to correct deliverables / expectations if deemed necessary.

Waterfall Model



Waterfall Model with Backflow (Feedback)



5. Timeline

	Week 1 September 13-19	Week 2 September 20-26	Week 3 September 27-October 03	Week 4 October 04-10	Week 5 October 11-17	Week 6 October 18-24	Week 7 October 25-31	Week 8 Novemeber 01-07	Week 9 November 08-14	Week 10 November 15-21	Week 11 November 22-30
1	Analyze existing applications related to image based processing.										
2	Prepare software requirements document and detailed timeline of the project.										
3			Basic UI development and procurement of data set for training the image processing unit.								
4				Implementation of a working prototype using the image captured and external amazon APIs to produce the recommendation system.							
5						Prepare version 2 report with several screenshots and use cases.					
6								Implement the feedback mechanism and perform several improvements in the UI and the model.			
7										Preperation of final report for the image based product recommender and possibly, deployment of the app.	

Appendix A: Glossary

The document uses the following abbreviations.

SRS	Software Requirements Specification
IEEE	Institute of Electrical and Electronics Engineers
IPU	Image Processing Unit
SDLC	Software Development Life Cycle
ML	Machine Learning

Appendix B: List of 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.