Shoppick

Requirement specification

Student Number	Name
2013310500	Heesung Kim
2014312385	Dasan Jeong
2016315382	Seunghoon Lee
2017312668	Hyunjee Shin
2019319366	Misael Zapata

Contents

1. Preface

1.1 Objectives

In this section we summarize the structure and the specifics of this document on the level that each groups of expected readers will understand.

1.2 Readership

This is divided into two parts user requirements and system requirements according to the readers. User requirements, often referred to as user needs, describe what the user does with the system, such as what activities that users must be able to perform. System requirements are the building blocks developers use to build the system.

A. User requirement

User requirement considers are statements in natural language plus diagrams of the services the system provides and its operational constraints. Written for system end-user, client-manager, client engineers, contractor-mangers, system architects.

B. System requirement

System requirement is a structured document setting out detailed descriptions of the system's functions, services and operational constraints. Defines what should be implemented so may be part of a contract between client and contractor.

1.3 Document Contents

This document is constituted in eleven categories. Categories include Preface, Introduction, Glossary, User Requirement Definition, System Architecture, System Requirement Specification, System Models, System Evolutions, Appendix, Index, Reference. Each category is as followed.

A. Preface

Preface talks about the expected readers, the abstract structure of this document and the summary of each categories.

B. Introduction

Introduction is about the value of our system 'Shoppick'. It includes the needs of the customers and what difference and strength our system has compared to other systems/software. It also presents the expected effects of the software.

C. Glossary

Glossary will be explaining the technical terms this document will be using. It is target to readers without any professional knowledge.

D. User Requirement Specification

The user requirement specification talks about the service 'Shoppick' will be offering. Functional and non-functional requirements will be both described. It will be explained in natural language on the level that even the readers without any background knowledge can understand the concept. The part where it is hard to

explain with the natural language will be replaced with objects such as diagrams or tables.

E. System Architecture

System architecture will be showing the abstract summary of the architecture of 'Shoppick'. It also describes how the functions consisting the system is distributed.

F. System Requirement Specification

System requirement specification will be a more detailed version of user requirement specification. It also shows the detailed scenarios of the process resulted while using 'Shoppick' for the sake of easier understanding and concrete idea of the system.

G. System Models

System models present the internal and external relations between the system and the components. Especially it is focused on the inputs and outputs and the operation process of the database. Most of the explanations will be made by using objects.

H. System Evolutions

System evolution is about the presumptions made after the technological advancements. This helps the system developers assume the changes that can be made in the future.

I. Appendix

Appendix will be describing the back-end process along with most details that has be omitted from the contents above.

J. Index

The index of the objects used in this document

K. Reference

References used while developing 'Shoppick'

2. Introduction

2.1 Objectives

This chapter explains the needs and how 'Shoppick' outperforms other systems on the market.

2.2 Needs

A. The advertisement of fashion brands

Fashion has been around for centuries and since the start of the industrial age when clothes have been manufactured in enormous volumes, clothes have been something people has been using to define themselves. There has been increasing amount of fashion brands and designers making their own clothes up to recent days and this is evidently a forever growing market.

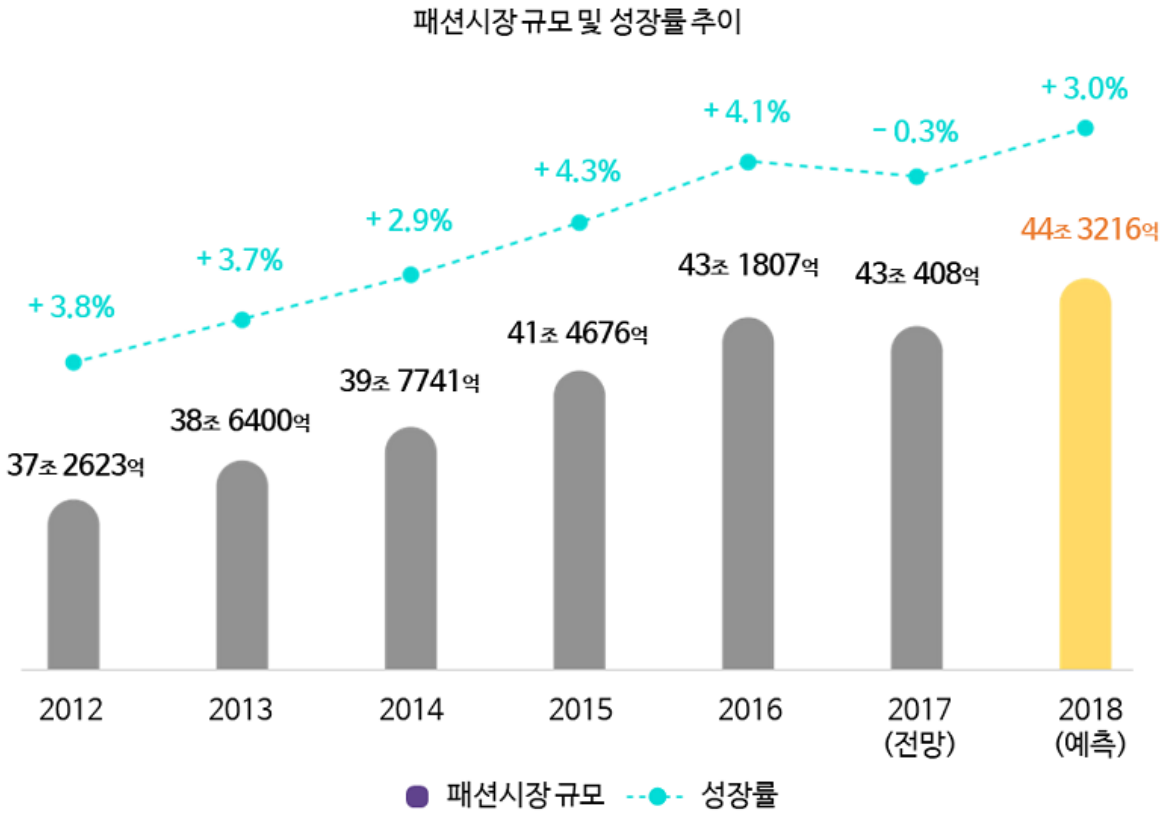


Figure 1. The growth rate of fashion market

Consumers nowadays have so many different brands of clothes to choose from, so it has been made essential for the brand/shop owners to do an excelling job on marketing or advertising their clothes. Direct advertising has been the mainstream until media developed and now companies are using indirect advertising way more than the direct ones. These advertisements have been started from magazines to celebrities coming out on TV shows with their clothes on and in recent days to influencers in social network services.

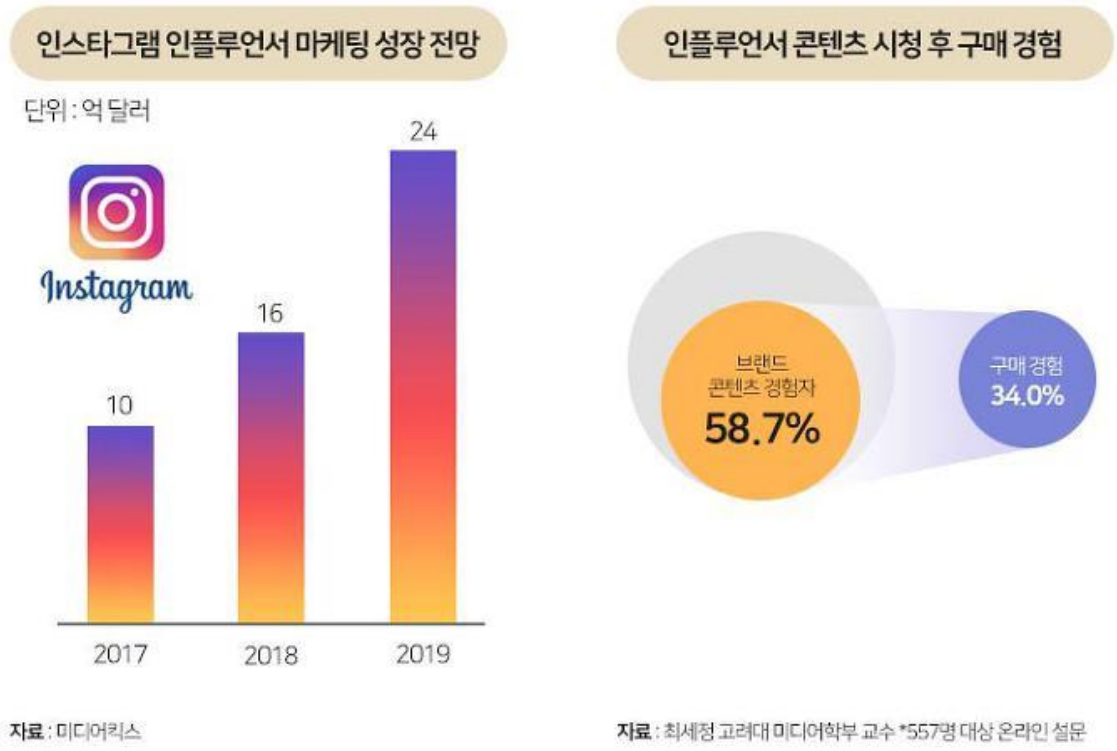


Figure 2. The effects of influencers

B. Problems of advertisements for companies

As much as fashion brands exist in the world so does advertisements. However, while there are variety of brands making similar clothing, consumers can have difficulties to find the exact clothing that was advertised on media. It would be a loss for the companies that the money put on advertisements aren't coming back as profits.

C. Problems of advertisements for consumers

Consumers who are interested in fashion tend to get curious about what their bias celebrities and influencers are wearing. It is not often that we meet with a question on the internet what specific clothing someone is wearing in that show or post. If someone doesn't answer for the companies, it is a loss of a potential customer. Consumers and fashion enthusiasts need a system that they can use real-time.

D. Shopping methods

There are so many shops in the world outdoors and on-line. It is hardly possible for people to find what shops fit their style. It is time consuming and just might end without any profit.

2.3 The resolution: Shoppick

Picture 1. Shoppick logo

Shoppick has two major functions.

1. Finding the exact or the closest clothes to given picture
2. Recommending on-line shops according to the user style information gathered through the SNS

Shoppick can find the closest clothes to the picture taken or given by the users. First Shoppick holds clothing information from the sellers and makes a huge database. Later whenever the user uses the service matches them with the pictures given from the customers.

When a user first logs in to Shoppick, they must log-in with their SNS account and is recommended to use the account they use the most. Shoppick also requests permission to use the pictures uploaded in the user's account. With the data Shoppick forms up to three style tags and saves it. Then the user is recommended with the tags matching their individual tags.

2.4 Expected effects

A. Effects to the companies

Companies wouldn't be wasting money on their advertisement and doesn't have to keep a team running to give answers to the questions that pop up in the internet about their clothes that the celebrities or influencers are wearing. They also would get to know which indirect advertisement was the most efficient and which style the users prefer.

B. Effects to the users/customers

The users trying to follow their bias's style wouldn't have any difficulty finding out the exact brand and clothes, style their bias is wearing. They would get the exact information as soon as they hit a picture into Shoppick.

Users not knowing what to wear or not knowing where to buy the clothes they want can easily have access to their wants while Shoppick gives them a recommendation according to the data gathered from the individual's SNS.

3. Glossary

3.1 Objectives

In this part we define the terms used in this document. It will be at the eye level of readers that doesn't have any background knowledge of this area. Hence most of the terms will be covered.

3.2 Definitions of terms

A. Terms for people

Terms	Definition
Company owners / Sellers	People who are owning a shop or involved in producing or selling clothing
Customers	People who are trying to or are buying clothing
User(s)	People who are using 'Shoppick'
Administrator(s)	People or an individual operating 'Shoppick'. They oversee management and maintenance of the server for the application.
Developer(s)	People or an individual who are involved in developing or evolving 'Shoppick'.

Table 1. Terms of people

B. Terms related to the service

Terms	Definition
Login	Process of users getting access to the service
Logout	Process of users leaving the service
Search	Looking up for the styles the user(s) want
Style Tag	Defining the users with specific style related words

Category	A section or a group where something or someone belongs
SNS (Social Network Service)	Online vehicle for creating relationships with other people who share an interest, background, or real relationship. Social networking service users create a profile with personal information, photos, etc. and form connections with other profiles
Recommendation	It is the act of suggesting a product to users

Table 2. Terms related to service

C. Technological Terms

Terms	Definition
Deep learning	Subset of machine learning in artificial intelligence (AI) that has networks capable of learning unsupervised from data that is unstructured or unlabeled.
Algorithms	A process or set of rules to be followed in calculations or other problem-solving operations, especially by a computer
Database	A structured set of data held in a computer, especially one that is accessible in various ways. It will be holding user style information
Back end	Relating to or denoting the part of a computer system or application that is not directly accessed by the user, typically responsible for storing and manipulating data.
Front end	Relating to or denoting the part of a computer system or application with which the user interacts directly
Android	Mobile operating system developed by Google. It is used by several smartphones and tablets
User Interface(UI)	Point of human-computer interaction and communication in a device.

URL	A URL is the fundamental network identification for any resource connected to the web

Table 3. Technological Terms

4. User Requirement Specification

4.1 Objectives

In this section we go through what the system offers the users. We will be focusing on functional and non-functional requirements. This section is written in the level for readers who doesn't have any expert knowledge on the field to be capable of understanding.

4.2 Functional Requirements

A. Sign up & Login

This is the process of users getting into 'Shoppick'. When a user first downloads and starts 'Shoppick' the user is asked to login with one of their SNS account that they use the most often. After they pressed login, they will be asked to agree with our terms of using the picture information uploaded in their personal accounts. The personal pictures gathered from the user's account will be used to run a deep learning algorithm. Through the algorithm 'Shoppick' will get up to three style tags and these tags along with the encrypted user ID and password will be stored in the database. All the personal pictures used will be deleted completely. The users will be automatically logged in whenever they have access to 'Shoppick'.

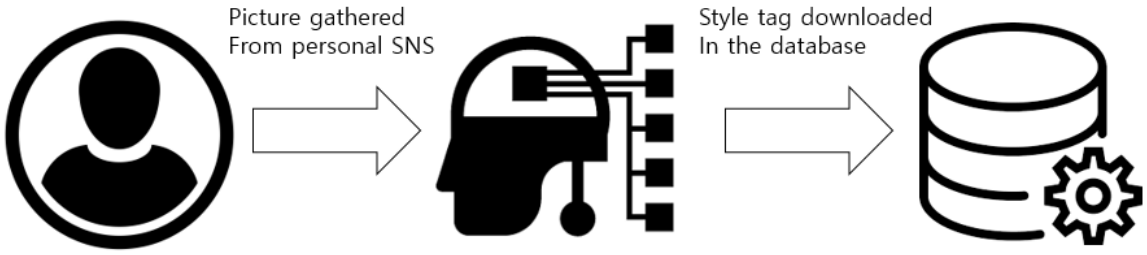


Figure 3. Login process

B. Shop recommendation

According to the user's personal tags that are saved in the database, when the user goes to the recommendation section individuals are given different results. When the user does not have any information on their SNS it is also noted in the app. The user's tags are noted on the recommendation page for the users to know what their style is defined as.

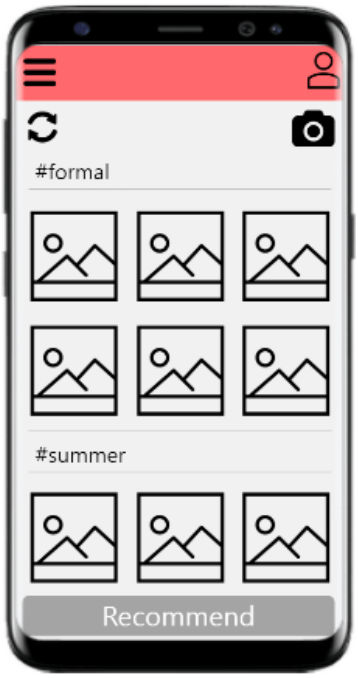


Figure 4. Recommendation screen

C. Real-time search

The users can use their cameras call a photo from their gallery to search for the same or identical clothes as the photo input to the application. The image the user sends to 'Shoppick' will go through the 'Shoppick' deep learning system and will be compared to the clothing data taken from the sellers. The best matches will be shown on screen with a URL attached leading to the seller's website for the specific product.

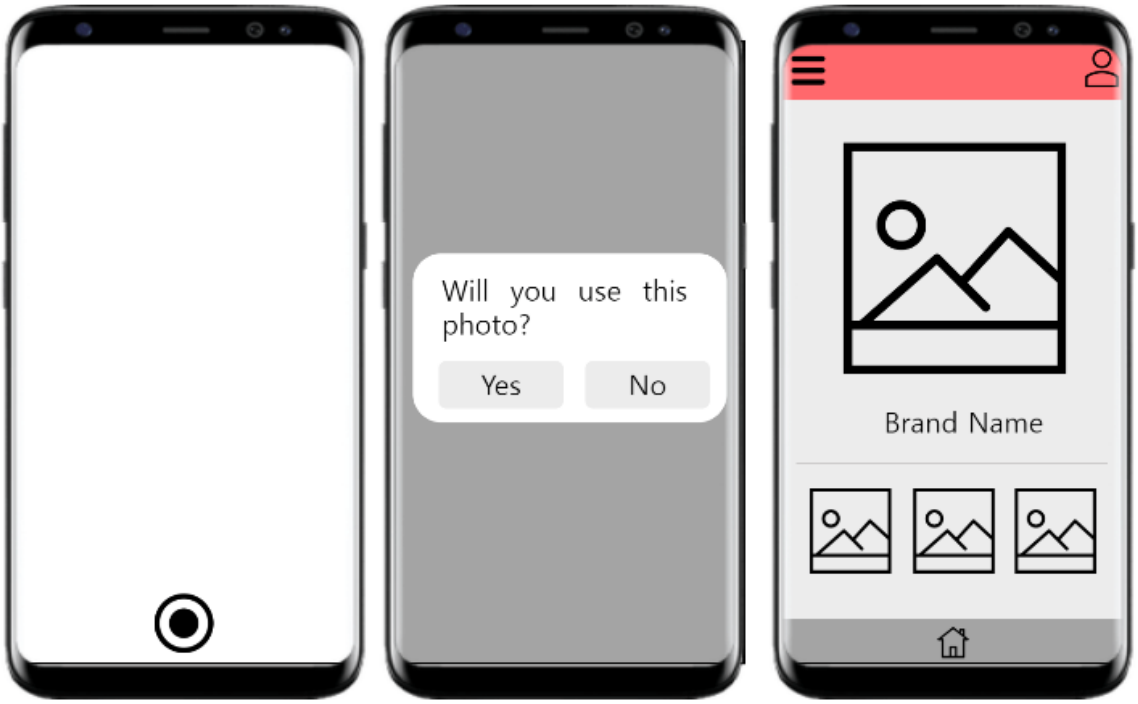


Figure 5. Real-time searching process screen

D. My page

The users can see the tags they have been assigned with the SNS data that was obtained during their sign in process. The users after a long time is recommended to refresh their tags by reloading the SNS info to 'Shoppick'. This process can be done anytime the user wants by clicking a refresh button on my page screen.

4.3 Non-functional Requirements

A. Product requirements

a. Performance requirement

When the user first signs in to the system the deep learning algorithm has to work without showing critical error rates. The system has to correctly pick out the clothing in the picture given. The clothes should be correctly defined and tagged

accordingly without much errors.

b. Security requirement

During the process of getting personal pictures from the user's SNS, the pictures shouldn't be obtainable by other intruders. 'Shoppick' is trying to block any security issues considering this by not keeping any pictures inside the database. However, it needs to be evolved in a way that no intruders can get in between the user and the system while the pictures are being sent.

c. Dependability requirement

The style tags shouldn't be lost due to database malfunction nor shouldn't be putting out wrong styles due to functional deep learning system.

d. Usability requirement

The UI (User interface) should be kept simple as there are just two functions. The front-end should be user friendly and easy to use and straight forward so that any users can use without much tutorials.

e. Efficiency requirement

The speed of the deep learning process doesn't matter in this case however it shouldn't be extremely slow (for example taking 1hr to process 1 picture). People should be able to use it without any hardships that might be in user's point of view be seen as a malfunction.

B. Organizational Requirement

a. Development Requirement

The development will be processed as parallel but also sequential. Back-end part deep-learning, web crawling and database will be developed in a parallel way. After the back-end is developed fully, the front-end will be developed with the data formats taken from the back-end. The whole system will be checked if there is no malfunction after the system is fully made.

b. Operational Requirement

The users have their SNS linked to the system so the application has to be careful of misuse of personal data by developers or and any outer intruders.

C. External Requirement

a. Profitability Requirement

'Shoppick' uses Deep fashion dataset for training and a deep learning algorithm that has been slightly altered from FashionNET. It doesn't have a noticed license yet but to use the application directly as a profitable model can be troublesome. The application will be run for free till contact has been made to the original developers and until then the profits will be made by putting on advertisements inside the application.

5. System Architecture

5.1 Objective

This part is for the readers with a technological background. The things are written in a more precise and professional way. It will be describing the parts of systems and how they cooperate with each other to form the whole system.

5.2 Front end Architecture

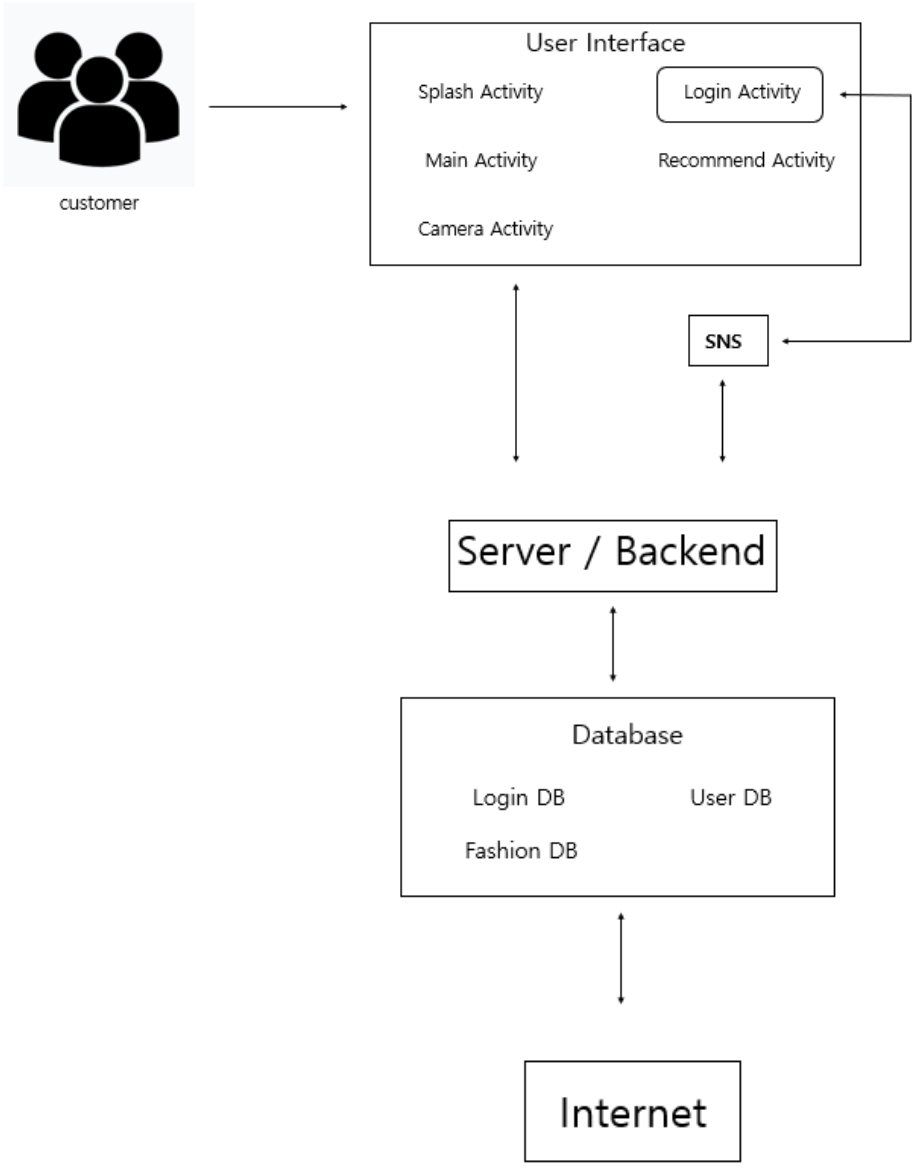


Figure 6. Front end Architecture

The frontend architecture is a structure that is requested and requested by the user. At the start, the splash activity is executed and the login activity comes out for a smooth screen transition. In the login activity, login determination is performed using the SNS account, and if successful, the main activity is transferred. In the main activity, whether customers use a camera activity or a photo from customer's SNS account, customers can make recommendations based on customer's intentions.

5.3 Back end Architecture

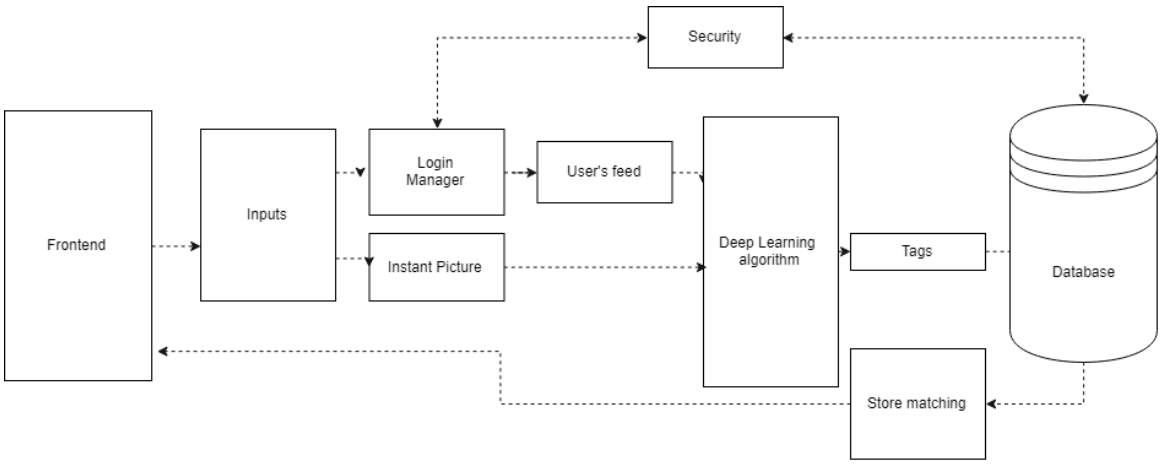


Figure 7. Back end Architecture

5.4 Recommendation system

Recommendation system draws out recommended items based on the style tags collected from users and input in the 'Shoppick' DB. Server when received login or reset call from the front end, it runs the deep learning algorithm and saves or updates the individual tags for users. From then the user is always provided with the recommendation that matches their individual tags.

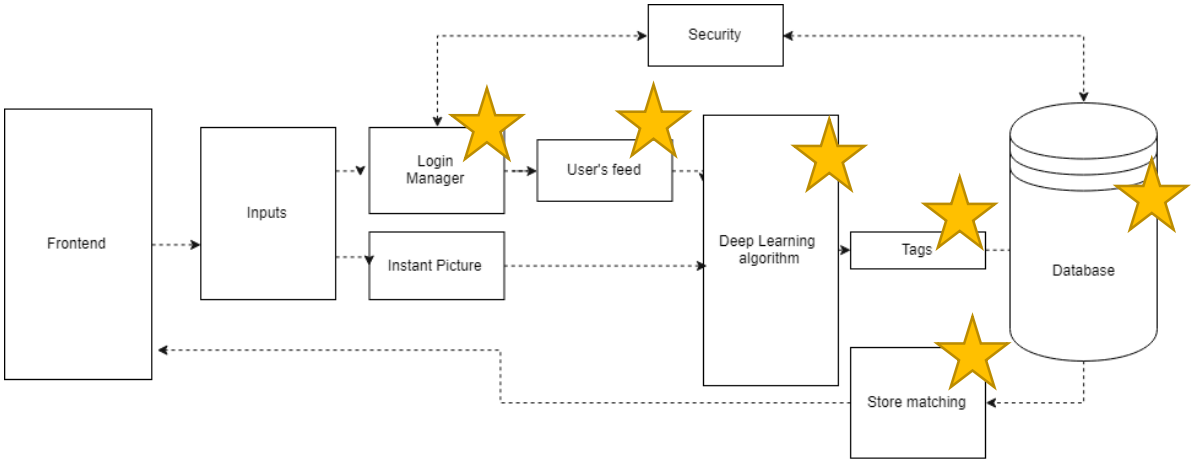


Figure 8. Recommendation system

5.5 Look-up system

Look up system works similarly to the recommendation system the only difference being is that instead of SNS data the recommendation system uses to put through the deep learning process, look-up system uses the given singular picture and tries to get the exact or at least similar clothing from the database.

6. System Requirements Specification

6.1 Objective

In this section the functional and non-functional requirements are written in an intensive depth. Also includes scenarios for better understanding.

6.2 Functional Requirements - Front end

A. Login

Name	**Login Activity**
Description	SNS계정과의 연동을 통한 로그인 액티비티
Input	SNS 로그인 버튼 클릭
Activity	SNS에 로그인 후 로그인 토큰 반환
Output	로그인 토큰의 유효성 여부 확인 후 Main Activity로 화면 전환, 실패 시 오류 출력 후 Login Activity에 머무름
Requirements	SNS계정 필요
Pre-condition	계정정보 無, 로그인 버튼
Post-condition	로그인 정보 유지, 로그아웃 버튼

Table 4. Functional Requirements: Login

B. Camera

Name	**Camera Function**
Description	패션 추천을 받기 위한 카메라 구현
Input	Camera icon click
Activity	실제 Camera를 통한 Image capture후 Backend Database로 전송 후 해당 정보들을 수신

Output	수신 받은 Data를 토대로 Recommend Activity로 전환 후 사진들을 List view형태로 Display
Requirements	Camera에 접근하기 위한 권한 획득
Pre-condition	-
Post-condition	-

Table 5. Functional Requirements: Camera

C. Recommendation

Name	**Recommend System**
Description	Fashion DB를 탐색한 후 알고리즘에 따른 결과에 대한 Fashion 추천받기.
Input	Recommend button click
Activity	SNS계정 Crawling을 통하여 사진들에 대한 정보를 Backend Database로 전송 후 해당 정보들을 수신
Output	수신 받은 Data를 토대로 사진들을 List view형태로 Display
Requirements	SNS계정 접근을 위한 권한 획득
Pre-condition	Database에 Fashion에 해당하는 정보들 존재
Post-condition	Customer와 관련된 정보들 노출

Table 6. Functional Requirements: Recommendation

D. Fashion site

Name	**Linking the site**
Description	추천받은 Fashion을 취급하는 브랜드의 홈페이지 접속.
Input	Fashion information에서 Site link 클릭
Activity	Link된 URL을 통하여 해당 사이트의 정보확인
Output	해당 사이트의 URL을 통한 화면전환
Requirements	해당 Fashion을 취급하는 브랜드 및 Homepage정보 존재
Pre-condition	-

Post-condition	브랜드 Homepage로의 전환

Table 7. Functional Requirements: Fashion site

6.3 Functional Requirements – Back end

A. Login Manager

Name	**Login function**
Description	Login to the user's SNS account
Inputs	username: username from the SNS account password: Password of the owner of the SNS account
Source (input)	User input
Outputs	User's SNS feed
Destination (output)	Database
Action	Takes both parameters and opens a new window in the browser to the SNS login page, introduces username and password and goes to the user's profile
Requirements	Username and password must be specified and correct
Pre-condition	No SNS-based recommendations for the user
Post-condition	Program will retrieve user's SNS pictures
Side effects	-

Table 8. Functional Requirements: Login Manager

B. User's feed

Name	GetPics
Description	Downloads the user's profile pictures and saves them to a temporal directory
Inputs	Profpage: User's SNS profile page
Source (input)	Login function

Outputs	User's pictures
Destination (output)	Deep Learning algorithm
Action	Analyzes the user's profile page and downloads the most recent posted pictures.
Requirements	Login must've been successful and user must have at least 3 pictures in its profile
Pre-condition	No pictures to analyze
Post-condition	The Deep learning algorithm can be used
Side effects	-

Table 9. Functional Requirements: User's feed

C. Instant Picture

Name	camera
Description	It sends the picture taken by the camera to the deep learning algorithm
Inputs	Picture: picture taken with the camera
Source (input)	Smartphone camera
Outputs	Picture
Destination (output)	Deep learning algorithm
Action	Receives the image taken with the smartphone's camera and sends it to the deep learning algorithm
Requirements	The user should have chosen to use a picture taken with the smartphone rather than the SNS profile.
Pre-condition	-
Post-condition	-
Side effects	Only the picture taken will be analyzed for matches

Table 10. Functional Requirements: Instant Picture

D. Deep learning algorithm

Name	Deep fashion
Description	Analyzes the received pictures and creates tags

	according to the style of the clothes found in them.
Inputs	Pictures: The pictures received from the user
Source (input)	SNS feed or smartphone camera
Outputs	Specific user style tags.
Destination (output)	Database
Action	Using deep learning, the received images are analyzed, the clothes found in them are classified according to its style and tags are created based on the styles that are used the most by the user, then the tags are saved in the database.
Requirements	There must be at least one image to analyze from either source.
Pre-condition	No user-specific tags.
Post-condition	Customized style tags.
Side effects	-

Table 11. Functional Requirements: Deep learning algorithm

E. Store matching

Name	Matching
Description	Chooses the stores that fits the best to the user's style tags
Inputs	Tags: The tags created for the current user
Source (input)	Deep learning algorithm
Outputs	Relevant stores
Destination (output)	Frontend
Action	The user's tags are taken from the database and are compared to the styles found in stores, then the ones that fit them the best are sent back to the user.
Requirements	-
Pre-condition	No stores available
Post-condition	Stores whose styles matches the best to the tags.

Side effects	-

Table 12. Functional Requirements: Store matching

F. Security

Name	EncryptDecrypt
Description	Takes the username and password from either the login function or the database and encrypts or decrypts them to store them or login, depending on the function
Inputs	User: The user's account Pw: User's password EncUser: User's encrypted username EncPW: User's encrypted password
Source (input)	Database or login function
Outputs	Encrypted or decrypted user and password
Destination (output)	Database or login function
Action	This function works both ways, it can take the username and the password from the login function, encrypt them and store them in the database, or take them from the DB, decrypt them and use them in the login function.
Requirements	The username and password must be correct if using it for encrypting, or the database should contain previous usernames and password if used for decrypting
Pre-condition	Encrypted/decrypted usernames and passwords
Post-condition	Encrypted/decrypted usernames and passwords
Side effects	-

Table 13. Functional Requirements: Security

6.4 Non-functional Requirements

A. Product Requirements

1. Usability

As with every application, its interface should be intuitive while keeping the simplicity in mind, the user should be able to interact in an easy manner and without needing any kind of advanced knowledge.

2. Performance

As we will handle a large amount of data, the algorithms should be optimized to get the tag styles without consuming lots of processing power, the mobile app should work smoothly in most devices, because all of the work will be handled in the backend and thus the frontend will just display the results.

3. Security

As we will be using personal information such as usernames, passwords and user's personal pictures we need a way to keep this safe so that no one can have access to them, the pictures need to be deleted once the algorithm determines the user's tags and as we want the user to be able to re-log in without needing to introduce its data again, we need to store the username and password into the database, to make this process more secure, the system should encrypt them before storing and decrypt them every time we want to use them.

4. Dependability

As our system will work with a rather large amount of data, the main goal is to keep everything working smoothly while trying to avoid errors. If an error arises it should be handled with the objective of not putting the user's information at risk, if there is any error with the login credentials, the user will be notified so that its account is not at risk of being blocked due to multiple login attempts, and all of the stages related with the retrieving and analyzing of the pictures should have many barriers to avoid errors, and finally if all of these steps still fail, the system

will immediately remove all of the user data that was created so far and deleted, to then ask the user for another login or communicating the bug with the developer.

B. Organizational Requirements.

1. Operational

All of the backend part is expected to be handled in only one server, this will be the one implementing all of the algorithms for the deep learning and style matching with the stores.

2. Development requirements

A parallel development will be used, because the frontend and backend will work almost independently from each other, the only interaction they will have is to send the user's data and return the recommended stores.

C. External requirements

1. Security Requirements

The user should only enter its username and password, it is probably that the SNS will communicate the user of the login from a new device, and could even ask for another authentication step to protect the account such a code sent to the registered e-mail, therefore the user should complete these steps to successfully login.

6.5 Scenario Examples

A. Recommendation scenario

a. Initial Assumption

The user has a SNS account where he/she has uploaded a lot of pictures of him/herself. The user logs in with that specific account.

b. Normal flow of events

The user when logging in agrees with the terms of usage and the user's preferred style gets saved in the database along with the user's ID.

c. What can go wrong

A specific user might not agree to the terms of usage and be skeptical and worried about their personal information being handed out to our deep learning algorithm.

d. System state on Completion

When a user disagrees to use their SNS data for using personal recommendation service, even then the user should have access to the other services the application provides.

B. Look-up scenario

a. Initial Assumption

A user takes a photo or a snapshot of a TV show that the user is watching.

b. Normal flow of events

The picture should be used inside the algorithm and the same or similar clothing should appear on the matching list.

c. What can go wrong

There might be more than one people inside the photo or snapshot that the user

has uploaded and the matching list might not be the one that the user wanted the information of.

d. System state on Completion

The user should be recommended to crop the picture down to a singular target or the system should try to print out all the clothing that matches the people on the screen.

7. System Models

7.1 Objectives

This section helps understand the relationships between the process in the entire system by using diagrams and charts.

7.2 Context models

A. Context Diagram

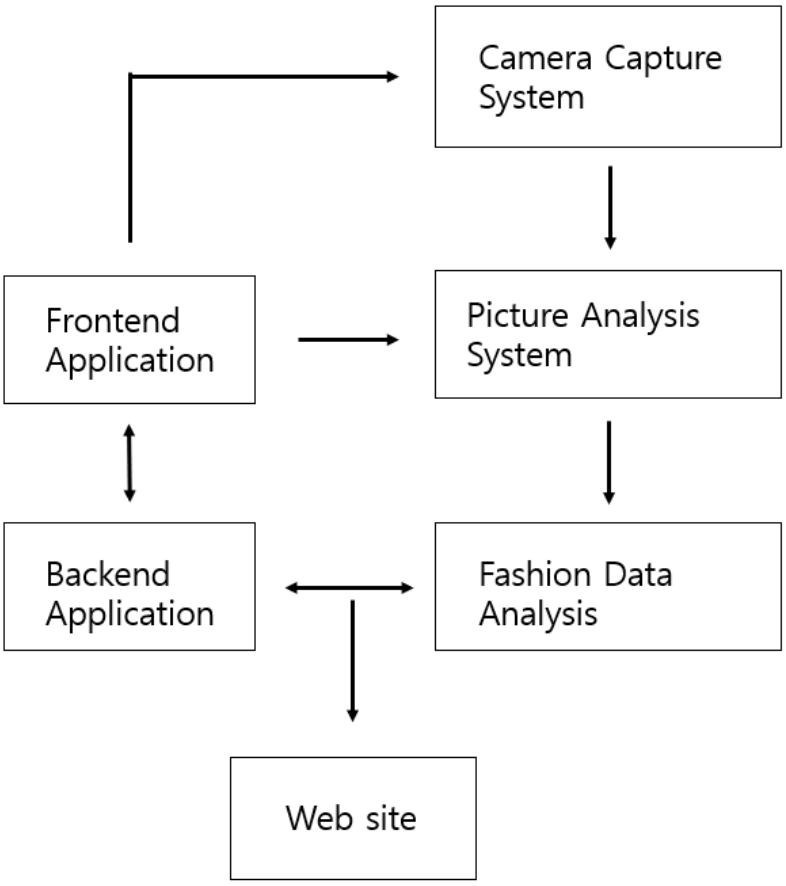


Figure 9. Context Diagram

B. Process Diagram

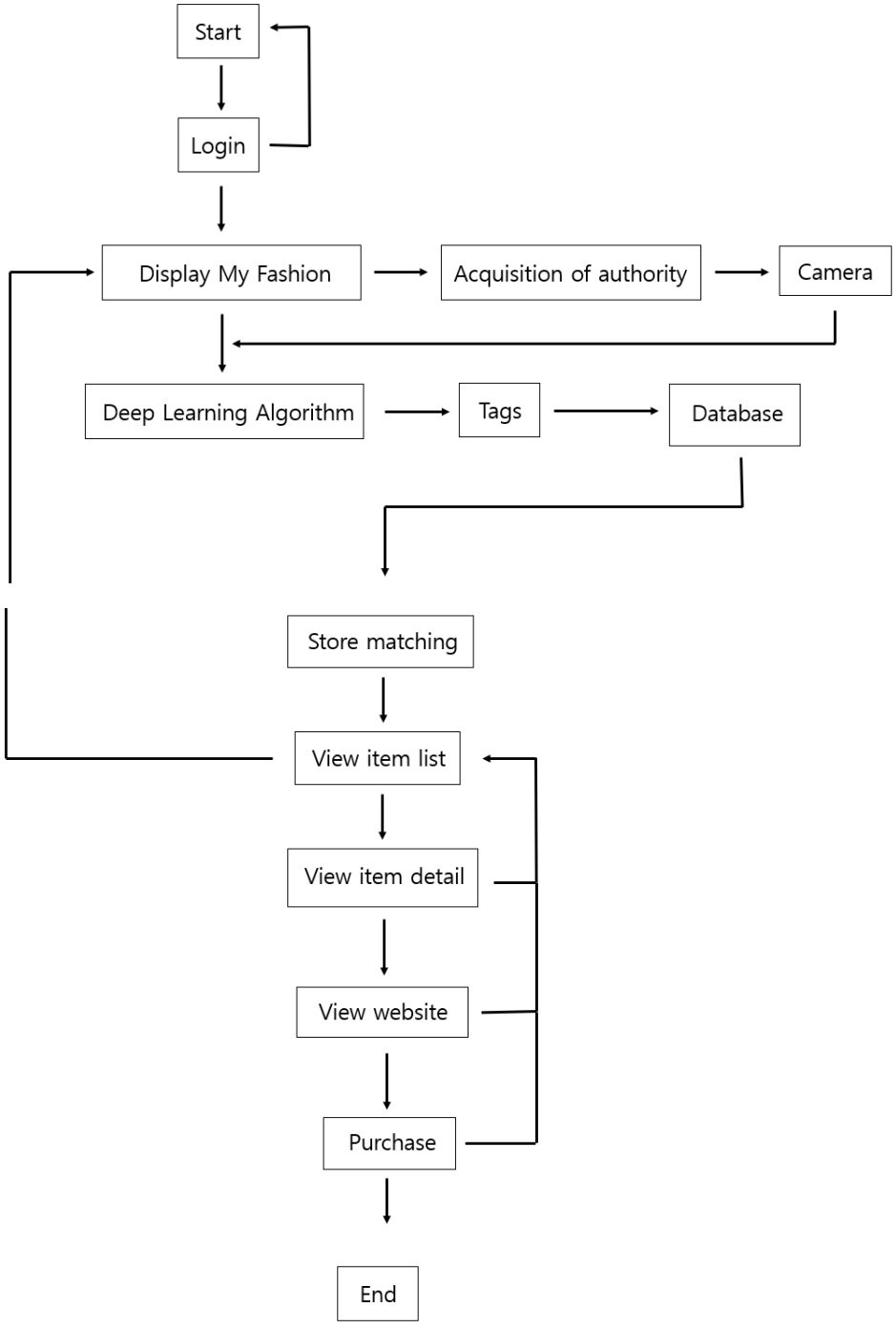


Figure 10. Process Diagram

7.3 Interaction models

A. Use case Diagram

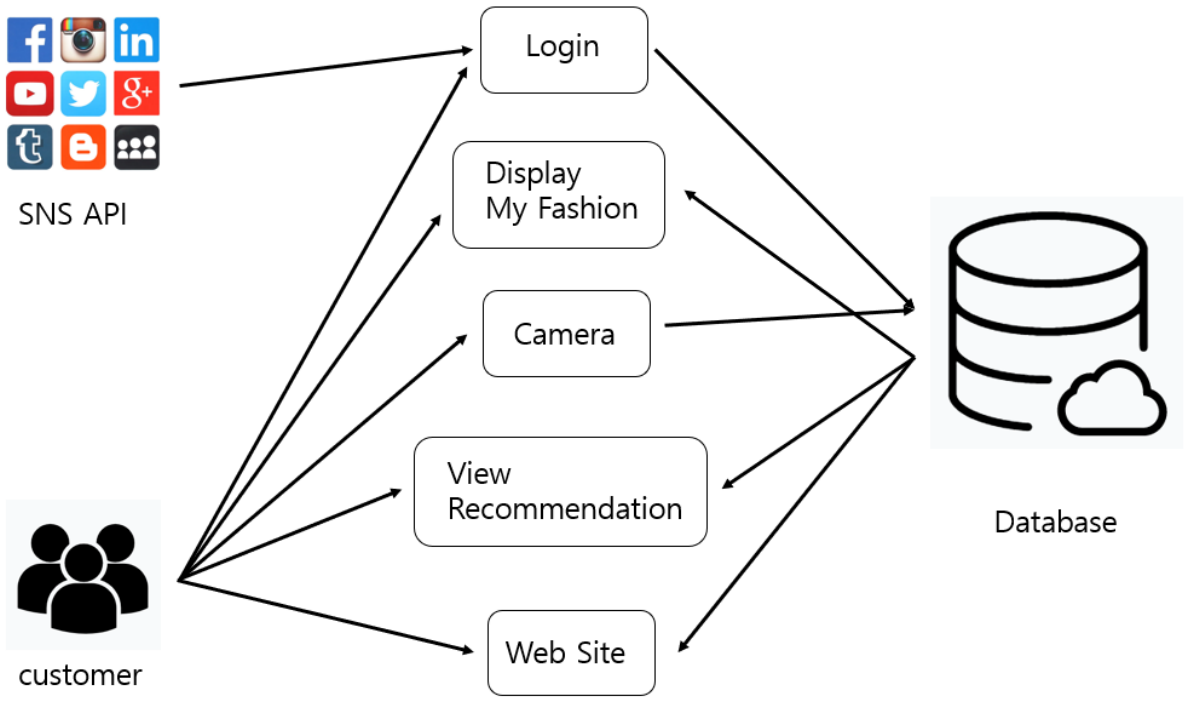


Figure 11. Use case Diagram

B. Tabular Description of Use case Diagram

a. Login

Use case	**Login**
Description	SNS계정을 통하여 로그인 할 수 있게 한다
Success Activity	로그인 성공 메시지를 출력하며 로그인 세션을 유지시킨다
Failure Activity	로그인 실패 메시지를 출력하며 로그인 화면으로 돌아간다

b. Display My Fashion

Use case	**Display My Fashion**
Description	Database에 저장된 내 스타일들과 관련 있는 Fashion종류

	들을 Display 한다
Success Activity	Fashion들을 List View형태로 나열한다.
Failure Activity	처음 사용하는 Customer들은 데이터가 없으므로 아무것도 출력하지 않는다

c. Camera

Use case	**Camera**
Description	실제 카메라로 사진을 찍어 관련된 Fashion들을 추천받을 수 있게 한다
Success Activity	권한 획득 메시지를 출력하며 동의하였을 경우 찍은 사진들을 서버로 보내 Database를 탐색한다
Failure Activity	권한 획득 메시지를 출력하며 거부하였을 경우 Toast 메시지로 거부하였다고 알린 후 원래 화면으로 돌아온다

d. View Recommendation

Use case	**View Recommendation**
Description	자신의 Fashion과 관련 있는 Fashion들을 나타낸다
Success Activity	사진과 정보들을 출력하며 클릭을 통해 자세한 정보 및 Fashion과 관련된 사이트의 정보를 출력한다
Failure Activity	관련 Data가 없다는 메시지를 출력한 후 Main Activity화면으로 돌아온다

e. Web Site

Use case	**Web Site**
Description	Fashion과 관련된 사이트의 정보를 출력한다
Success Activity	클릭 시 해당 사이트로 이동하게 된다
Failure Activity	사이트가 존재하지 않다는 오류메시지를 출력한다

7.4 Behavioral models

A. Deep Learning Algorithm

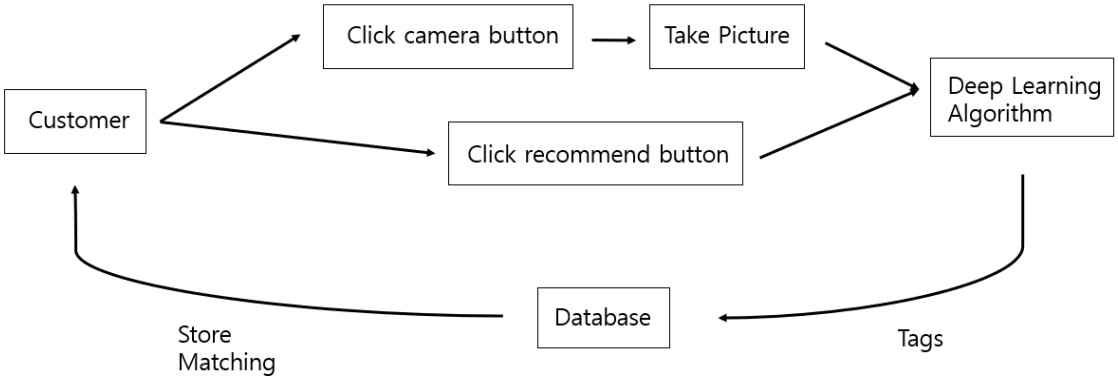


Figure 12. Deep learning Algorithm

8. System Evolution

8.1 Objectives

This section gives the draft idea of what we might be doing in the future and the effects 'Shoppick' can have while the technology evolves. By using this section, the system developer can develop the system while foreseeing future implementations and evolutions. The developers will be able to adapt to changes quicker by reading this section.

8.2 Applying GPS system

Picture 2. GPS system

As much shops there are on-line there are a lot of retailer shops offline. There are times when people lose their way in a huge department store or spend loads of time in one.

When the companies (retail shops) give their off-line GPS data to developers, they can store them by giving the GPS tags one to one matching coordinates to the map. The users will get a button that will show them the shops near them that

matches their personal tags.

8.3 Progress of computer calculation speed

As the computer calculation speed evolves, it is highly likely for the applications to send the deep learning data real time. The need of a safety tool such as a progress bar is still needed while the users are waiting for the deep learning process to finish. However, progress bars have to be developed in a way that uploading the process bar on the screen wouldn't be the cause of lags.

8.4 Advances of cameras attached on mobile phones

The cameras are getting better nowadays and there are smart phones that even have more than three cameras attached to it. Soon it wouldn't be hard to see phones that can converge a picture to 3D image. This would be helpful for 'Shoppick' since it can get more information of the clothing. The developers have to make sure that even if the extension of the image files differ it shouldn't have any trouble being run through the application.

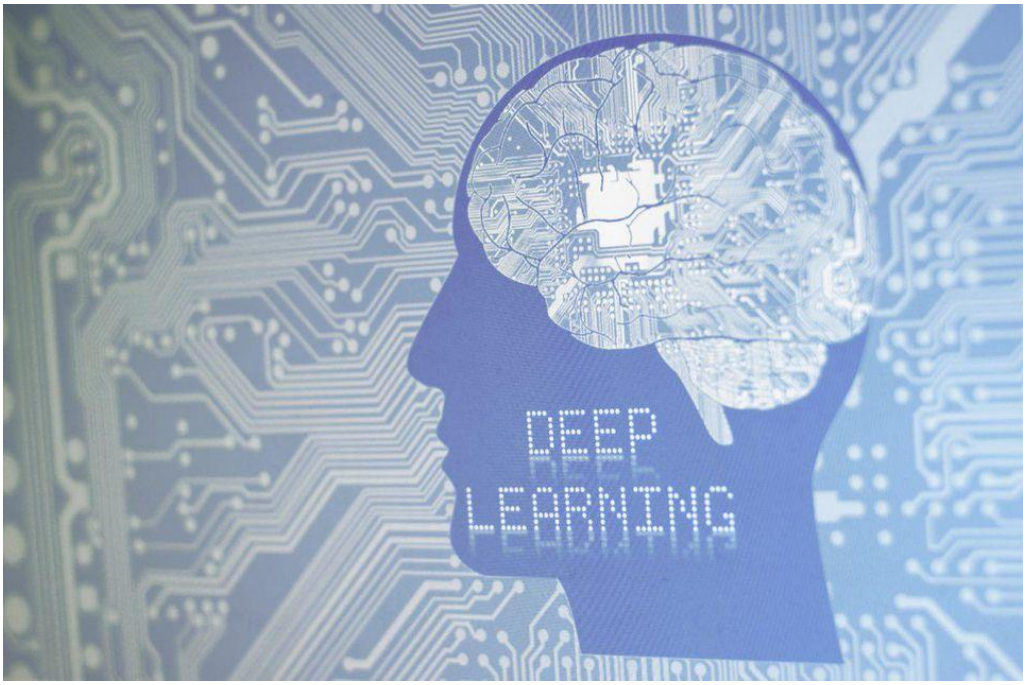
9. Appendix

9.1 Deep learning

Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on artificial neural networks. Learning can be supervised, semi-supervised or unsupervised.

Deep learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases superior to human experts

Artificial Neural Networks (ANNs) were inspired by information processing and distributed communication nodes in biological systems. ANNs have various differences from biological brains. Specifically, neural networks tend to be static and symbolic, while the biological brain of most living organisms is dynamic (plastic) and analog.



Picture 3. Deep learning

9.2 DeepFashion

Picture 4. DeepFashion

DeepFashion database is a large-scale clothes database, which has several appealing properties:

- First, DeepFashion contains over 800,000 diverse fashion images ranging from well-posed shop images to unconstrained consumer photos.
- Second, DeepFashion is annotated with rich information of clothing items. Each image in this dataset is labeled with 50 categories, 1,000 descriptive attributes, bounding box and clothing landmarks.
- Third, DeepFashion contains over 300,000 cross-pose/cross-domain image pairs.

Four benchmarks are developed using the DeepFashion database, including Attribute Prediction, Consumer-to-shop Clothes Retrieval, In-shop Clothes Retrieval, and Landmark Detection. The data and annotations of these

benchmarks can be also employed as the training and test sets for the following computer vision tasks, such as Clothes Detection, Clothes Recognition, and Image Retrieval

9.3 FashionNET

With the rapid growth of fashion-focused social networks and online shopping, intelligent fashion recommendation is now in great need. Tong He and Yang Hu design algorithms which automatically suggest users' outfits (e.g. a shirt, together with a skirt and a pair of high-heel shoes), that fit their personal fashion preferences. Recommending sets, each of which is composed of multiple interacted items, is relatively new to recommender systems, which usually recommend individual items to users. It explores the use of deep networks for this challenging task.

The system, dubbed FashionNet, consists of two components, a feature network for feature extraction and a matching network for compatibility computation. The former is achieved through a deep convolutional network. And for the latter, we adopt a multi-layer fully-connected network structure. It designs and compares three alternative architectures for FashionNet. To achieve personalized recommendation, we develop a two-stage training strategy, which uses the fine-tuning technique to transfer a general compatibility model to a model that embeds personal preference. Experiments on a large scale data set collected from a popular fashion-focused social network validate the effectiveness of the proposed networks.

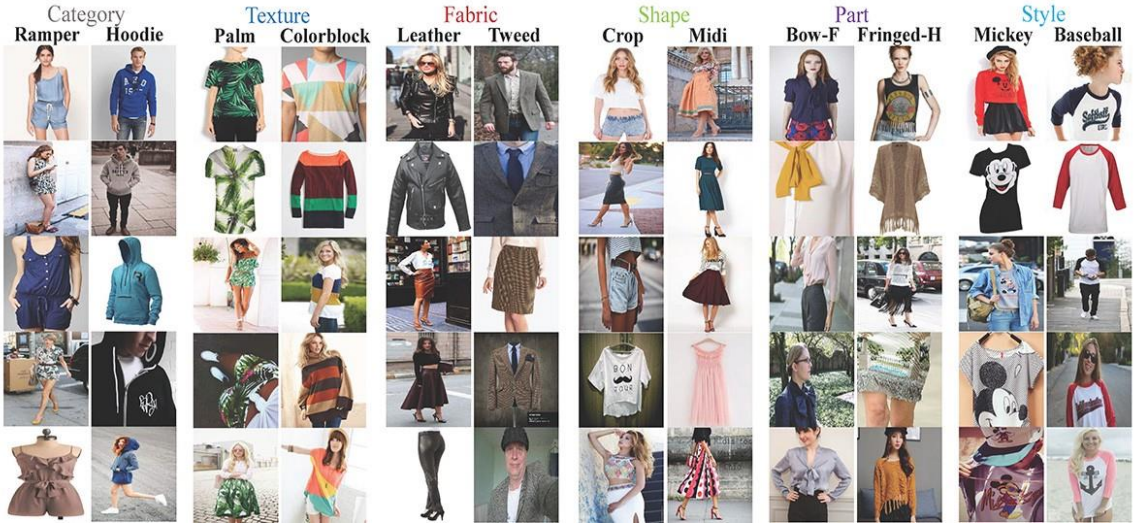


Figure 13. Clothes categorized with FashionNET

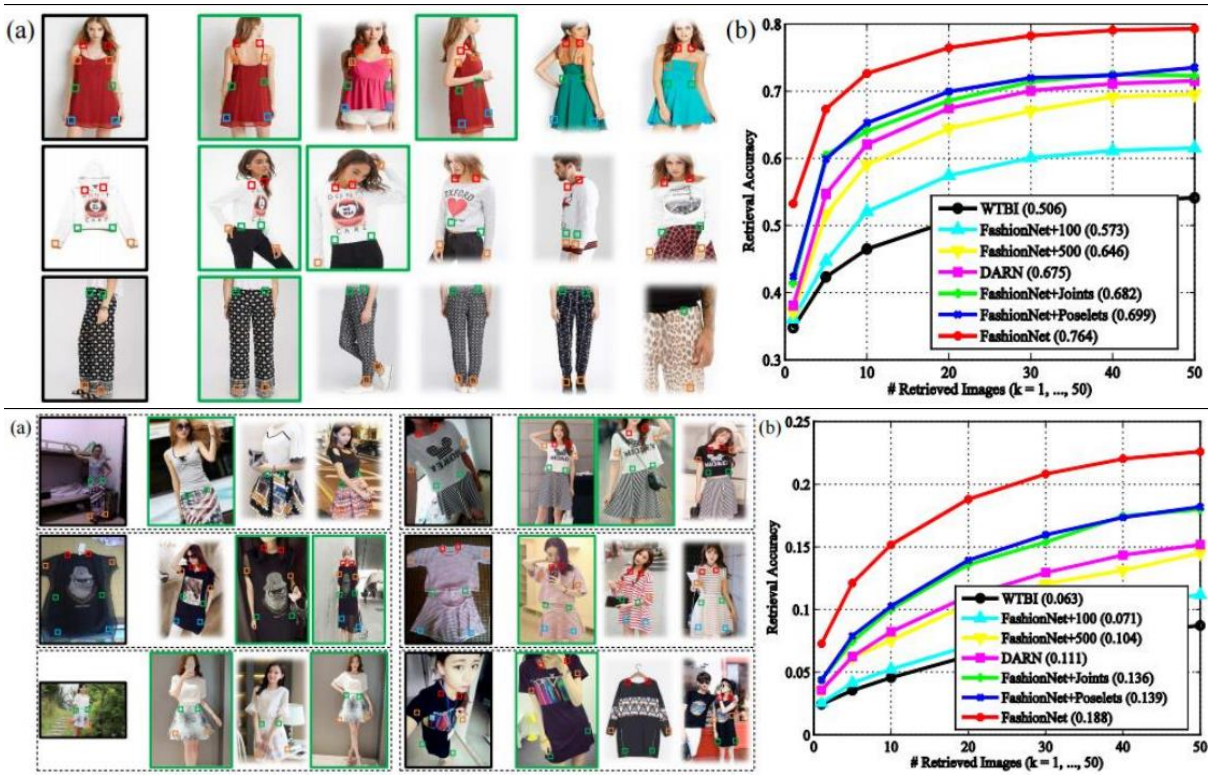
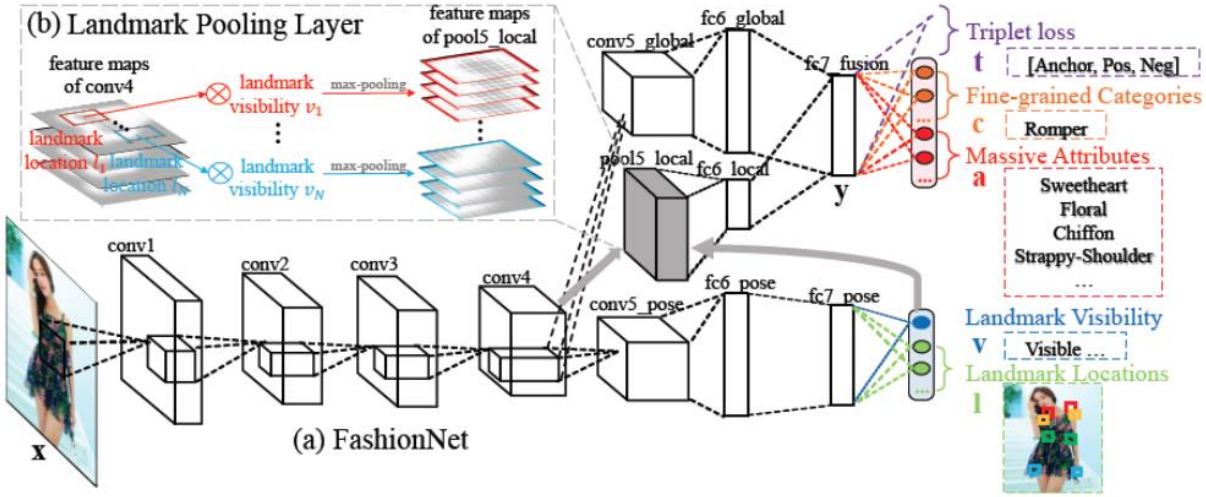


Figure 14. FashionNET correction rate

When using FashionNET an average over 85% correct rate is guaranteed. The picture of clothing doesn't necessarily have to be in-shop; model pictures it can be any types of picture of people wearing clothes as seen in the Figure above.



Picture 2. FashionNET architecture

10. Index

10.1 Figures

10.2 Tables

10.3 Pictures

11. Reference

a. Large-scale Fashion (DeepFashion) Database, Multimedia Laboratory, The Chinese University of Hong Kong: http://mmlab.ie.cuhk.edu.hk/projects/DeepFashion.html

b. FashionNet: Personalized Outfit Recommendation with Deep Neural Network, Tong He, Yang Hu

c. Wikipedia, https://en.wikipedia.org/wiki/Deep_learning