JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 21, No. 6, pp. 1025-1032, Jun. 2020

식품 불내성 환자를 위한 포장 식품의 OCR 기반 안전확인 시스템

자라 카미스[1] · 신옥근[2*]
[1]한국해양대학교 대학원 컴퓨터공학과 석사과정
[2]한국해양대학교 해사IT공학부 교수

OCR-Based Safety Check System of Packaged Food for Food Inconvenience Patients

Nurul Azzahra Putri Kamis[1] · Ok-Keun Shin[2*]

[1]Master's course, Department of Computer Engineering, Korea Maritime and Ocean University, Taejong-ro 727, Busan 606-791 Korea

[2]Professor, Division of Marine Information Technology, Korea Maritime and Ocean University, Taejong-ro 727, Busan 606-791 Korea

[요약]

음식 알러지나 면역질환으로 고통받는 환자가 늘고 있으며, 가공포장식품이 그 원인인 경우가 많다. 식품 불내증 환자가 해로운 가공식품의 섭취를 회피할 수 있는 스마트폰 기반 시스템을 제안한다. 식품포장지의 구성성분 사진을 찍어 서버에 보내면 서버는 OCR기술로 문자열을 추출한다. 인식의 정확도를 높이기 위하여 문자열은 토큰화와 사전비교의 두단계 후처리 과정을 거친다. 토큰화는 문자열을 토큰으로 분리하고 알파벳이 아닌 기호들은 제거한다. 토큰들은 영어사전과 비교되어 잘못된 철자의 교정이 이루어진 다음, 전문용어사전과 비교되어 음식의 구성성분인지를 검사한다. 서버는 사용자 DB와 구성성분–질병 DB를 이용하여 식품의 안전성 을 판별한다. 서버의 OCR, 토큰화, 사전검색의 기능 등은 오픈소스 소프트웨어를 이용하여 구현되었다.

[Abstract]

Patients suffering from food allergy or immune disease are increasing, and packaged foods are often the cause of outbreak of the symptoms. We propose a smart-phone based system that helps patients from consuming harmful packaged foods. The user takes the picture of the ingredients list printed on the envelop, and send it to the server. The server adopts OCR technology to retrieve the strings of the image. To increase the accuracy of the recognition, the output of the OCR is treated by two step post-OCR processes: tokenization, and lexicon look-up. The tokenization separates the strings into tokens and removes non alphabetic signs. Then the tokens are compared with english and proprietary dictionaries. The former corrects misspelled words. The latter selects only meaningful words in the context of ingredients. Using the user's information and ingredient-disease DB, the server makes decision on the safeness of the food. The server is built using open source softwares to process OCR, tokenization, and dictionary comparison.

색인어 : 식품 불내증, 사전, 광학문자인식, 후처리, 스마트폰
Key word : Food intolerance, Lexicon, OCR, Post-Processing, Smart phone

http://dx.doi.org/10.9728/dcs.2020.21.6.1025

Received 11 May 2020; **Revised** 15 June 2020
Accepted 25 June 2020

***Corresponding Author; Ok-Keun Shin**

Tel: +82-51-410-4572
E-mail: okshin@kmou.ac.kr

I. Introduction

Recently, patients suffering from food allergy, food intolerance and auto-immune disease are increasing, especially in young generations[1]. As slightest amount of the allergen contained in the food might trigger symptoms, the patients require precautions in their daily food intake. The increase of these abnormalities seem to be highly related with the increased consumption of packaged foods[2].

There has been studies to help these patients by warning the harmfulness of specific foods. A mobile application 'Eatable'[3] provides the location information of the stores selling foods that are safe from allergens chosen by the user. This application needs product bar-code and geo-location data. Another mobile application 'Food Allergy Scanner'[4] is an application which gives allergy information of a product based on the user provided allergens and scanned QR-code of the product. Both of theses applications use on-line information of the products.

In Eatable, they makes use of 'Label API'[3] which is a database of grocery products found only in the United States, and provides product information such as ingredient list, name, brand. Except Label API, it is difficult to find on-line food information, since there are, to the best of our knowledge, no other food product database, and few product manufacturers provide on-line information.

In this study, we propose a system to check whether the packaged food in hand contains harmful ingredients for the user by simply taking the picture of the list of ingredients printed on the envelope of the product. The main merit of this approach is that it's independent of the database of the packaged foods, and hence no need to gather food informations and maintain the database. However, the proposed method incorporates the optical character recognition (OCR) technology, and OCR of the list of ingredients is heavily error prone. In this paper, we try to minimize the recognition error of OCR process by appending a series of post-processing steps which makes use of two dictionaries, an ordinary on-line dictionary, and a domain specific, proprietary dictionary.

To use adequate terms, it is necessary to distinguish between food intolerance and food allergy. The former is a type of symptoms to food that might cause a mild physical reaction to the patient. It does not involve the response of immune system of the patient. The latter causes a serious reaction to the patient's body, affects numerous organs, and in some cases, a severe food allergy reaction could be life threatening[5]. In this study, we use the term food inconvenience to incorporate these two food reactions as well. In the following sections, we describe the OCR related topics in section 2, the Design of the system in section 3, the implementation of the system in section 4. The last section concludes the paper.

II. System Design

The overall system has a server-client model, where the client is a mobile application which sends the image of the ingredients list to the server to ask whether the product in hand is safe to consume. The server is supposed to keep each users personal information such as id, name, and in particular, their disease conditions, and processes the received image of ingredient list to decide whether the product is safe for the specific user.

In Fig. 1, the overall flow chart of the server is shown. We explain each steps in subsequent sections.

2-1 Optical Character Recognition

Once the image of the ingredient list is sent to the server, OCR technology is used to extract ingredients information in editable text format from the image. OCR adopts pre-processing steps before the main process, such as image rescaling, skew correction, binarization, noise removal, and segmentation (character isolation). Then the feature extraction and classification of each segmented character follows which play the major roles of the OCR. The extracted features must be independent of the scalable font characteristics such as type, size, style, tilt, rotation and should be able to describe the complex, distorted, broken characters effectively[6].

In this study, we adopt Tesseract OCR engine to extract the texts from the image. Tesseract is an offline OCR engine, free and open for software development[7]. The Tesseract OCR engine is combined with Leptonica[8] Image Processing Library that can recognize varieties of image format and convert them to text in over 100 languages.

2-2 Post Processing of OCR

The output of the OCR is a string of characters which contains the candidates of the ingredients as well as unnecessary symbols, numbers, and noises. Hence the results of the OCR itself is not suitable for ingredient identification. We introduce two post processing processes which make use of natural language processing technology for efficient ingredient extraction. The first is the tokenization step which splits the results of the OCR into words. Then the output of the

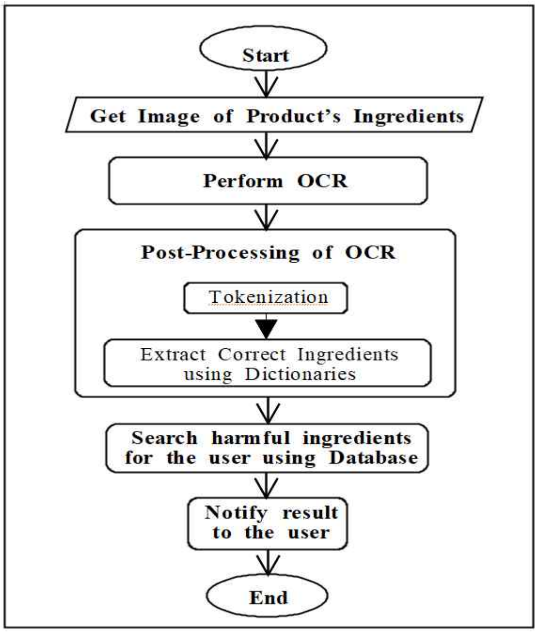

Start
Get Image of Product's Ingredients
Perform OCR
Post-Processing of OCR
Tokenization
Extract Correct Ingredients using Dictionaries
Search harmful ingredients for the user using Database
Notify result to the user
End

그림 **1.** 서버의 흐름도
Fig. 1. Flowchart of the Server

tokenization step is constrained by a set of lexicons, an ordinary dictionary and a proprietary dictionary. Each post-processing steps are explained in sequence.

1) Tokenization

In general, the tokenization is the act of splitting a sequence of strings into pieces called tokens, which can be words, phrases or symbols[9]. Tokens are separated by white spaces, punctuation marks or line breaks. On some cases, white spaces or punctuation marks are not needed, these marks may be thrown away during the tokenization. In this study, we consider the following four tokenization steps.

(1) Sentence segmentation: This step converts the strings into sentences, and are based on punctuations marks such as '.', '?' and '!' as they mark the boundaries of the sentences.
(2) Words segmentation: This step segments the sentence into words. Word is separated via space character.
(3) Character removal and convert to lowercase: In this study, we do not need any other information than the name of the ingredients. Hence, the presence of non-alphabetic characters is an indication of word boundaries. Numbers and specifiers like '%', '$', hyphens, comma and period signs are removed together. Then, to make the recognition easier, any uppercase alphabets are converted into lowercase alphabet.
(4) Multi-word tokenization: The ingredients name might be composed of more than one word, for instance, 'green tea', 'sour cream', or 'maple syrup'. It is necessary to takes a sequence of tokenized words and merge it to a multi-word token[10].

In this study, we do not take into account lemmatization and stemming processes to preserve the original tokens. In our experience, those processes deform the original tokens too much and makes the tokens meaningless as ingredient.

2) Extract Correct Ingredients Using Dictionaries

The second is the constraint of the retrieved ingredients by lexicons. Listed below are the dictionaries used in this system.

(1) Ordinary dictionary: The Ordinary dictionary in this study is an english dictionary which includes more than 370,000 words. The English dictionary database is obtained from a website trusted by many software developers[11].
(2) Proprietary dictionary: The proprietary dictionary is used as a guide to select the real ingredients out of the candidates of ingredients selected from the Ordinary dictionary. We prepared a proprietary dictionary which consists of about 5,000 entries for this study by collecting ingredients list from various sources. Most food ingredients, as well as additives and colors are obtained from the US food and drug administration[12] and an open source repository github.com[13]. As these lists contains a large number of redundancy, it was necessary to exclude the duplicates.

The main purpose of the Ordinary dictionary is to check wether the retrieved tokens are meaningful words and in correct spelling. Then, the candidates are compared with the proprietary dictionary to make sure that only the correct ingredients are selected, The flowchart in Fig. 2, shows the procedure how the correct ingredients are extracted using dictionaries. Using a spell checker tool, each word tokens are first searched for in the Ordinary dictionary. If the word is found in this dictionary, the word is considered as correctly recognised word, and continues to the next step. If the token is not found in the Ordinary dictionary, the spell checker proposes a word nearest to the token. We consider this operation as a correction of the token. The corrected word, as well as the word that passed Ordinary dictionary test will be tested again by comparing against the proprietary dictionary. Here also, the words are compared via a spell checker tool. If the word is found in the proprietary dictionary, it is an ingredient. If the spell checker fails to find the word in this dictionary, the system notifies the end-user as unrecognized. The system however, append this token to the update list so that the administrator of the system can occasionally review and update the proprietary dictionary.

2-3 Search for Harmful Ingredients to the User

A database called 'diseaseDB' is used to decide wether the ingredient list obtained from the envelop of the product contains any ingredient harmful to the user. The database includes two tables: 'user' table and 'disease_ingredients' table. The user table contains columns that are filled with the food intolerance disease of the user. The 'disease_ingredients' table contains a list of diseases, and the ingredients which causes the disease.

Once the list of the ingredients of a product is obtained in text format, the server retrieves the user's disease from the user table and obtains the ingredients related to the disease from the disease_ingredients table.

Then, the extracted ingredients from the envelop are compared with the ingredients from the disease_ingredients table. If there are any match, it means that the product is not safe for the user.

2-4 Notify the Result to the User

The server's decision on the safeness of the food is sent to the user's terminal. The user's mobile application receives either 'UNSAFE' or 'SAFE' notification. When the product is 'UNSAFE', the list of harmful ingredients are provided together. If there are any unrecognized ingredients, the system makes decision excluding the unrecognized ingredient. The list of the unrecognized ingredient is also shown to the user along with 'UNSAFE' or 'SAFE' notification.

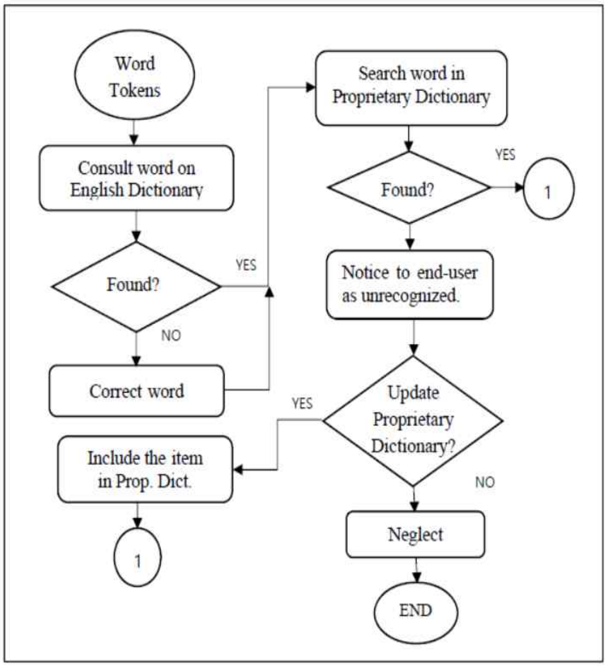

Word Tokens
Consult word on English Dictionary
Found?
YES
NO
Correct word
Include the item in Prop. Dict.
1
Search word in Proprietary Dictionary
Found?
YES
1
Notice to end-user as unrecognized.
Update Proprietary Dictionary?
YES
NO
Neglect
END

그림 **2.** 사전을 이용한 구성성분 추출
Fig. 2. Extraction of Ingredients Using Dictionaries

III. Implementation

As explained in the previous chapter, the system consists of server and client, where the front-end client system is an android mobile application that provides UI to the user. The back-end sever receives image of ingredients list, analyzes the ingredients of the food and returns the safeness result to the user. To implement the server system, appropriate third-party open source softwares are chosen and assembled in a python environment.

For the text recognition, the Tesseract's OCR library is chosen. This program recognizes the input text image and converts them into editable text. The output text is fed into the first post-processing processes, which are tokenization, segmentation and multi-tokenization. In this step, the open source software Natural Language Tool Kit (NLTK)[14] is adopted to perform these functions and outputs so called word tokens.

The retrieved word tokens are input to the second post-processing processes, where each tokens are compared with two dictionaries in sequence by means of PyEnchant[15], a python spell checker tool. Firstly, the tokens are compared with the ordinary dictionary, "Dataset English-Words List"[16]. The spell checker can suggest most similar words for any unrecognized word. We make use of this option to correct any misspelt word. Then, the proprietary dictionary is used to select only ingredients components which are registered in this dictionary. The words that are not recognized by the proprietary dictionary will be accumulated in a list called 'unrecognized ingredients'. In this study, the proprietary dictionary is assumed incomplete and the administrator can append new ingredients by examining 'unrecognized ingredients' list. MySQL is used to implement the diseaseDB which contains the user's personal data as well as the disease-ingredient data. Fig 3 shows the sequence of the software tools used to implement the system.

In the following sub-sections, we show the inputs and outputs of each processes of the system, where the sample image chosen as the system's input is an image taken from the envelop of Hershey's Milk Chocolate.

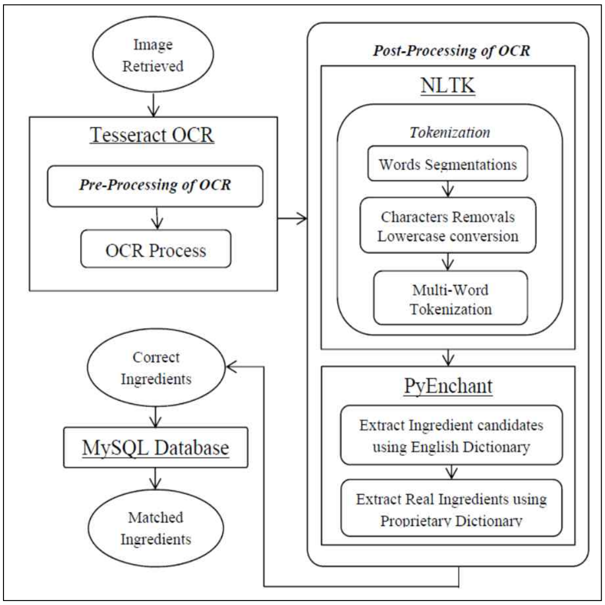


그림 3. 서버 시스템의 구현
Fig. 3. Implementation of the Server System

3-1 Pre-Processing and OCR

As mentioned above, Tesseract OCR library is used to perform OCR. In Fig.4, the original image captured from the envelope is shown.

Before character recognition, it is necessary to go through pre-processing steps such as noise removal, image rescaling, skew correction and binarization. These functions, based on Numpy and OpenCV utilities, are also included in the Tesseract OCR library. The result of the pre-processing is shown in Fig. 5 below. The pre-processed image is then sent for OCR process to retrieve the editable texts. Fig. 6 shows the OCR output. As can be seen from Fig.6, the output of OCR contains many unnecessary components such as punctuation symbols, numbers, special characters, and etc. These will be removed in the post processing step that follows.

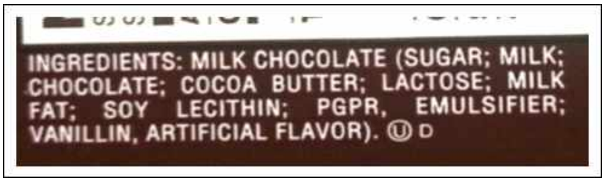

INGREDIENTS: MILK CHOCOLATE (SUGAR; MILK;
CHOCOLATE; COCOA BUTTER; LACTOSE; MILK
FAT; SOY LECITHIN; PGPR, EMULSIFIER;
VANILLIN, ARTIFICIAL FLAVOR). Ⓤ D

그림 **4.** 상품의 구성성분 이미지
Fig. 4. Original Image of the Ingredient of the Product

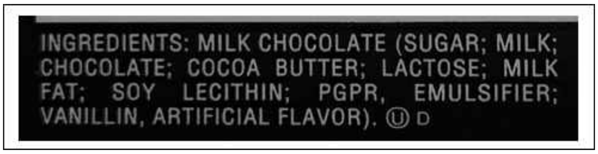

INGREDIENTS: MILK CHOCOLATE (SUGAR; MILK;
CHOCOLATE; COCOA BUTTER; LACTOSE; MILK
FAT; SOY LECITHIN; PGPR, EMULSIFIER;
VANILLIN, ARTIFICIAL FLAVOR). Ⓤ D

그림 **5.** 전처리 과정을 거친 이미지
Fig. 5. Image after Pre-Processing

3-2 Post-Processing of OCR

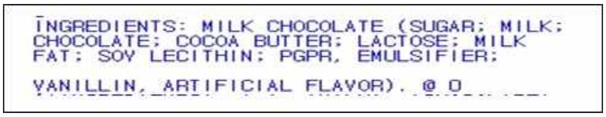
The post-OCR process is divided into two steps. Each steps are explained in the order.

1) Tokenization Process

The tokenization of the OCR output is achieved by one of the NLTK modules, nltk.tokenize.API module. This tokenizer segments the input string into words. In Fig. 7 below, we show the result of word unit segmentation step. Then unwanted characters, numbers and symbols are removed and the uppercase alphabets are converted into lowercase. Shown in the Fig. 8 below is the result of these steps. It happens that the name of an ingredient is composed of more than two words. The final step of tokenization is to reunite words that was originally a single ingredient, and separated into multiple words during the segmentation process. The result of multi word tokenization is shown in the Fig. 9 below. Here, the retokenized word is separated by '_' to indicate that there are white spaces in a token.

2) Get Correct Ingredients using Dictionaries

The outputs of the tokenizer are fed to two spell checkers to perform the function called constraint by lexicon. PyEnchant is


```
INGREDIENTS: MILK CHOCOLATE (SUGAR; MILK;
CHOCOLATE; COCOA BUTTER; LACTOSE; MILK
FAT; SOY LECITHIN; PGPR, EMULSIFIER;

VANILLIN, ARTIFICIAL FLAVOR). @ O
```

그림 **6.** OCR 결과물
Fig. 6. Result of OCR

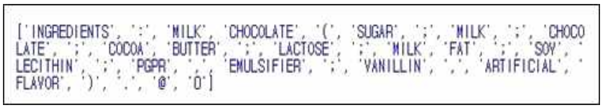

```
['INGREDIENTS', ':', 'MILK', 'CHOCOLATE', '(', 'SUGAR', ';', 'MILK', ';', 'CHOCO
LATE', ';', 'COCOA', 'BUTTER', ';', 'LACTOSE', ';', 'MILK', 'FAT', ';', 'SOY', '
LECITHIN', ';', 'PGPR', ',', 'EMULSIFIER', ';', 'VANILLIN', ',', 'ARTIFICIAL', '
FLAVOR', ')', '.', '@', 'O']
```

그림 7. 단어 단위의 세그멘테이션 결과물
Fig. 7. Result of Word Unit Segmentation

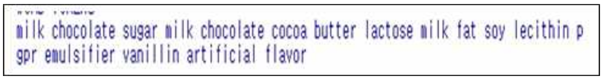
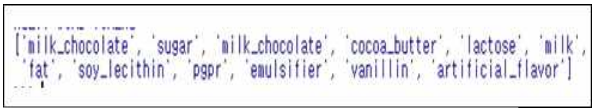

```
milk chocolate sugar milk chocolate cocoa butter lactose milk fat soy lecithin p
gpr emulsifier vanillin artificial flavor
```

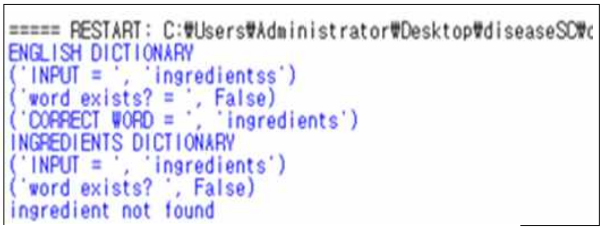
그림 **8.** 기호 제거 및 소문자 변환 결과
Fig. 8. Result of symbol removal and conversion



```
['milk_chocolate', 'sugar', 'milk_chocolate', 'cocoa_butter', 'lactose', 'milk',
 'fat', 'soy_lecithin', 'pgpr', 'emulsifier', 'vanillin', 'artificial_flavor']
```

그림 9. 멀티-워드 토큰화 결과

Fig. 9. Result of multi-word tokenization



```
===== RESTART: C:₩Users₩Administrator₩Desktop₩diseaseSC₩c
ENGLISH DICTIONARY
('INPUT = ', 'ingredientss')
('word exists? = ', False)
('CORRECT WORD = ', 'ingredients')
INGREDIENTS DICTIONARY
('INPUT = ', 'ingredients')
('word exists? ', False)
ingredient not found
```

그림 10. 사전을 이용한 단어 교정 결과

Fig. 10. Correction of Words Using Dictionaries

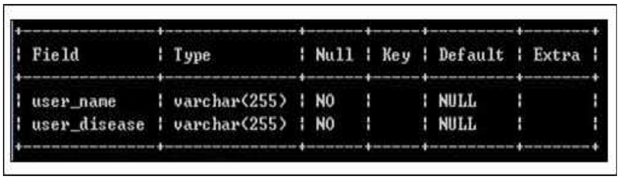
adopted to perform spell checking with an ordinary english dictionary as well as a domain specific proprietary ingredient dictionary. The main purpose of the ordinary dictionary is to correct any misrecognized words in the OCR process. For instance, in the upper part of the example shown in Fig.10, the token 'ingredientss' is a misrecognized word and is corrected to 'ingredients'. The second spell checking process makes use of the proprietary dictionary which is actually a list of ingredients. In the example shown in lower part of the Fig. 10, the word 'ingredients' is not an actual name of an ingredient. The spell checker reports that the 'ingredients' is not found in the dictionary. It is neglected and not included in the final ingredients list.

3-3 Database of the System

The database of the system is named as 'diseaseDB' and is used to keep the information of the patients as well as the relation of the disease and ingredients. Fig. 11 shows the implementation of user table which consists of the 'user_name' and 'user_disease' column. The information is composed of the users' personal data and their disease types. Shown on Fig. 12 is the implementation of the 'disease_ingredients' table, which consists of 'disease_name' column for the names of the diseases and 'disease_ingredients' column for the ingredients of the diseases. For example, the ingredients alfalfa, garlic, or etc. in the first row of the table can provoke a seizure of Lupus. The table is shown in two rows.

3-4 User Interface on Android Studio

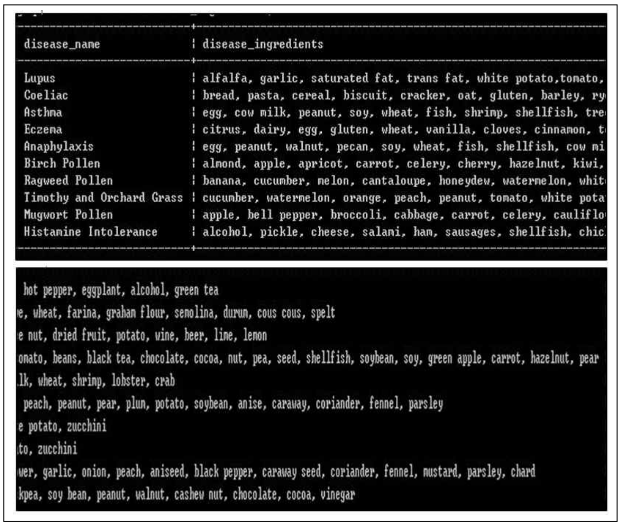
The user interfaces of the mobile application is implemented on Android studio. The user, after user registration, enters his disease information on the health profile page by selecting one of the disease's names available on the list. Then, the user can input the image of the product's ingredients by either taking a picture, or uploading an image from a folder of the mobile phone. When the image is uploaded, the server processes the image and the recognized strings are sent back to the mobile phone. The mobile phone displays it along with the uploaded image. An example of the input image and the recognized text output are shown on Fig. 13. The text recognized by the OCR is analyzed through the post-OCR process and database comparison, and the safeness information of the food is sent back to the user. In Fig. 14, the examples of safeness notice on the mobile phone is shown. When server informs the user with 'UNSAFE' notice, the ingredients that can cause trouble are shown together, telling that the product is not suitable for the user's diet.



```
+--------------+--------------+------+-----+---------+-------+
| Field        | Type         | Null | Key | Default | Extra |
+--------------+--------------+------+-----+---------+-------+
| user_name    | varchar(255) | NO   |     | NULL    |       |
| user_disease | varchar(255) | NO   |     | NULL    |       |
+--------------+--------------+------+-----+---------+-------+
```

그림 11. 사용자 테이블의 컬럼

Fig. 11 User Table Columns

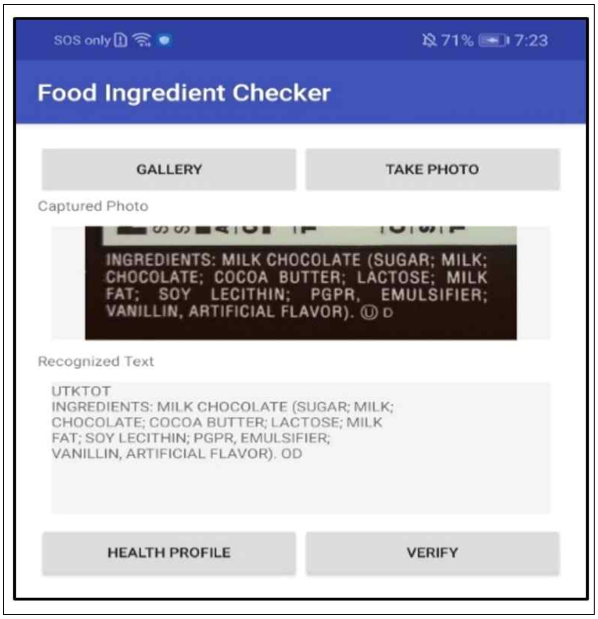


```
-----------------------------+----------------------------------------------------------
 disease_name                | disease_ingredients
-----------------------------+----------------------------------------------------------
 Lupus                       | alfalfa, garlic, saturated fat, trans fat, white potato,tomato,
 Coeliac                     | bread, pasta, cereal, biscuit, cracker, oat, gluten, barley, ry
 Asthma                      | egg, cow milk, peanut, soy, wheat, fish, shrimp, shellfish, tre
 Eczema                      | citrus, dairy, egg, gluten, wheat, vanilla, cloves, cinnamon, t
 Anaphylaxis                 | egg, peanut, walnut, pecan, soy, wheat, fish, shellfish, cow mi
 Birch Pollen                | almond, apple, apricot, carrot, celery, cherry, hazelnut, kiwi,
 Ragweed Pollen              | banana, cucumber, melon, cantaloupe, honeydew, watermelon, whit
 Timothy and Orchard Grass   | cucumber, watermelon, orange, peach, peanut, tomato, white pota
 Mugwort Pollen              | apple, bell pepper, broccoli, cabbage, carrot, celery, cauliflo
 Histamine Intolerance       | alcohol, pickle, cheese, salami, ham, sausages, shellfish, chic
-----------------------------+----------------------------------------------------------

 hot pepper, eggplant, alcohol, green tea
e, wheat, farina, graham flour, semolina, durum, cous cous, spelt
e nut, dried fruit, potato, wine, beer, lime, lemon
omato, beans, black tea, chocolate, cocoa, nut, pea, seed, shellfish, soybean, soy, green apple, carrot, hazelnut, pear
lk, wheat, shrimp, lobster, crab
 peach, peanut, pear, plum, potato, soybean, anise, caraway, coriander, fennel, parsley
e potato, zucchini
to, zucchini
wer, garlic, onion, peach, aniseed, black pepper, caraway seed, coriander, fennel, mustard, parsley, chard
kpea, soy bean, peanut, walnut, cashew nut, chocolate, cocoa, vinegar
```

그림 12. 질병명-구성성분 테이블

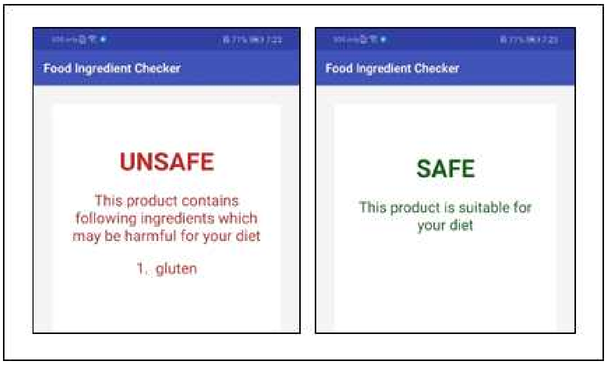
Fig. 12. Disease's-Ingredients table



INGREDIENTS: MILK CHOCOLATE (SUGAR; MILK; CHOCOLATE; COCOA BUTTER; LACTOSE; MILK FAT; SOY LECITHIN; PGPR, EMULSIFIER; VANILLIN, ARTIFICIAL FLAVOR). ⓤ D

UTKTOT
INGREDIENTS: MILK CHOCOLATE (SUGAR; MILK;
CHOCOLATE; COCOA BUTTER; LACTOSE; MILK
FAT; SOY LECITHIN; PGPR, EMULSIFIER;
VANILLIN, ARTIFICIAL FLAVOR). OD

그림 13. 문자 인식 출력
Fig. 13. Example of Text Recognition Output



UNSAFE

This product contains following ingredients which may be harmful for your diet

1. gluten

SAFE

This product is suitable for your diet

그림 14. 모바일 어플리케이션에 표시된 식품의 안전성
Fig. 14. Safeness Notice of the Product on UI of the Mobile Application

IV. Conclusion

A software system was proposed to help food intolerance patients suffering from allergies or immune disorder by informing them if the packaged food in question contains harmful ingredients for the user. The approach taken in this paper is to take the image of the ingredients list printed on the envelop of the packaged food, and adopt optical character recognition technology to extract the ingredients of the product. This approach seems not quite efficient since correct character recognition is not always possible, especially in noisy environment like the task of ingredient information retrieval. However, as few packaged food companies provide on-line ingredients information, the OCR is our best measure to retrieve ingredients information.

To reduce the effects of noise, and to increase the recognition rate of the correct ingredients, we applied two post-processing procedures to the output of the OCR process: the tokenization, and the correction and extraction of ingredients using dictionaries. The tokenization separates OCR-retrieved string into word tokens and removes irrelevant symbols and characters. Then the separate tokens are compared with english dictionay and proprietary dictionary in sequence. The english dictionary converts possible mis-recognized word unit tokens into correct words. The proprietary dictionary contains domain specific words and is used to select only meaningful words in the current task, i.e., ingredients in this study. The proprietary dictionary is open in the sense that the administrator can append previously unknown tokens when they are found. The proposed scheme was implemented in a server-client environment. The client is a smart phone application which sends image of the ingredient list to the server, and receives and shows the safeness information to the user. Open source softwares are assembled for the processes OCR, tokenization, and dictionary comparison in the server. The server also contains a database with two tables, the user information table which keeps user's disease information, and the disease table which holds ingredients that can provoke the outbreak of the disease.

The server was built on MS Windows and python environment in a PC, and the client app is simulated on the Android studio. Using simulated user and disease tables, the system was tested with a few envelops of the packaged foods, and worked as expected. It is highly desirable that the packaged food companies provide their ingredient information in the public domain for the user's safety. The proposed method can be one of the alternatives until on-line ingredients information become available to the public. The method is expected to be useful in the domain of medicine, cosmetics, etc.

References

[1] A. Santos, Why the world is becoming more allergic to food, King's College London [internet]. Available: https://www.bbc.com/news/health-46302780/.

[2] M.P. Oria, Finding a Path to Safety in Food Allergy, National Academies of Sciences, Engineering and Medicine, 2016.

[3] S. Prajapati, Eatable: An Application That Helps People with Food Allergies Check and Locate Allergen-Free Food Products, Master's thesis, Harvard University, 2017.

[4] G. Avani, Food Allergy Scanner, Android Mobile App., [internet]. Available: https://play.google.com/store/apps/details?id=appinvento.aiavnig2005.FoodAllergyScannerv1_checkpoint1&hl=en.

[5] A. Schaefer, Food Allergy vs. Sensitivity: What's the Difference? [internet]. Available: https://www.healthline.com/health/allergies/food-allergy-sensitivity-difference#food-allergies.

[6] Wikipedia, Comparison of OCR Software, [internet]. Available: https://en.wikipedia.org/wiki/Comparison_of_optical_character _recognition_software.

[7] Wikipedia, Comparison of OCR Software, [internet]. Available: https://en.wikipedia.org/wiki/Comparison_of_optical_character_recognition_software.

[8] D. Bloomberg, Leptonica, [internet]. Available: http://www. leptonica.org/.

[9] Techopedia, Tokenization, [internet]. Available: https://www.techopedia.com/definition/13698/tokenization.

[10] L. Michelbacher, Multi-Word Tokenization for Natural Language Processing, Ph.D. dissertation, University of Stuttgart, 2013.

[11] Infochimps, Dataset English-Words List, [internet]. Available: https://github.com/dwyl/english-words.

[12] Names of Packaged Food Ingredients, Additives & Colors, U.S Food and Drug Administration, FDA (2010, April) Overview of Food Ingredients, Additives & Colors, [internet]. Available: https://www.fda.gov/food/food-ingredients-packaging/overview-food-ingredients-additives-colors.

[13] Z. Schollz, Food Ingredients, Food Identicon. [internet]. Available: https://github.com/schollz/food-identicon/blob/master/ingredients.txt.

[14] Team NLTK, (2019, Aug) NLTK tokenize package, [internet]. Available: http://www.nltk.org/api/nltk.tokenize.html#module-nltk.tokenize. mwe.

[15] Python bindings for the Enchant spellchecking system, [Internet]. Available: https://pypi.org/project/pyenchant/.

[16] Infochimps (2014) Dataset English-Words List, Retrieved From https://github.com/dwyl/english-words.

자라 카미스(Nurul Azzahra Putri Kamis)

2017년 : Management & Science University (Computer Forensic)

2020년 : Korea Maritime and Ocean University (Master, Computer Eng.)

※관심분야 : 인공지능 (AI), IoT, 텍스트마이닝 (Text-Mining) 등

신옥근(Ok-Keun, Shin)

1983년 : Graduate School, Busan Univ. (master)

2005년 : Universite de Franche-Comte (Ph.D. Computer Eng.)

1983년~1995년: ETRI

1995년~현 재: 한국해양대학교 해사 IT공학부 교수

※관심분야 : Signal Processing, Embedded System 등