

소프트웨어공학소사이어티 논문지

Journal of Software Engineering Society

VOLUME 28, NUMBER 2, December 2019

이더리움 스마트 컨트랙트 상태 모니터링 시스템의 설계 및 구현	홍준기, 김순태, 류덕산	1
실시간 인식 기반의 제품의 표시 정보 추출을 위한 모바일 어플리케이션 설계 및 구현	민경식, 최지수, 이철훈, 정동주, 이병정	7



이더리움 스마트 컨트랙트 상태 모니터링 시스템의 설계 및 구현[†]

(Design and Implementation of Ethereum Smart Contract State Monitoring System)

홍준기[‡] 김순태[§] 류덕산[¶]
(Joongi Hong) (Suntae Kim) (Duksan Ryu)

요약 블록체인 생태계에는 다양한 이해관계자들이 존재한다. 이더리움이 등장한 이후 스마트 컨트랙트를 활용한 거래가 많이 이루어지게 되었고 개발자뿐만 아니라 투자자, 은행, 기업, 일반 사용자 등등 더욱 폭넓은 이해관계자들이 참여하고 활동하고 있다. 하지만 다양한 이해관계자가 스마트 컨트랙트의 상태를 확인하기 위해서는 어렵고 복잡한 과정을 거쳐야 한다는 문제점을 가지고 있다. 상태 확인이 어렵게 된다면 스마트 컨트랙트에 대한 신뢰도가 낮아지게 되어 활용도가 낮아질 것이다. 개발자에게 있어서도 상태 확인이 어렵고 복잡하다면 자신이 개발한 스마트 컨트랙트의 테스트와 디버깅을 하는데 어려움을 겪어 높은 품질을 제공하기 힘들 것이다. 본 연구에서는 다양한 이해관계자와 개발자들이 스마트 컨트랙트의 상태를 쉽고 지속적으로 확인할 수 있으며 히스토리 데이터를 활용하여 분석할 수 있도록 하는 이더리움 스마트 컨트랙트 상태 모니터링 시스템의 설계 및 구현 방법을 제안한다.

키워드 : 이더리움, 스마트 컨트랙트, 상태, 이해관계자, 모니터링, 개발자

Abstract There are various stakeholders in the blockchain ecosystem. Since the emergence of Ethereum, many transactions have been made using smart contracts, and a wider range of stakeholders are participating, including not only developers, but also investors, banks, companies, and general users. However, various stakeholders have a problem in that it is difficult and complicated to check the state of smart contracts. If it becomes difficult to check the state, the reliability of the smart contract will be lowered and the utilization will be lowered. Also, if the state check is difficult and complicated for the developer, it will be difficult to provide high quality due to the difficulty of testing and debugging the smart contract developed by the developer. In this research, we propose a design and implementation method of the Ethereum Smart Contract State Monitoring System that enables various stakeholders and developers to easily and continuously check the state of smart contracts and analyze them using historical data.

Key words : Ethereum, Smart Contract, State, Stakeholder, Monitoring, Developer

1. 서론

블록체인(Blockchain)은 2009년 가상화폐인 비트코인(Bitcoin)의 탄생과 함께 세상에 나타났다[1]. 2015년에는 단순히 가상화폐로의 기능으로만 활용되는 것을 넘어서 스

마트 컨트랙트(Smart Contract)라는 개념을 도입한 이더리움(Ethereum)이 등장한다[2]. 이더리움은 스마트 컨트랙트를 기반으로 다양한 거래를 가능케 하였으며, 가상화폐를 기반으로 한 비즈니스가 가능하기에 개발자뿐만 아니라 투자자와 은행, 기업, 일반 사용자 등 다양한 이해관계자가 참여하고 활동하고 있다. 다양한 이해관계자가 참여하고 있는 프로젝트로는 대표적으로 DAO(Decentralized autonomous organization)가 있다.

스마트 컨트랙트를 활용하여 거래하기 위해서 이해관계자들은 작성한 내용이 요구사항대로 처리가 되는지 확인 하길 원할 것이며, 개발자 또한 자신이 작성한 로직이 설계한 대로 동작하는지 검증하고 싶어 할 것이다. 하지만 스마트 컨트랙트의 상태를 확인하기 위해서는 프로그래밍 지식이 필요하여 개발자 이외에는 어려울 수 있고, 복잡하고 번거로운 과정을 거쳐야 한다. 상태 확인을 위해서는 먼저

[†] 이 논문은 정부(정보통신기술진흥센터)의 재원으로 대학ICT연구센터육성지원사업의 지원을 받아 수행된 연구임(No. IITP-2018-2017-0-01628).

이 논문은 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2019R1G1A1005047)

[‡] 학생회원 : 전북대학교 소프트웨어공학과
rlwns012@jbnu.ac.kr

[§] 종신회원 : 전북대학교 소프트웨어공학과 교수
stkim@jbnu.ac.kr

[¶] 종신회원 : 전북대학교 소프트웨어공학과 교수
duksan_ryu@jbnu.ac.kr

논문접수 : 2019년 11월 4일
심사완료 : 2019년 12월 1일

스마트 컨트랙트 언어인 Solidity와 EVM(Ethereum Virtual Machine)의 메모리 구조를 이해하고 있어야 하며, 스마트 컨트랙트에서 선언한 상태의 타입과 순서에 따라 EVM의 메모리 구조에 맞게 값을 가져오고 변환하는 작업을 거쳐야 한다. 이와 같은 불편함은 스마트 컨트랙트의 상태에 대한 투명성을 낮춰 이해관계자들의 신뢰도를 떨어뜨리며, 개발자에게는 자신이 개발한 로직이 정상적으로 동작하는지 검증하기 어렵게 한다.

이더리움 네트워크의 스마트 컨트랙트에 대한 정보를 제공하는 방법으로 여러 모니터링 웹어플리케이션이 존재하고 있다. 가장 대표적으로 이더리움에서 자체적으로 제공하는 이더스캔(Etherscan)이 존재하며, 시각적인 형태로 정보를 제공하는 알비오(Alvio), Solidity IDE인 리믹스(Remix)도 존재한다. 하지만 스마트 컨트랙트의 상태를 상세하게 보여주는 기능을 하고 있지 않다.

개발자에게 스마트 컨트랙트에 대한 검증을 돕기 위한 연구로는 SmartInspect이 있다. 이더리움에 배포된 스마트 컨트랙트의 상태를 소스 코드의 변수명과 타입에 맞게 값을 변환해주어 소스 코드의 내용대로 파악할 수 있게 도움을 준다. 하지만 SmartInspect은 특정 함수간의 상태만을 제공해주기 때문에 지속적인 파악은 힘들다는 단점을 가지고 있다.

본 연구에서는 이해관계자들이 스마트 컨트랙트의 상태 확인이 어렵고 복잡하여 발생하는 문제를 해결하고자 블록 생성 시마다 자동으로 처리되고 소스 코드에서 작성한 내용대로 상태 확인이 가능한 모니터링 설계 및 구현 방법을 제안한다. 지속적인 상태 변화 파악을 위하여 모니터링 방법을 선택하였으며, SmartInspect[4]의 상태 변환 과정을 기반으로 하여 동작한다. 이해관계자들에게 스마트 컨트랙트의 검증과 확인을 위해서 과거의 상태 변화 히스토리를 남겨두고 이전 상태를 확인할 수 있도록 지원한다. 다시 말하자면 이해관계자들은 본 연구에서 제안하는 기법을 통해 별다른 과정 없이 스마트 컨트랙트의 상태 정보를 확인 가능하며, 지속해서 변화하는 상태 정보를 보고 요구사항에 맞는 로직이 수행되었는지 파악할 수 있도록 지원받는다.

이더리움 스마트 컨트랙트의 상태는 이해관계자들에게 올바른 거래 수행을 위해 매우 중요한 정보이다. 본 연구에서 제안하는 기법을 통해 중요한 상태 정보를 파악하기 쉽고 용이하게 하였으며, 순간적인 상태가 아닌 지속적으로 변화하는 상태를 보여줌으로써 진행한 거래를 바로 확인할 수 있게 하였다. 스마트 컨트랙트 상태의 히스토리를 저장하여 오류 또는 결함이 발생하였을 때 해당 문제에 추적이 가능하고 개발자에게는 코드의 디버깅 요소가 어떤 곳인지 확인할 수 있도록 하였다. 이는 현재 이더리움에 배포된 스마트 컨트랙트의 검증을 위한 도구가 부족한 상황에서 디버깅 도구로서의 역할을 수행할 수 있다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 배경 지식과 관련 연구에 대해 알아보고 3장에서는 스마트 컨트랙트 상태 확인을 위한 모니터링 기법에 대해 이야기한

다. 4장에서는 결론 및 향후 연구에 대해 이야기하고 마무리한다.

2. 배경 지식 및 관련 연구

2.1 배경 지식

본 연구에서는 생소할 수 있는 블록체인과 스마트 컨트랙트에 대해 다룬다. 이해를 돕기 위해서 블록체인과 스마트 컨트랙트에 대한 개념을 간략하게 설명한다.

2.1.1 *블록체인(Blockchain)*: 블록체인은 탈중앙화된 분산 시스템이다. 탈중앙화란 기능 수행을 위해 하나의 중앙기관이 존재하지 않고 다수의 참여자가 함께 수행해 나가는 것을 의미한다. 기본적으로 블록체인 네트워크는 P2P 네트워크로 구성이 되며, 합의 알고리즘을 활용하여 공동된 과정을 거쳐 블록을 생성한다. 해시의 활용은 블록체인에서 핵심적이며 해시와 함께 다수의 참여자가 같은 데이터를 통해 같은 처리를 한다는 점이 결합하여 데이터의 위변조를 방지한다.

블록체인에는 비트코인과 이더리움 등이 가장 대표적인 어플리케이션이며, 비트코인은 블록체인의 탄생을 알리고 현재 가장 인기 있는 가상화폐이다. 이더리움은 2015년에 등장하였으며, 스마트 컨트랙트의 적용으로 블록체인의 커다란 변화를 이끌었다. 위변조될 수 있는 스마트 컨트랙트를 블록체인에 배포함으로써 작성 내용을 무결하게 유지할 수 있게 되었으며, 블록체인의 기능 확장을 가능하게 했다.

블록체인은 많은 사람의 관심을 끌었으며, 특히 이더리움은 스마트 컨트랙트를 가능하게 하여, 단순히 개발자들뿐만 아니라 은행, 투자자, 기업, 일반 사용자들이 참여하여 투자와 펀드 등 다양한 활동을 이어가고 있다. 다양한 이해관계자가 참여하는 대표적인 예로 DAO(Decentralized autonomous organization) 프로젝트가 존재하고 있다.

2.1.2 *스마트 컨트랙트(Smart Contract)*: 현실의 계약을 프로그래밍 코드를 활용하여 작성한다는 개념이며, 조건이 충족되면 계약서에 있는 내용대로 자동으로 처리된다[3]. 하지만 디지털 문서 같은 경우 위변조가 쉬우므로 당시에는 활용되기에는 한계점이 존재하였다. 이더리움에서는 블록체인에 스마트 컨트랙트 배포하여 활용할 수 있는 개념을 제안하였고, 블록체인에 배포되어 저장된 데이터는 위변조가 어렵다는 점을 활용하여 한번 작성되어 배포되었다면 스마트 컨트랙트는 무결성을 보장받을 수 있다.

이더리움에서 작성되는 스마트 컨트랙트는 Solidity 언어를 활용하며, EVM 위에서 동작한다. Solidity는 Javascript와 Python 등에 영향을 받은 언어이며, 정수형 데이터를 취급하지 않고 Address와 mapping 등 독자적으로 제공하는 데이터 타입이 존재한다. EVM은 JVM과 비슷한 개념이며, 블록체인의 특성에 맞는 제약이 존재한다.

```

1  pragma solidity ^0.4.16;
2
3  contract TestContract {
4      int64 age;
5      int64 num;
6      int64 num2;
7
8      bytes32 bytedata1;
9      bytes32 bytedata2;
10
11     string name;
12     address chairperson;
13     uint256 time;
14
15     function TestContract() public {
16         chairperson = msg.sender;
17         age = 100;
18         num = 200;
19         num2 = 300;
20
21         bytedata1 = "a";
22         bytedata2 = "b";
23         bytedata1 = "c";
24         name = "Hong";
25
26         time = now;
27     }
    
```

그림 1 Solidity를 활용한 TestContract 스마트 컨트랙트 예제 코드

[그림 1]은 Solidity 언어를 활용하여 작성된 TestContract 라는 스마트 컨트랙트의 예제 코드이다. 스마트 컨트랙트에서 상태란 전역변수로 선언되어 이더리움 상에서 지속적으로 유지되는 변수를 의미한다. 소스 코드 확인해 볼 수 있듯이 int64 age, int64 num, byte32 bytedata1, string name 등이 전역으로 선언되어 있으며, 이 변수들을 상태로 정의한다. 해당 스마트 컨트랙트는 생성자를 활용하여 블록체인에 배포됨과 동시에 상태 값이 초기화가 되도록 작성해두었다. 해당 예제 코드에서는 상태를 파악하는 것에 초점을 맞추기 위해 상태 이외의 함수, 이벤트 등의 정보는 생략하였다.

2.2 관련 연구

이더리움에 대한 정보를 파악하고 개발자들의 테스트와 디버깅을 위한 도구들을 알아보고 현재 진행되고 있는 연구에 대해 알아보도록 한다.

2.2.1. 이더스캔(Etherscan): 이더리움은 블록의 생성, 트랜잭션의 발생, 스마트 컨트랙트 활용 정보 등 블록체인의 현황 파악을 위해 이더스캔(Etherscan)을 기본적으로 제공하고 있다. 이더스캔에서는 이해관계자들이 스마트 컨트랙트를 사용했다는 트랜잭션 정보를 확인해 볼 수 있다. 하지만 이해관계자에 의해 발생한 트랜잭션이 변화시킨 스마트 컨트랙트의 상태 변화에 대한 정보는 인코딩된 형태로 제공되어 이해하기 어려우며 전체의 상태를 파악할 수 없다.

2.2.2 리믹스(Remix): 리믹스는 이더리움 스마트 컨트랙트 개발을 위한 Web 기반의 IDE다. 리믹스에서는 개

발뿐만 아니라 이미 배포된 스마트 컨트랙트의 정보도 확인해 볼 수 있다. 하지만 상태를 확인하기 위해서는 Solidity에서 상태가 public으로 선언되어 있어야 하거나 개발자가 get 함수를 만들어야만 확인할 수 있다. get 함수로 만들어 제공할 경우 사용자는 이더리움의 수수료 개념인 gas를 지불해야 하므로 지속해서 사용해야 한다면 적절하지 못한 방법일 것이다. 또한, OPCODE의 형태로 디버깅 도구로 제공하지만, 해당 지식을 이해하고 있지 않다면 이해하기 어려운 내용이다.

2.2.3 SmartInspect: SmartInspect는 개발자가 스마트 컨트랙트의 디버깅이 어렵다는 점을 지적하며 Solidity 소스 코드에서 선언한 상태 변수를 기반으로 하여 상태 값을 확인하는 방법을 제안하였다[4]. 이더리움 상에 스마트 컨트랙트의 상태 정보는 인코딩(Encoding)되어 저장된다. 따라서 실제 소스 코드와 받아온 데이터를 직접 연관 지어 볼 수 없으며, 데이터 변환과정을 거쳐야 최종적으로 사용자가 원하는 형태로 볼 수 있게 된다. SmartInspect 는 해당 과정 해결을 위해 mirror-based architecture를 설계하고 XML, JSON 등 파일의 형태로 만들어 쉽게 확인 가능한 것을 보여주었다. 하지만 SmartInspect 는 순간적인 상태만을 제공하며, 상태의 변화를 자동으로 파악하고 히스토리를 쌓는 기능이 없기에 잦은 변화가 있는 블록체인 생태계에는 적합하지 않은 형태이다.

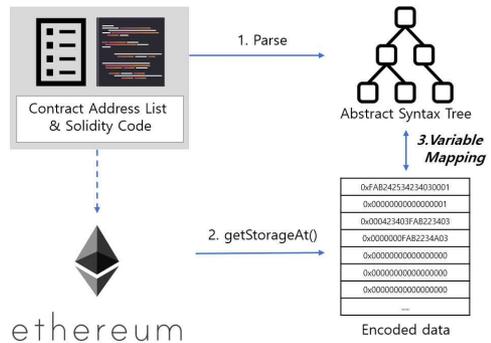


그림 2 SmartInspect에서 제안한 스마트 컨트랙트 상태 확인 방법

SmartInspect는 [그림 2]와 같은 형태로 소스 코드와 상태 값을 매핑한다. 사용자로부터 Solidity 소스 코드와 Address를 입력받고 소스 코드는 Parser를 활용하여 상태 변수를 추출하고 Address를 활용하여 이더리움으로부터 상태 값을 가져온다. EVM의 메모리 구조와 상태 변수의 타입 및 순서를 고려하여 매핑을 진행하고 실제 값을 매핑하는 방식이다. 해당 방법은 본 연구의 기반으로 활용된다.

3. 이더리움 스마트 컨트랙트 상태 모니터링 시스템의 설계 및 구현

이 장에서는 본 연구의 접근 방법인 이더리움 스마트 컨트랙트 상태 모니터링 시스템의 설계 및 구현에 관해 이야기한다. 상태에 대한 모니터링에 대한 이해를 돕기 위해 2.1.2 장에서 제시한 [그림 1]의 Solidity 스마트 컨트랙트 예제를 활용하여 설명을 이어나간다.

본 연구에서는 이해관계자와 개발자에게 상태 정보를 전달하기 위해 적합한 형태가 모니터링 시스템이라고 판단하였다. 이더리움에는 수많은 트랜잭션이 발생하고 트랜잭션에 의해 스마트 컨트랙트의 상태는 실시간으로 변화하고 있다. 지속해서 변화하는 데이터를 프로파일링하고 분석, 진단, 디버깅해야 하는데 모니터링 시스템은 이와 같은 기능을 수행하기 위해 발전해왔다 [5].

모니터링 시스템은 Observe와 Analyze로 두 개의 부분으로 구성된다. 하나는 소프트웨어에 대한 정보를 파악하고 해당 정보를 지속해서 관찰 기능을 수행하는 Observe 부분과 Observe에서 파악한 정보를 기반으로 도메인별로 필요한 분석을 진행하고 분석한 내용에 따라 대처를 가능케 하는 Analyze 부분이 있다. 본 연구에서는 Observe 부분에 초점을 두었다.

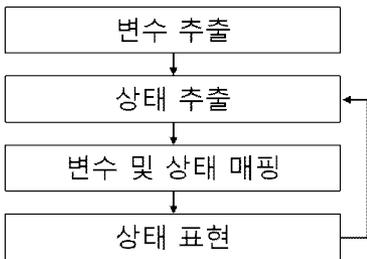


그림 3 모니터링 개요

[그림 3]에서는 모니터링을 위한 단계의 개요를 나타내고 있다. 단계는 변수 추출, 상태 추출, 변수 및 상태 매핑, 상태 표현 등 4단계로 구성이 되어있으며, 모니터링의 기본적인 구성인 Observe와 Analyze 중 Observe 부분에 해당하는 단계이다. 본 연구에서는 스마트 컨트랙트의 상태를 파악하기 위한 모니터링 기법에 관해 설명하고 있으며 블록체인에 배포된 코드는 변경이 불가능하므로 문제가 발생하여도 대처할 수 없다는 특성이 있다. 따라서 모니터링 중 프로파일링에 초점을 두어 연구를 진행하였다. 아래에서는 단계별로 수행되는 상세 내용을 서술한다.

첫 번째로 변수 추출 단계에서는 사용자로부터 Solidity source code와 Address를 입력받으며, 입력받은 Source code를 Antlr4[6]를 활용하여 만든 Solidity Parser로 파싱을 진행한다. 파싱을 통해 상태로 선언된 변수들을 가려내고 변수들의 타입과 식별자 이름을 JSON의 형태로 Token 화한다. [그림 4]는 Solidity Parser를 활용하여 만들어낸 Token JSON이다.

```

[{"type": "int64", "name": "age", "size": 64},
{"type": "int64", "name": "num", "size": 64},
{"type": "int64", "name": "num2", "size": 64},
{"type": "bytes32", "name": "bytedata1", "size": 32},
{"type": "bytes32", "name": "bytedata2", "size": 32},
{"type": "string", "name": "name", "size": 256},
{"type": "address", "name": "chairperson", "size": 160},
{"type": "uint256", "name": "time", "size": 256}]
    
```

그림 4 Variable Token JSON

Token에는 상태의 타입, 식별자 이름, 타입의 크기로 구성이 된다. [그림 1]의 예제에서 선언된 age, num, num2 등의 상태들이 Token JSON의 형태로 변환된 것을 확인할 수 있다. Token JSON은 스마트 컨트랙트 별로 단 하나만이 생성되며, 등록과 동시에 DB에 Source code와 Address, Token JSON이 함께 저장된다. 데이터는 블록이 생성될 때마다 다시 불러와 다음 과정을 수행하기 위해 저장되어 관리된다.

두 번째로 상태 추출 단계를 진행한다. 이더리움 네트워크로부터 Web3의 getStorageAt() 함수를 활용하여 스마트 컨트랙트의 상태를 추출한다. EVM의 메모리 구조에 맞춰 저장된 16진수의 값을 얻어 오는 과정이다. [그림 5]와 같이 이더리움으로부터 인코딩된 데이터(Encoded data)를 불러오는 과정이다. 이렇게 추출된 16진수의 상태 정보는 사용자들에게 당장 쓸 수 있는 형태로 제공되는 것이 아니며 값을 파악하기 어렵다. 또한, 상태의 정확한 값을 알기 위해서는 상태의 순서와 타입에 맞게 변환하는 과정이 필요하다.

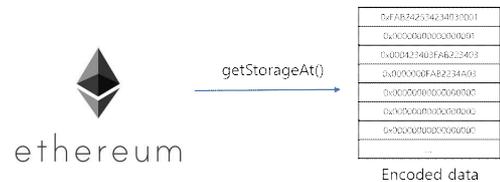


그림 5 Web3를 활용하여 이더리움 네트워크에 배포된 스마트 컨트랙트의 상태를 추출한다.

세 번째 과정은 변수 및 상태 매핑 작업이다. 이더리움 네트워크에서 가져온 인코딩된 데이터는 Source code에서 선언한 변수명과 1:1로 연결되지 못한다. [5]에서 제안하는 방법을 활용하여 [그림 7]와 같이 실제 값과 선언된 상태가 연결될 수 있도록 매핑하는 작업을 진행한다. EVM은 여러 개의 Slot을 가지고 있으며, 한 개의 Slot은 256bit로 구성된다. 데이터 타입별로 사이즈가 다르며, 사이즈에 맞춰 어떤 상태가 어디에 저장되어 있는지를 가려내야 한다. 또한, 모든 변수가 사이즈에 맞춰 저장되는 것이 아니므로 이에 따른 처리를 진행하여야 한다. 예제의 int64 age, int64 num, int64 num2 등등이 메모리

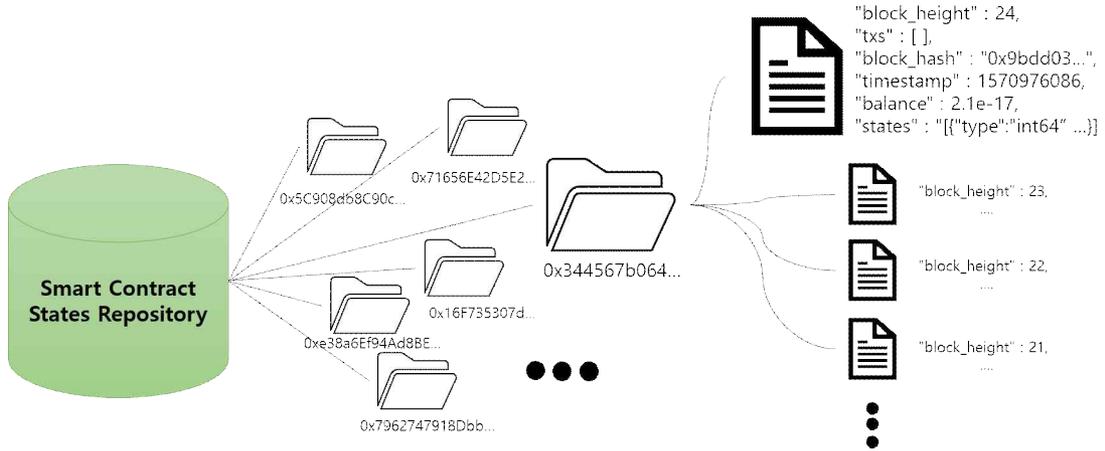


그림 6 스마트 컨트랙트별 상태 저장 방법

구조에 맞게 매핑이 진행된다.

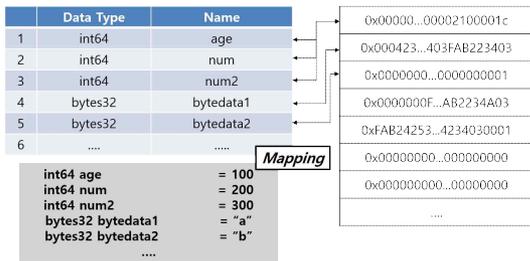


그림 7 EVM 메모리는 한 Slot이 256bit로 구성되어 있으며, 여러 개의 Slot으로 구성되어 있다. 상태는 타입별로 다양한 크기를 가지고 있으며 Slot에는 크기에 맞게 여러 개의 상태가 들어갈 수 있게 설계되어 있다.

마지막으로 사용자에게 스마트 컨트랙트의 상태를 출력하는 단계이다. 출력되는 정보는 앞의 과정을 거쳐 만들어진 매핑된 상태 정보와 함께 현재 생성된 블록의 블록 번호, 스마트 컨트랙트를 활용한 트랜잭션, 보유 이더량을 추가하여 하나의 데이터로 구성한다. 사용자에게 해당 상태 정보는 과거의 정보도 함께 보여주게 된다. 과거의 정보는 Smart Contract States Repository에 스마트 컨트랙트 별로 관리된다. 상태 정보를 출력한 후 블록체인으로부터 모니터링을 위한 프로토타입은 블록이 생성될 때까지 대기한다. [그림 2]에서 확인할 수 있듯이 블록이 생성될 경우 두 번째 과정인 상태 추출과정부터 다시 진행한다. 이 과정은 한 번으로 끝나지 않고 이더리움 네트워크에 블록이 쌓이는 한 무한히 반복하며 스마트 컨트랙트의 상태 변화의 지속적인 모니터링을 제공한다.

블록이 생성될 때마다 스마트 컨트랙트 별로 상태 정보를 DB에 저장하게 된다. 데이터는 [그림 6]와 같은 구조로 저장되고 사용자에게 의해 스마트 컨트랙트가 등록된

순간부터 저장한다. 해당 저장소는 Solidity source code와는 별개의 저장소이며, 스마트 컨트랙트 하나당 블록이 쌓인 수 만큼 상태를 저장하게 된다. 상태 데이터 구조에 관해 설명하자면 우선 상태는 블록이 생성될 때마다 모니터링이 되기 때문에 구분하기 위한 기준으로 블록의 번호를 선정하였다. 더불어 블록의 해시를 함께 저장한다. 트랜잭션의 경우 블록에 담겨 있던 모든 트랜잭션을 저장하는 것이 아닌 스마트 컨트랙트를 활용한 트랜잭션만을 저장하게 된다. 저장되는 내용은 트랜잭션의 해시값이 들어가게 된다. 스마트 컨트랙트의 상태 중 하나인 이더량도 함께 저장하며, 모니터링 시간을 추가하여 구성하였다. 마지막으로 앞의 작업을 수행하여 얻어낸 매핑된 상태를 JSON의 형태로 담아 저장한다. 상태 정보에는 Source code에 선언된 모든 상태가 표현되며 실제 값이 어떤 값인지도 볼 수 있다.

[그림 8]은 최종적으로 사용자가 확인 가능한 텍스트 기반의 스마트 컨트랙트 상태이다. 상태의 데이터 구조는 앞에서 언급했듯이 이더량, 블록 해시, 블록 번호, 상태 정보, 스마트 컨트랙트와 관련된 트랜잭션 해시, 타임스탬프로 구성되어 있는 것을 확인할 수 있다. 또한, 상태 정보를 들여다보면 Source code에서 상태로 선언되었던 변수의 이름과 타입을 확인할 수 있고 블록이 생성된 시점에 유지하고 있는 실제값을 realValue에서 확인할 수 있다.

```

▼ (3) [{"-"}, {"-"}, {"-"}]
▶ 0: {txs: Array(0), _id: "Sda331245ee9fe6148f6279c", block_height: 20, block_hash: "0x96ec0b0ad3a486af00b81a9fe530"}
▶ 1: {txs: Array(0), _id: "Sda331365ee9fe6148f6279d", block_height: 22, block_hash: "0x6bdaa0fce8e4bb949b3440ad7022"}
▼ 2:
  balance:
  2.1e-17
  block_hash: "0x544348546767b2d4accb0a9d9a7d1c20893d93c9a57c49521f0a2321b831a38a"
  block_height: 23
  states: "[{"type":"int64","name":"age","size":64,"basetype":null,"realValue":100},{type:"int64","name":"num","timestamp":1570976078}]"
  txs: []
  _v: 0
  _id: "Sda331505ee9fe6148f6279e"
  __proto__: Object
length: 3

```

그림 8 스마트 컨트랙트 상태 확인 텍스트 기반 모니터링 기법 출력 결과물

4. 결론 및 향후 연구

이더리움 생태계에는 다양한 이해관계자가 존재하며, 스마트 컨트랙트를 활용하여 다양한 계약과 거래를 진행한다. 하지만 모든 이해관계자가 스마트 컨트랙트가 요구 사항대로 동작하고 있는지 파악하기 위해서는 프로그래밍 지식이 필요하며, 데이터 변환을 위한 복잡한 과정을 거쳐야 한다. 본 연구에서는 이더리움의 스마트 컨트랙트에 대한 이해가 부족하더라도 현재 상태가 어떤 값이고 어떻게 변화하고 있는지 파악할 수 있는 모니터링 시스템 설계 및 구현 방법에 대해 이야기하였다. 더불어 실제로 데이터가 어떻게 보이는지에 대한 텍스트 형태의 예시를 보여주었다. 본 연구를 통해 우리는 이해관계자들의 스마트 컨트랙트에 대한 이해도가 증가할 것으로 기대하고 있으며, 상태에 대한 투명성이 증가하여 당사자 간에 신뢰를 높여 계약을 진행할 수 있게 지원 도구로서의 역할을 할 수 있으리라 기대하고 있다.

향후 계획으로는 모니터링 시스템에서 데이터에 대한 표현을 텍스트로만 보여주는 것을 그래프와 타임라인 등을 활용하여 사용자들이 더욱 쉽게 인식할 수 있는 연구를 진행할 것이다. 또한, 다양한 이해관계자들은 거버넌스를 형성할 수 있고 거버넌스에 필요한 다양한 정보를 제공할 수 있는 모니터링 기능을 연구하고 확장할 계획이다. 관찰된 데이터를 분석하여 악의적인 사용과 이상 현상이 발생할 수 있는 점을 경고하고 파악하기 위한 기능을 연구할 예정이다.

참 고 문 헌

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." 2009. [Online]. Available: bitcoin.org

[2] Buterin, Vitalik. "Ethereum white paper." GitHub repository (2013): 22-23.

[3] Szabo, Nick. "Smart contracts." Unpublished manuscript (1994).

[4] Bragagnolo, Santiago, et al. "SmartInspect: solidity smart contract inspector." 2018 International Workshop on Blockchain Oriented Software Engineering (IWBOSE). IEEE, 2018.

[5] Delgado, Nelly, Ann Q. Gates, and Steve Roach. "A

taxonomy and catalog of runtime software-fault monitoring tools." IEEE Transactions on software Engineering 30.12 (2004): 859-872.

[6] Parr, Terence. The definitive ANTLR 4 reference. Pragmatic Bookshelf, 2013.



홍 준 기
2019년 전북대학교 소프트웨어공학과 졸업(학사) 2019년~현재 전북대학교 석사 재학



김 순 태
2007년 서강대학교 컴퓨터공학과 졸업(석사) 2010년 서강대학교 컴퓨터공학과 졸업(박사) 2014년~현재 전북대학교 소프트웨어공학과 부교수



류 덕 산
2012년 카이스트 및 카네기멜론대학교 소프트웨어공학 복수학위 졸업(석사). 2016년 카이스트 전산학부 졸업(박사). 2018년~현재 전북대학교 소프트웨어공학과 조교수

실시간 인식 기반의 제품의 표시 정보 추출을 위한 모바일 어플리케이션 설계 및 구현[†]

(Design and Implementation of Mobile Application for Extracting Information of Products based on Real-Time Recognition)

민 경 식 [‡] 최 지 수 [§] 이 철 훈 [§] 정 동 주 [§] 이 병 정 [¶]

(Kyeongsic Min) (Ji-Su Choi) (Cheolhun Lee) (Dongju Jung) (Byungjeong Lee)

요 약 소비자는 제품을 소비할 때 여러 외적 요소를 보지만 가장 중요한 것은 제품의 기능과 성분이 다. 특히, 화학품은 오남용을 방지하고자 구성 성분을 정확히 파악하는 것이 중요하며, 체계적인 관리 방법이 필요하다. 하지만 소비자들이 화학 성분을 체계적으로 관리하는 방법이 부족하며, 이를 수동으로 관리하기에는 많은 노력이 따른다. 따라서, 본 논문에서는 소비자들이 화학 성분을 체계적으로 관리하는데 활용될 수 있는 실시간 인식 기반의 제품의 표시 정보 추출 기법을 제안하고, 해당 기법의 모바일 어플리케이션 프로토타입을 설계 및 개발한다. 개발된 프로토타입은 추후 복수 개의 화학 제품들을 체계적으로 관리할 수 있도록 발전 가능하다.

키워드 : 실시간 인식, 제품 표시 정보 추출, YOLO, Tesseract

Abstract Consumers see many external factors when they buy a product, but the most important thing is the function and composition of the product. In particular, it is important to accurately identify the components of chemicals in order to prevent misuse, and a systematic management method is required. However, there is a lack of a systematic way for consumers to manage chemical components, and much effort is required to manage them manually. Therefore, in this paper, we propose a method of extracting product's information of real-time recognition based products that can be used to systematically manage chemical components, and design and develop a mobile application prototype of the method. The prototype can be developed to systematically manage multiple chemical products in the future.

Key words : Real-time recognition, Extracting product's information, YOLO, Tesseract

1. 서 론

소비자는 제품을 소비할 때 여러 외적 요소를 보지만 가장 중요한 것은 제품의 성분이다. 특히, 그 중에서도 화학품은 구성성분의 성분이 중요하다. 화학품의 대표적인 예로는 건강 보조 식품, 의약품, 화장품 등이 있다. 이런 제품들은 잘못 사용할 시 신체에 나쁜 영향을 줄 수 있어 특별히 더 주의가 필요하지만 이런 화학품의 구성 성분들은 일반 사용자들이 확인하기는 어렵다. 소비자들은 이런

화학품을 잘 알지 못하고 사용하는 과정에서 오용 및 남용의 우려가 있기 때문에 제품의 구성 성분을 정확하게 파악하고 사용하여 부작용을 막아야한다.

이미지를 인식할 수 있는 텍스트로 바꾸는 방법은 많이 소개되었고 여러 방법으로 활용되고 있지만, 이 방법들은 저마다 여러가지 제한사항이 있다. 따라서 본 논문에서는 실시간 객체 추적 기술을 통해 화학품에 있는 구성 성분 정보를 인식하고 어떤 성분이 어느정도 들어있는지 알려주는 앱을 설계 및 개발하여 그 기법의 유효성을 입증한다. 본 연구는 아래의 동기로부터 발전하였다.

- 사용자들은 다양한 화학품을 선택할 때 구성 성분은 대부분 파악하지 못한다. 따라서 알지 못하는 성분에 의해 부작용이 생길 수 있다.
- 전문가가 아닌 일반 사용자들이 모든 화학품의 구성 성분을 파악하는 것은 사실상 불가능하다. 이를 해결하기 위해 체계적이고 쉽게 볼 수 있는 어플리케이션

[†] 이 논문은 이 논문은 2019년도 정부(과학기술정보통신부)의 재원으로 한국연구재단-현장맞춤형 이공계 인재양성 지원사업의 지원을 받아 수행된 연구임(NRF-2017H1D8A1030582).

[‡] 학생회원 : 서울시립대학교 컴퓨터과학부

[§] 비 회원 : 서울시립대학교 컴퓨터과학부

[¶] 중신회원 : 서울시립대학교 컴퓨터과학부 교수

bjlee@uos.ac.kr

논문접수 : 2019년 11월 19일

심사완료 : 2019년 12월 1일

이 필요하다.

- 구성 성분을 분석하는 기존 연구들은 실시간으로 표시 정보를 인식하기에는 비효율적이고 비현실적인 기법들을 제안한다.
- 본 논문은 다음의 기여도를 갖는다.
- 시중에 있는 화학품의 구성 성분을 실시간으로 파악 가능하고 각 성분의 정보를 알 수 있게 한다.
- 구성 성분을 파악하기 위해 객체 추적을 할 때, 객체 추적을 단단계로 인식하여 실시간으로 구성 성분의 확인이 가능하다.
- 화학품 중에서 특히 신체와 관련 있는 제품의 구성 성분과 함량을 추출하여 제품의 성분을 정확히 알고 사용 가능 하도록 하는 어플리케이션을 설계 및 개발 하는 방법을 제안한다.

앞으로의 본 논문은 다음 구성으로 이루어진다. 2장에서는 본 논문을 이해하기 위한 배경지식을 설명한다. 3장에서는 본 논문과 관련된 연구를 소개하고, 4장에서는 본 논문에서 제안하는 표시 정보 단단계 실시간 인식 기법을 상세하게 서술한다. 그리고 5장에서는 제안한 기법을 이용한 사례연구에 대해 서술하고, 6장에서는 본 논문에서 제안하는 모바일 어플리케이션 설계 방법에 대해 서술한다. 7장에서는 본 논문에서 구현하는 모바일 어플리케이션 구현에 대해 서술한다. 마지막으로 8장에서는 본 논문의 결론으로 마무리한다.

2. 배경 지식

객체 탐지는 탐지하고자 하는 객체를 라벨을 이용하여 어느 위치에(x,y) 어느 정도의 크기(w,h) 있는지를 탐지하고 파악하는 기법이다. 주어진 영상을 객체화 하고 객체화 된 정보를 인식하려는 연구는 계속해서 이루어지고 있다. 특히, 굴절, 왜곡, 빛 번짐 등의 요인으로 낮아진 품질의 영상에서도 빠르게 인식을 목표로 하는 연구는 현재 진행형이다. 대부분의 객체 탐지 기법들은 영상에서 객체를 바운딩 박스(Bounding Box)로 나타낸다. 이렇게 나타낸 바운딩 박스를 라벨을 이용하여 탐지된 객체가 어떤 것인지 표시한다.

현재 많이 연구되고 사용되는 영상 객체 탐지 기법은 대표적으로 R-CNN, Fast/Faster R-CNN, YOLO(You Only Look Once)가 있다. 특히, 이 중에서도 YOLO는 현재 가장 많이 사용되며 빠른 속도를 보여준다[1, 3].

R-CNN은 CNN을 이용하여 분류한다. 이때, input 이미지를 sub image로 추출하는 기법을 사용한다. 그리고 SVM(Support Vector Machine)을 이용하여 각 Object를 분류한다. Fast/Faster R-CNN은 R-CNN에서 병목현상을 개선한 것이다. Proposal들이 각각 CNN을 거치지 않고 RPN(Region Proposal Networks)을 이용하여 전체 이미지가 한번에 CNN을 지나게 한다[5].

YOLO는 기존의 R-CNN과 Fast/Faster R-CNN과 달리 전체의 이미지를 하나의 신경망에 적용한다. YOLO는 신경망을 이용하여 이미지를 각각의 바운딩

박스로 분할하고 각 바운딩 박스의 확률을 예측한다. 분할된 바운딩 박스에는 가중치가 적용된다. 다른 기법들과 달리 YOLO는 이미지의 단편이 아닌, 이미지 전체를 보고 객체를 예측한다. 또한, 최대 처리속도가 45FPS (Frame Per Second)까지 달하며 이로 인해 영상에서 객체를 추출하는 것에 뛰어난 성능을 보인다. 이는 R-CNN보다 1000배 이상 빠르고, Fast R-CNN보다 100배 이상 빠른 처리속도이다. 반면에 YOLO는 영상에서 객체를 찾기 유리하게 설계되어 있기 때문에 단일 이미지에서는 좋은 성능을 발휘하지 못한다.

광학 문자 인식(Optical Character Recognition: OCR)은 본 논문에서 객체로 추출된 라벨을 텍스트로 변환하는 과정에 필요한 기술이다. 현재 OCR에 관해 많은 연구가 이루어지고 있다[2, 4, 6]. OCR은 이미지에서 문자를 스캔하여 기계가 이해할 수 있도록 변환하는 것을 뜻한다. OCR이 잘된 경우 출력 값은 실제 이미지와 같아야 한다.

현재 가장 많이 사용하는 OCR 프로그램은 Google의 Tesseract, Adobe의 Acrobat Pro/DC 등이 있다. OCR은 영상에서의 문자열 추출 일반 이미지에서의 문자열 추출 등에 있어 중요하다. 현재 많이 사용되는 프로그램은 모두 이미지에 초점이 맞춰져 있기 때문에 영상에서 OCR을 하는 분야는 추가적인 연구가 필요하다.

3. 관련 연구

제품의 라벨 인식의 효율성을 증진시키기 위해 다양한 기법을 적용한 연구들이 있다.

대표적으로, [4]는 기존 OCR 기법을 개선하여 제품의 라벨을 인식하는 연구를 진행하였다. 해당 연구에서는 전처리를 통해 모바일 이미지의 인식 정확도를 제고하였다. 그렇지만 해당 연구의 인식 대상은 정적 이미지이며, 실시간 영상에 적용하기에는 한계가 존재한다. 즉, 해당 논문에서 제안한 기법을 기반으로 개발된 응용 프로그램은 이미지를 촬영하고 이를 인식하게 된다. 이로 인해 사용자는 문자 인식을 위해 양질의 이미지가 촬영되기까지 촬영 과정을 반복하게 되어 그 사용이 번거롭다.

[7]은 제품의 라벨 인식을 위해 실시간 객체 추적을 사용하고, 추적된 라벨의 정보를OCR기법을 이용해 추적하는 기본적인 형태의 기법을 제안하였다. 하지만 해당 방법론만으로는 OCR에 소요되는 시간으로 인해 제한된 하드웨어로는 실시간 라벨 정보 추출이 현실적으로 불가능하다는 제약사항이 존재한다. 실제로, OCR을 실시간으로 수행하기 위한 기법이 제안된 [6]에서도 표준 PC환경의 OCR실험에서 평균 0.3초가 소요되었다. 따라서, 본 연구의 목적이 모바일 어플리케이션을 통한 실시간 표시 정보 추출임을 감안할 때, 단순히 제품 라벨의 모든 문자를 인식하는 것은 실시간 처리에 문제가 됨을 알 수 있다.

이를 해결하기 위한 방법으로 [8]에서는 제품 라벨의 실시간 인식 시 OCR 처리 시간을 최소화하는 방법을 제안한다. 해당 논문에서는 단단계 실시간 객체 추적을 적용하여 첫번째 단계로 제품 라벨을 인식하고, 두번째 단계로는 제품 라벨에서 성분명에 해당하는 객

체를 인식한다. 즉, OCR기법을 통해 라벨 전체의 문자를 인식하는 것이 아닌, 성분 함량에 해당하는 문자만을 인식하도록 인식 대상 범위를 최소화하였다.

하지만, [7]과 [8]에서는 제품 표시 정보의 실시간 인식을 위한 방법론을 제시했을 뿐 실제 응용에 대한 설계 및 구현 방법에 대한 구체적 내용은 서술되어 있지 않다.

따라서 본 논문에서는 [8]에서 제안된 기법이 실제로 모바일 환경에서 실시간으로 작동할 수 있도록 실제 구현 가능한 모바일 어플리케이션을 설계하고, 프로토타입을 구현하여 제안된 방법론이 유효함을 검증한다.

4. 다단계 실시간 표시 정보 인식

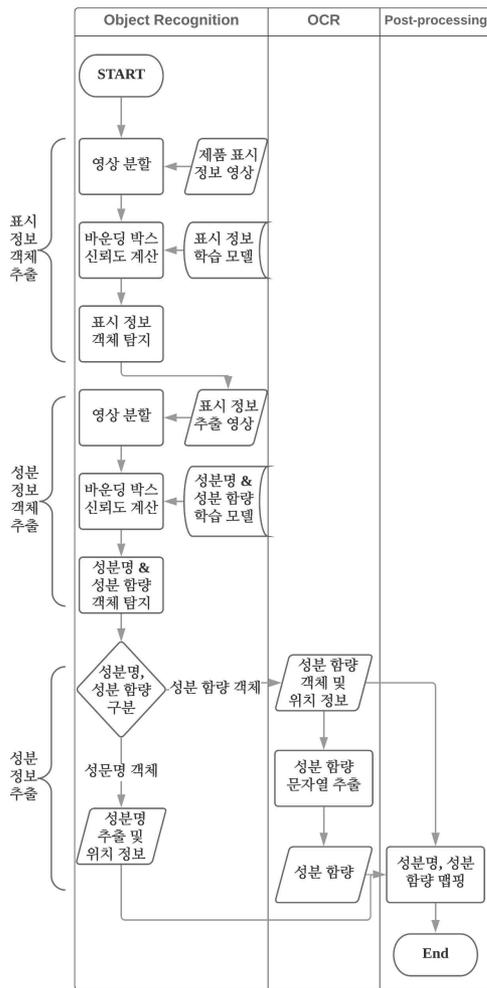


그림 1. 다단계 실시간 표시 정보 인식 개요
Figure 1. An Overview of Multi-Staged Real-Time Label Recognition

4.1 개요

본 연구에서 제안하는 다단계 객체 추적을 통한 표

시 정보 인식 기법의 개요는 그림 1과 같다. 이 기법은 크게 3단계로 나뉜다. 첫 단계는 표시 정보 객체 추출 단계로써, 촬영된 영상에서 표시 정보 부분만을 추출하는 역할을 맡는다. 두번째 단계는 성분 정보 객체 추출 단계로, 추출된 표시 정보에서 성분명 객체와 성분 함량 객체를 추출하게 된다. 마지막 단계인 성분 정보 추출 단계는 OCR을 이용하여 성분 정보 객체에서 정확한 정보를 추출한다.

4.2 표시 정보 객체 추출

첫 단계인 표시 정보 객체 추출 단계는 제품을 근접 촬영한 제품 표시 정보 영상을 입력 값으로 받아, 표시 정보 영역만을 추출한 영상을 결과물로 생성한다.

이를 위한 자세한 과정으로는, 먼저 입력 값인 제품 표시 정보 영상을 행렬의 형태로 분할하여 각 영역을 바운딩 박스로 구분하게 된다. 그 후, YOLO의 합성곱 신경망과 데이터셋을 통해 사전에 학습된 표시 정보 학습 모델을 이용하여 각 바운딩 박스가 어떤 객체로 분류되는지와 그 신뢰도를 계산한다. 그 다음 단계로, 분할된 바운딩 박스들의 각 신뢰도를 고려하여 최종 바운딩 박스를 결정한다. 마지막으로, 결정된 바운딩 박스의 위치 값을 통해 제품 표시 정보 영역만을 추출하게 된다.

4.3 성분 정보 객체 추출

두번째 단계인 성분 정보 객체 추출 단계는 제품 표시 정보 영역을 대상으로 객체 추적을 통해 성분명과 그에 따른 성분 함량 정보를 추출하는 단계이다. 이를 위해, 이전 단계인 표시 정보 추출 단계와 동일하게 입력 받은 표시 정보 추출 영상을 분할한다. 해당 단계에서는 성분명과 성분 함량 객체 학습 모델을 사용하여 각 바운딩 박스의 분류 신뢰도를 판단하게 된다. 그 후, 최종 바운딩 박스를 통해 각 성분명과 성분 함량 객체들을 인식하여 결정한다.

4.4 성분 정보 추출

성분 정보 추출 단계는 최종 단계로써, 정확한 성분 정보 추출을 위해 이전 단계에서 탐지된 객체를 성분명과 성분 함량으로 구분한다. 그 후, OCR처리를 통해 성분 함량 객체에 표현된 성분의 함량 숫자를 인식한다. 최종적으로, 인식된 숫자는 추출된 성분명과 맵핑되어 제품의 표시 정보 인식 과정을 마무리 짓게 된다. 성분명과 성분 함량의 맵핑은 객체 추출시 추적된 위치 정보를 활용하여 진행된다.

5. 사례 연구

제품의 표시 정보가 포함된 영상은 그림 2와 같다. 5장에서는 해당 영상을 대상으로 필요한 정보를 추출

하는 사례 연구를 설명한다. 해당 영상으로부터 정보를 추출하는 과정에 앞서, YOLO의 객체 인식을 위해 Darkflow를 통해 기계 학습 모델은 학습된 상태이다.



그림 2. 제품 표시 정보가 포함된 정적 영상

Figure 2. A Static Image with Product Information

5.1 표시 정보 객체 추출

학습 모델을 이용한 객체 추적을 위해 먼저 이미지를 그리드 형태로 분할한다. 그 후, 각 영역에 해당하는 세부 바운딩 박스의 신뢰도 계산을 진행한다. 세부 바운딩 박스들의 신뢰도를 기반으로 최종 바운딩 박스의 신뢰도를 계산하고 이를 통해 가장 가능성 높은 객체를 인식하여 표현한다. 그 후, 해당 바운딩 박스에 해당하는 이미지만을 잘라내면, 그림 3과 같이 제품의 라벨 부분을 할 수 있다.

5.2 성분 정보 객체 추출

해당 단계에서는 5.1절에서 추출한 라벨 이미지를 대상으로 객체 인식을 진행한다. 라벨 부분만을 잘라낸 이미지는 그림 4와 같다. 성분명과 성분 함량 데이터로 학습시킨 학습 모델을 통해 객체 인식을 다시 한번 진행하면, 그림 4의 바운딩 박스와 같이 각각의 객체를 인식할 수 있다. "Sodium", "Cholesterol"와 같은 성분명은 각 성분명에 해당하는 객체로 인식되었으며, "%"가 포함되는 성분 함량의 경우에는 공통적으로 "content"객체로 인식되었음을 확인할 수 있다.



그림 3. 제품 표시 정보 객체 추출 영상
Figure 3. The Extracted Image of Product Information Object

5.3 성분 정보 추출

5.2절에서 인식된 객체들 중 성분 함량 객체의 경우에는 모두 "content"객체로 인식되었기에, 정확한 숫자를 알지 못하는 상태이다. 또한, 어떤 성분이 어느 정도의 함량을 가지는 지에 대한 것도 결정되지 않은 상태이다. 따라서, 본 과정에서는 이 2가지 목표를 가리게 된다.

먼저, "content"객체에 있는 정확한 숫자를 알아내기 위해 OCR기법을 사용한다. 본 연구에서는 모바일 어플리케이션에서도 문자 인식이 빠른 속도로 동작할 수 있도록 Tesseract를 사용하였다.



그림 4. 성분명 및 성분 함량 객체 인식 결과
Figure 4. The Recognition Result of Component Name and Component Content Object

그 다음 단계로, 5.2절의 성분명, 성분함량 객체들의 위치정보를 가져와 y좌표에 따라 성분명과 OCR을 통해 인식된 문자열을 맵핑하게 된다. 그림 4의 경우, 각각 “Sodium”과 “7%”, “Cholesterol”와 “0%” 등이 맵핑된다.

위의 단계들을 거쳐 최종 결과물로 맵핑된 성분명과 성분함량 문자열을 얻고, 이를 통해 촬영한 제품에 어떤 성분이 어느 만큼의 함량이 들어있는지에 대한 정보를 추출할 수 있다.

6. 모바일 어플리케이션 설계

6장에서 4장에서 제안된 기법을 모바일 어플리케이션으로 구현하기 위해 제품 정보 인식 기능을 가진 모바일 어플리케이션을 설계한다. 그림 5는 사용자가 해당 모바일 어플리케이션을 통해 제품 정보를 인식하는 경우의 시퀀스 다이어그램을 나타낸다.

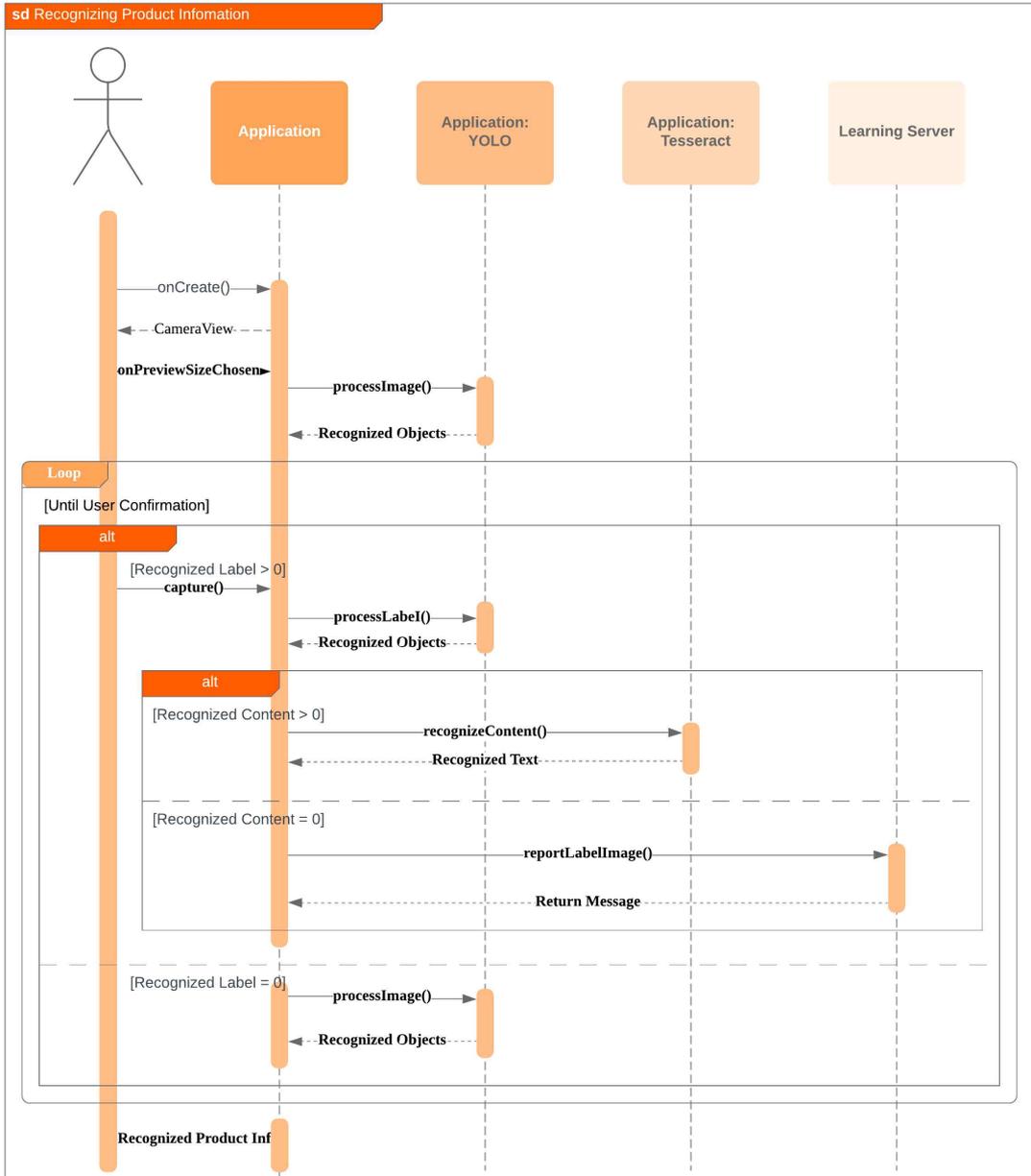


그림 5. 제품 정보 인식 기능의 시퀀스 다이어그램
Figure 5. A Sequence Diagram of the Product Information Recognition Function

먼저, 사용자는 어플리케이션을 시작하게 되고 (onCreate), 카메라의 프리뷰를 확인하게 된다. 어플리케이션은 YOLO 라이브러리를 통해 촬영된 영상을 실시간으로 처리하게 된다 (processImage). 처리된 영상에서 라벨이 인식되어 있고, 사용자가 인식 기능을 작동시키게 되면 (capture), 어플리케이션은 다시 한번 YOLO 라이브러리를 통해 성분명과 성분 함량 객체를 인식한다 (processLabel).

이 과정에서 인식된 성분명, 성분 함량 객체가 있다면, 어플리케이션은 Tesseract 라이브러리를 통해 성분 함량 객체의 문자를 인식한다 (recognizeContent). 혹은 인식된 객체가 없다면, 인식된 제품 라벨을 서버로 전송하여 후후 학습에 활용될 수 있도록 한다 (reportLabelImage).

위 과정은 사용자가 원하는 제품 정보를 얻을 수 있을 때까지 반복되어, 최종적으로 사용자는 인식된 제품 정보를 확인하게 된다.

6. 모바일 어플리케이션 구현

그림 6은 구현된 모바일 어플리케이션의 실제 동작 화면을 나타낸다. 본 화면에서 사용자는 카메라 프리뷰를 통해 촬영된 영상에서 어떤 객체가 인식되는지를 확인할 수 있다. 또한, Content 객체로 인식되어 객체에 해당하는 영상만 잘라낸 결과를 임시적으로 확인할 수 있다.

다만, 그림 6의 각각의 화면은 인식된 객체의 결과가 다르다. 이는 사용자가 모바일을 통해 인식하기에 작은 떨림이 있고, YOLO를 통한 인식이 실시간이기 때문이다. 따라서, 사용자는 YOLO를 통한 객체 추적이 적절히 수행되었을 때, ‘텍스트 인식’ 버튼을 눌러 현재 인식된 정보를 저장하게 된다.

그림 7은 그림 6의 왼쪽 화면의 상태에서 사용자가 정보 저장을 시도하였을 때의 결과화면이다. 인식된 성분명 객체에 해당하는 성분명이 표시되어 있고, 좌표에 맞춰 성분명과 맵핑된 성분 함량이 존재하는 것을 확인할 수 있다.

7. 결론

본 논문에서는 실시간 영상 객체 추적 기법인 YOLO를 이용하여 의약품, 건강 보조 식품 등의 화학품의 표시 정보 객체를 인식하고 인식된 객체를 다시 객체 추적 기법을 이용하여 성분 객체와 함량 객체로 분리한다. 분리된 함량 객체와 성분 객체는 각각 한 세트로 맵핑하고, 함량 객체는 OCR을 이용하여 정확한 수치를 알아내는 기법을 제안한다.

기존의 OCR 방식은 표시 정보를 한번에 변환하는 방법을 채택하였다. 기존의 방식은 다량의 데이터를 한번에 처리하기 때문에 시간이 느릴 뿐만 아니라 정확성도 높지 않다. 이를 해결하기 위해서 다단계 객체 인

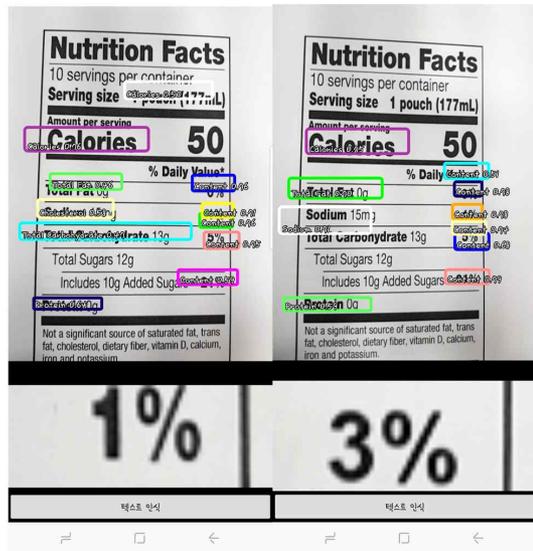


그림 6. 모바일 어플리케이션의 인식 화면 (1, 2)
Figure 6. Recognition Screens of the Mobile Application (1, 2)

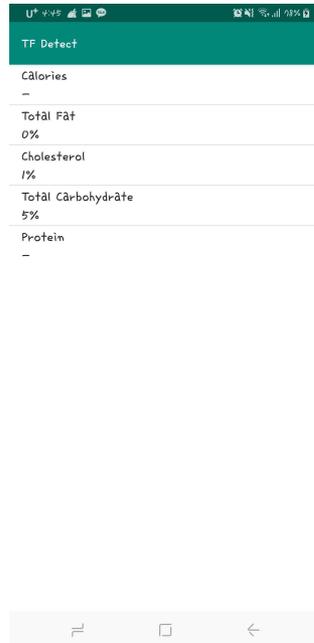


그림 7. 모바일 어플리케이션의 결과 화면
Figure 7. A Result Screen of the Mobile Application

식을 통하여 성분 표시 정보로부터 정보 추출을 하는 과정을 최적화하였다. 또한, 시간을 줄여 실시간으로

정보 추출이 가능하도록 설계하였다.

또한, 일반 사용자들이 편리하게 사용할 수 있도록 모바일 어플리케이션에서 다단계 객체 인식 및 OCR을 수행하는 어플리케이션을 설계하였다.

모바일 어플리케이션을 설계함에 있어 서버-클라이언트 방식의 전송시간 지연을 최소화 하기위해 모바일 기기 내에서 수행할 수 있도록 개발하였다. 또한 모바일의 부족한 연산능력으로도 빠른 연산을 할 수 있도록 가장 오래 걸리는 OCR 작업을 최소화하였다. 향후 연구로는 실험을 통해 기존의 다른 연구와 정량적 비교를 할 연구를 할 것이다.

참 고 문 헌

- [1] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," In Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Jun. 2016
- [2] A. Coates, B. Carpenter, C. Case, S. Satheesh, B. Suresh, T. Wang, D.J. Wu, A.Y. Ng, "Text Detection and Character Recognition in Scene Images with Unsupervised Feature Learning," In Proc. of International Conference on Document Analysis and Recognition (ICDAR), Oct. 2011
- [3] J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement," Technical Report, University of Washington, Apr. 2018, arXiv:1804.02767
- [4] O. Grubert, L. Gao, "Recognition of Nutrition Facts, Labels from Mobile Images," Technical Report, Stanford University, Apr. 2014
- [5] 주미소, "딥 러닝 및 영상 처리 기법을 활용한 실시간 객체 분리 연구," 고려대학교 대학원, 석사학위 논문, 2018
- [6] L. Neumann, J. Matas, "Real-Time Scene Text Localization and Recognition," In Proc. of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012
- [7] 이철훈, 민경식, 이병정, "객체 추적과 OCR 을 통한 표시 정보의 실시간 인식", 한국정보과학회 한국컴퓨터종합학술대회, 2019
- [8] 최지수, 정동주, 민경식, 이병정 "다단계 객체 추적을 통한 표시 정보의 인식 기법," 한국정보처리학회 추계학술대회, 2019



최 지 수

2014년 ~ 현재 서울시립대학교 컴퓨터과학부(학사) 재학중. 관심분야 소프트웨어 공학, 인공지능



이 철 훈

2014년 ~ 현재 서울시립대학교 컴퓨터과학부(학사) 재학중. 관심분야 소프트웨어 공학, 인공지능



정 동 주

2018년 서울시립대학교 컴퓨터과학부(학사). 2018년~현재 서울시립대학교 컴퓨터과학부 석사과정. 관심분야는 소프트웨어테스트, 소프트웨어공학



이 병 정

1990년 서울대학교 계산통계학과(학사). 1990년~1998년 (주)하이닉스반도체 연구원. 1998년 서울대학교 전산학과(석사). 2002년 서울대학교 전기컴퓨터공학부(박사). 2002년~현재 서울시립대 컴퓨터과학부 교수. 관심분야는 소프트웨어테스트, 소프트웨어 진화, 소프트웨어공학



민 경 식

2018년 서울시립대학교 컴퓨터과학부(학사). 2018년~현재 서울시립대학교 컴퓨터과학부 석사과정. 관심분야는 소프트웨어테스트, 소프트웨어공학, 블록체인, 스마트 컨트랙트



논문지 논문 모집 (Call for Papers)



한국정보과학회 소프트웨어공학 소사이어티에서는 매년 4회에 걸쳐 ‘소프트웨어공학 소사이어티 논문지’를 발간하고 있습니다. 이 논문지에는 소프트웨어공학 전반에 걸친 연구 성과 및 산업계 동향을 소개하고 있습니다. 이에 다음과 같은 소프트웨어공학 주제에 관련된 논문을 모집하고 있으니 학계와 산업계의 여러분의 적극적인 논문투고를 바랍니다.

◆ 논문 주제

- 소프트웨어 설계, 아키텍처 및 프로덕트라인
- 요구공학
- 소프트웨어 품질 및 테스트
- 프로젝트 관리 및 프로세스
- 소프트웨어 정형 기법 및 검증
- 임베디드, 모바일, 웹기반 소프트웨어 개발
- 기타 소프트웨어 응용 (국방, 자동차, 의료, 조선, IoT, 빅데이터 등의 분야)

◆ 논문심사

- 투고된 논문은 편집위원회에서 심사 선정하며, 필요 시 외부 심사위원을 위촉하여 심사를 합니다. 제출된 논문은 반환하지 않습니다.
- 심사료 및 게재료: 없음

◆ 논문 제출

- 한국정보과학회 논문지 투고 양식(www.kiise.or.kr)을 사용하며, 논문의 분량은 10장으로 제한합니다.
- 논문지 투고규정에 따라 작성된 심사용 논문파일과 함께 논문제목, 저자, 소속, 초록을 편집위원장 또는 편집위원에게 이메일로 송부하시기 바랍니다.

◆ 문의처 (편집위원회)

- 편집위원장 : 이선아 교수 (경상대학교, 055-772-1377, saleese@gnu.ac.kr)
- 편집이사 : 배경민 교수 (POSTECH 054-279-2256, kmbae@postech.ac.kr)
- 편집이사 : 홍 신 교수 (한동대학교, 054-260-1409, hongshin@handong.edu)



1. 소프트웨어공학소사이어티 논문지에 실리는 원고는 주제 논문, 일반 논문, 산업체 기고 등으로 구분하며 다음과 같은 분야에 대하여 모집한다.
 - 가. 소프트웨어공학 및 그 응용분야에 대한 연구결과
 - 나. 강좌 및 관련 교육사항 소개 (목적, 과정, 일정, 대상, 특징)
 - 다. 소프트웨어 도구 및 방법론 소개 (가격, 특징, 종류, 적용사례)
 - 라. 소프트웨어 산업에 대한 학계, 업계의 주요 관심사
 - 마. 기타 관련 사항
2. 투고자는 원칙적으로 본 소사이어티의 회원으로 한다. 다만 공동 또는 초청 기고자는 예외로 한다.
3. 논문은 원칙적으로 한글로 작성한다.
4. 원고는 한글(hwp), 워드(MS Word), PDF 형식 중 하나를 택하여 작성하며, 그림과 표를 포함하여 10쪽 이내로 한다.
5. 논문 내용에 직접 관련이 있는 문헌에 대해서는 이들 문헌에 관련이 있는 본문 중에 참고 문헌 번호를 쓰고 그 문헌을 참고문헌 난에 인용 순서대로 기술한다. 참고문헌은 학술지의 경우 저자, 제목, 학술지명, 권, 호, 쪽수, 발행 연도의 순서로, 단행본은 저자, 서명, 쪽수, 발행처, 발행 연도의 순서로 기술한다.
6. 작성된 논문은 논문파일과 함께 논문제목, 저자, 소속, 초록을 편집위원장 또는 특집 주제 담당 편집위원에게 이메일로 제출한다.
7. 기타 자세한 사항은 한국정보과학회 논문지 투고 요령을 따른다.



2018-2019 소프트웨어공학소사이어티 논문지 편집위원회

편집위원장 이선아 교수(경상대학교)

편집위원 배경민 교수(POSTECH)
홍신 교수(한동대학교)
이정원 교수(아주대학교)
고인영 교수(KAIST)
김정아 교수(가톨릭관동대학교)
백종문 교수(KAIST)
유준범 교수(건국대학교)
김순태 교수(전북대학교)
한종대 교수(상명대학교)



소프트웨어공학소사이어티 논문지 제28권 제1호 (통권 102호)

발행일 || 2019년 12월 31일

발행인 || 이병정

편집인 || 이선아

발행처 || 사단법인 한국정보과학회 소프트웨어공학소사이어티

연락처 || 서울특별시 동대문구 서울시립대로 163(전농동) 서울시립대학교

정보기술관 202호 컴퓨터과학부 이병정

전화 : 02-6490-2451, 팩스 : 02-6490-2444

홈페이지 : <http://www.sigsoft.or.kr/>

Journal of Software Engineering Society

VOLUME 28, NUMBER 2, December 2019

Design and Implementation of Ethereum Smart Contract State Monitoring System	Joongi Hong Suntae Kim Duksan Ryu	1
Design and Implementation of Mobile Application for Extracting Information of Products based on Real-Time Recognition	Kyeongsic Min Ji-Su Choi Cheolhun Lee Dongju Jung Byungjeong Lee	7

Korean Institute of Information Scientists and Engineers
Software Engineering Society