논문 19-44-01-20 The Journal of Korean Institute of Communications and Information Sciences '19-01 Vol.44 No.01
https://doi.org/10.7840/kics.2019.44.1.137

# 대용량 웹툰 이미지 검색을 위한 디스크 없는 분산 데이터베이스 검색 시스템 구현

김 도 영◆, 박 현 우*, 박 종 화*, 권 혁 주*, 이 상 훈°

# Implementation of a Diskless Distributed Database System for Large-Scale Webtoon Image Fingerprint Data Retrieval

Doyoung Kim◆, Hyunwoo Park*, Jonghwa Park*, Hyuck-Joo Kwon*, Sanghoon Lee°

요 약

온라인 웹툰(Webtoon) 시장의 인기가 증가함에 따라, 웹툰 불법 복제 및 유통의 규모도 증가하고 있다. 불법 복제 및 유통을 방지하기 위한 웹툰 식별 기술은, 이미지에서 추출한 핑거프린트를 데이터베이스에 저장된 대용량 핑거프린트에서 검색하는 방식을 사용한다. 데이터베이스에 저장된 데이터양이 매우 많기 때문에 효율적인 데이터 관리와 고속화된 검색 알고리즘이 필수적이다. 본 논문에서는 검색을 고속화하기 위해, 다수의 CPU를 활용한 diskless 분산 시스템을 설계하였다. 엔트로피 가중치를 적용한 Visual word 클러스터링을 통해 식별 성능을 향상시켰다. DBMS(database management system)의 인덱싱과 질의 최적화를 통해 검색을 고속화하였으며 이를 통해 실험적으로 성능을 평가하였다. 그 결과 실제 대용량 웹툰 이미지 환경에서 적용 가능한 식별 성능을 보여주었다.

**Key Words** : Webtoon, identification, large-scale, Visual word, clustering, diskless, distributed system

ABSTRACT

As the popularity of online Webtoon market grows, the size of illegal copying and distribution of Webtoons is also increasing. Toward the end, Webtoon identification technology is essential to prevent illegal copying and distribution. Among those technologies, the method of searching a fingerprint is highly focused to find the corresponding original image in a large-scale fingerprint stored in a database. Because the amount of data stored in the database is very large, efficient data management and fast search algorithms are essential. In this paper, a diskless distributed system is designed by using multiple CPUs to speed up the search. Visual word clustering with entropy weighting improves the identification performance. The search was speeded up by indexing and query optimization of the DBMS (database management system). The performance was evaluated experimentally. As a result, the identification performance which can be applied in a real-scale Webtoon image environment is proved.

※ 본 연구는 문화체육관광부 및 한국저작권위원회의 2018년도 저작권기술개발사업의 연구결과로 수행되었음.

◆ First Author : (ORCID:0000-0002-8156-9738)Yonsei University, Department of Electrical and Electronic Engineering, tnyffx@yonsei.ac.kr, 학생회원

° Corresponding Author : (ORCID:0000-0001-9895-5347)Yonsei University, Department of Electrical and Electronic Engineering, slee@yonsei.ac.kr, 종신회원

* (ORCID:0000-0002-2007-5267)Yonsei University, Department of Electrical and Electronic Engineering, beobest2@yonsei.ac.kr

* (ORCID:0000-0002-2664-6307)Yonsei University, Department of Electrical and Electronic Engineering, lanian@yonsei.ac.kr

* (ORCID:0000-0002-2619-5534)Yonsei University, Department of Electrical and Electronic Engineering, hyuckjookwon@yonsei.ac.kr

논문번호 : 201806-D-002, Received June 14, 2018; Revised December 20, 2018; Accepted December 20, 2018

## Ⅰ. 서 론

불법 유통되는 웹툰을 단속하기 위해서는 해당 웹툰을 식별하여 저작권 정보를 확인해야 한다. 보통 하루 약 7천 편의 웹툰이 온라인에 업데이트 되며 불법으로 웹툰을 게시하는 웹사이트는 계속 증가하고 있다[1]. 한 편의 웹툰은 수십 회를 거쳐 연재되기 때문에 저작권 단속을 위해 총 식별해야 할 웹툰의 수는 수십만 건이다. 이처럼 불법적으로 업로드 된 웹툰의 규모가 매우 크기 때문에 사람이 일일이 단속하는 것이 쉽지 않은 실정이다. 따라서 웹툰 이미지를 식별하는 자동화된 시스템이 개발되고 있다.

기존에 연구된 웹툰 식별 시스템은 웹툰 이미지를 주파수 영역으로 변환하여 핑거프린트(fingerprint)를 추출하는 방법을 사용한다[12]. 추출된 핑거프린트는 128차원 벡터로 구성된다. 입력 웹툰 이미지에서 추출된 핑거프린트는 데이터베이스에 저장된 대용량 핑거프린트와 유사도 연산을 수행하여 가장 유사도가 큰 핑거프린트의 메타 정보(meta data)를 출력한다.

데이터베이스에 저장된 핑거프린트를 모두 검색하는 방식은 CPU에 상당히 많은 부하를 초래한다. 수십만 건의 불법 게시 웹툰을 식별하는 실제 서비스를 위해서는 CPU에 부하를 최소화하고 식별 속도는 5초 이내, 정확도는 90% 이상의 높은 성능을 만족 시켜야 한다. 이러한 조건을 만족시키기 위해서, 대용량 데이터를 효과적으로 관리하고 처리하는 시스템 설계가 필수적이다. 데이터를 독립된 형태로 분류하고 병렬적으로 처리하는 방식으로 효율성을 향상 시킬 수 있다. diskless 기반의 분산 시스템은 이러한 데이터 처리에 적합하다. 먼저 diskless 시스템은 모든 데이터를 서버에서 관리한다. 매번 서버로부터 데이터를 불러와 클라이언트에서 실행하기 때문에, 클라이언트의 운영체제에 문제가 발생하더라도 서버에서 다시 데이터를 불러오면 쉽게 해결할 수 있다. 서버에서 데이터 백업만 잘 이루어진다면 가장 고장이 빈번한 저장 장치만 교체하면 되기 때문에, 하드웨어 유지 및 보수도 용이하다. 또한, 클라이언트 노드를 추가하는 것이 매우 용이하다. 저장 장치를 제외한 나머지 장치가 클라이언트에게 주어진다면, 서버에서 데이터를 불러와서 실행할 수 있다. 따라서, 네트워크로 연결만 가능하다면 많은 양의 클라이언트 노드를 통해 대용량 데이터를 처리하는 것이 가능하다. 이러한 하드웨어 상의 이점을 바탕으로 클러스터링(clustering)은 초기 검색 연산을 감소시켜 효율적인 대용량 데이터 검색이 가능하도록 해준다. 클러스터링을 통해 줄어든 검색 연산을 여러 대의 CPU에 나누어 동시에 연산을 수행하면 보다 더욱 고속화된 처리 속도를 기대할 수 있다.

본 논문에서는 웹툰 이미지 식별을 위한 diskless 분산 데이터베이스 검색 시스템을 구현하였다. Visual word 클러스터링을 통한 데이터 분류와 인덱싱(indexing) 및 질의(query) 최적화를 통해 데이터 검색을 고속화하였다. 그 결과 실제 대용량 웹툰 이미지 환경에 적용 가능한 수준의 식별 성능을 확인할 수 있었다.

논문의 전체적인 구성은 다음과 같다. Ⅱ에서는 분산 데이터베이스 시스템 구현을 단계적으로 설명한다. Ⅲ에서는 제안한 시스템의 성능 평가가 이루어진다. 마지막으로 Ⅳ에서는 결론 및 향후 연구를 제시한다.

## Ⅱ. 대용량 웹툰 이미지 식별을 위한 분산 데이터베이스 검색 시스템

### 2.1 전체 시스템 구조

제안하는 시스템은 그림 1과 같이, 웹툰 이미지 핑거프린트 추출부, 클래스 결정부, 분산 CPU를 사용한 핑거프린트 검색부를 특징으로 한다.

서버단 에서는, 온라인상에 존재하는 웹툰 데이터베이스를 회 차별로 분류하여(ID) 데이터베이스에 저장한다. 웹툰 프레임(frame)을 클래스로 분류하고 핑거프린트를 추출한다. 이렇게 생성된 핑거프린트와 클래스 정보가 데이터베이스에 저장된다.

클라이언트단에서는, 식별을 요청하는 질의 이미지가 입력되면 서버에서 처리한 방식과 동일하게 웹툰 이미지의 클래스를 결정하고 핑거프린트를 추출한다.

분산 핑거프린트 검색 단계에서는, 데이터베이스내의 핑거프린트를 각 클라이언트 노드에서 병렬적으로 나누어 유사도 연산을 수행한다. 유사도 연산에는 유클리디안 거리(euclidean distance)를 사용하여 가장

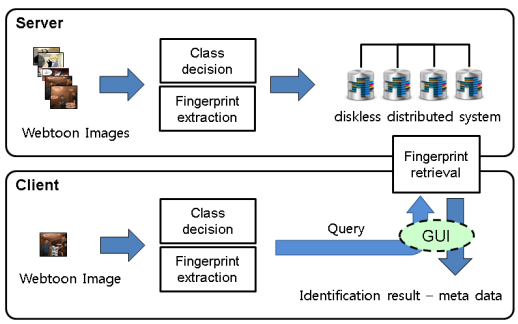



그림 1. 전체 시스템 구조
Fig. 1. Overall system structure

거리가 작은 벡터를 선택한다. 각 클라이언트 노드에서 출력된 유사도 정보는 서버 노드로 수집되어 최종 취합된 식별 결과를 출력한다. 사용자는 GUI 환경을 통해 최종 출력된 결과를 확인한다.

이미지에서 핑거프린트를 추출하고, 클래스를 결정하는 부분은 한 프레임 당 0.002초 이내에 모두 처리된다. 그러나 생성된 핑거프린트를 서버 데이터베이스와 유사도 연산하고 취합하는 과정에서 대용량 데이터를 다루기 때문에 전체 시스템 속도 저하의 원인이 된다. 따라서 병목구간이 되는 대용량 데이터베이스 검색을 분산 시스템으로 구성하였다.

### 2.2 Visual word 클러스터링

데이터베이스 시스템의 SQL 질의 최적화, 인덱싱 등의 구조개선 만으로는, 점차 늘어가는 대용량 데이터에 핑거프린트 검색 시간을 감소시키는데 한계가 있다. 효율적인 데이터 검색을 위해, 데이터 콘텐츠 자체에서 의미를 추출하여 클러스터링 함으로써 불필요한 검색 영역을 제거하는 데이터 마이닝 기술의 적용이 필수적이다[2].

저작물 고속 식별을 위해서 웹툰 프레임을 동일 클래스끼리 묶어주는 방법을 사용한다. 클래스 정보가 핑거프린트 정보와 함께 데이터베이스에 저장된다. 검색할 때 해당 클래스에 해당하는 핑거프린트만 검색하면 되므로 검색 효율이 매우 높아진다. 또한, 매칭 연산 시 유사도가 떨어지는 이미지를 검색에서 제외하고 식별하기 때문에 식별 정확도가 향상된다[3].

추출한 특징점의 발생 빈도를 사용하여 웹툰 이미지를 특정 클래스로 분류 할 수 있다[4]. 이미지의 특징점은 주로 SIFT(Scale Invariant Feature Transform)나 SURF(Speeded Up Robust Features) 특징점을 사용한다. SURF 특징점은 이미지 내에서 부분적으로 불변한 특징 영역을 나타낸다[5]. 각각의 특징점은 방향, 좌표정보, 라플라시안, 헤이시안, 사이즈 등의 정보를 가지고 있다. 이 정보들 중에 이미지의 위치 관계와 무관한, 방향, 라플라시안, 사이즈 3차원의 데이터를 사용하였다.

클래스 결정부는 트레이닝부와 테스팅부로 구분된다. 그림 2와 같이 트레이닝부 에서는 대용량의 트레이닝용 이미지 데이터셋을 가지고 이미지의 SURF 특징점을 클러스터링하여 DB로 구축한다. 본 논문에서는 K-means 알고리즘을 사용하였다[6][7]. 각 클러스터들의 대표 값(centroid)은 초기 설정한 클러스터 개수 K만큼 생성된다. 이렇게 생성된 클러스터들의 centroid를 시각 단어(Visual word)라고 한다.

테스팅부는 클래스를 결정하는 단계이다. 입력 질의 이미지에서 SURF 특징점을 추출한다. 추출된 각각의 특징점 벡터를 클러스터 대표 값인 Visual word 벡터와 유클리디안 거리를 계산하여 가장 가까운 Visual word로 해당 특징점을 양자화(quantization)한다. 양자화된 Visual word의 히스토그램을 계산하여 클래스를 결정한다.

히스토그램 계산을 통해 대표 Visual word가 클래스로 결정된다. 따라서 총 클래스 수는 클러스터링 된 Visual word의 개수와 동일하다. 같은 ID의 웹툰 이미지라도 프레임에 따라 콘텐츠 특성이 다르므로 다른 클래스 정보를 가질 수 있다.

기존 클래스 결정 방식은 그림 3과 같이 K개의 클러스터 centroid인 Visual word와 모두 유사도 연산을 하는 방식을 사용하기 때문에 클러스터 수가 많은 경우 연산 비용이 크다. 또한 Visual word 히스토그램 계산 후에 가장 높은 빈도의 Visual word를 대표 클래스로 설정하는 방식을 사용하였는데, 이는 밀도가 높은 특정 클러스터에 Visual word가 집중될 가능성이 있기 때문에 검색 효율이 떨어지게 된다.

본 논문에서는 위계 트리(hierarchical tree)구조 클러스터링 방식을 사용하여 Visual word 양자화를 고속화 했다[8,9]. 그림 4와 같이 클러스터 centroid를 위계적으로 구성하여 Visual word 트리를 생성한다. 기

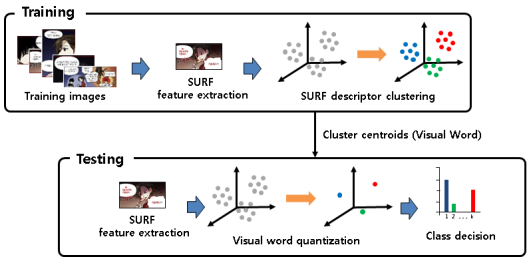



그림 2. 웹툰 이미지 클래스 결정
Fig. 2. Webtoon image class decision

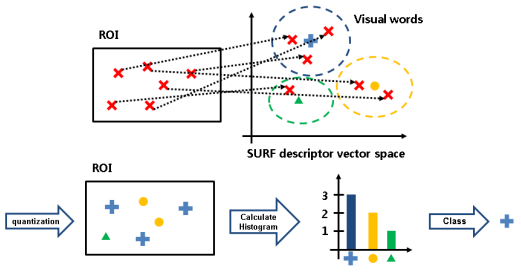



그림 3. 기존 클래스 결정 방식
Fig. 3. Conventional class decision method

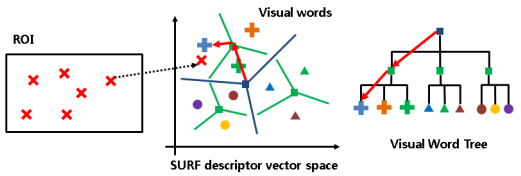



그림 4. 위계 트리 구조 클러스터링
Fig. 4. Hierarchical tree clustering

존에는 SURF 디스크립터(descriptor)를 양자화하기 위해 모든 K개의 centroid와 유사도 연산을 했던 반면, Visual word 트리를 사용하면 부모 노드에서 자식 노드의 일부 centroid만 검색하고 가장 유사도가 큰 자식 노드의 자식 노드만 검색하므로 알고리즘의 효율성이 O(N)에서 O(logN)으로 향상된다.

### 2.3 엔트로피 가중치

Visual word의 특정 클러스터에 밀도가 집중되는 경우 Visual word에 엔트로피 기여도를 반영하여 클래스를 균일하게 분류할 수 있다. 이미지에서 보편적으로 발생하는 Visual word에는 낮은 가중치를 주고 드물게 발생하는 Visual word에는 높은 가중치를 주는 방식을 사용한다.

$$w_k = \ln \frac{M}{M_k} \tag{1}$$

$$n_k = N_k w_k \tag{2}$$

k번째 Visual word의 발생 빈도를 $N_k$라고 할 때, 기존 방식은 $N_k$만 사용해서 클래스를 결정하였다. $w_k$는 엔트로피 가중치이다. 식 1과 같이 $w_k$는 트레이닝 이미지의 전체 개수인 M에 비례하고, k번째 Visual word에 한 번이라도 도달한 이미지 수인 $M_k$에 반비례한다. 식 2와 같이 $N_k$에 가중치 $w_k$를 곱해주어, 가중치로 조정된 히스토그램의 카운트값 $n_k$를 결정한다. 보편적으로 발생하는 Visual word의 경우 $M_k$가 매우 크기 때문에 곱해지는 $w_k$가 작아진다. 드물게 발생하는 Visual word는 $M_k$가 작기 때문에 큰 가중치 $w_k$를 갖게 되어 큰 영향력을 나타내게 된다.

그림 5의 예시와 같이 보편적으로 많이 발생하는 Visual word(파란색 십자 모양)이 존재한다. 가중치를 적용하기 전에 $N_k$만으로 계산한 히스토그램을 살펴보면, 특정 Visual word의 영향이 크게 나타나므로

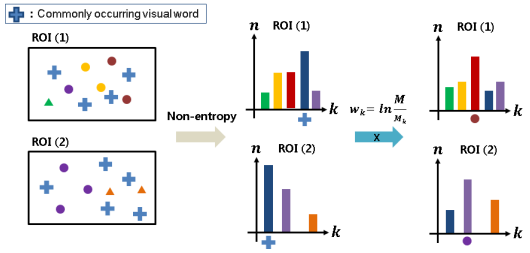



그림 5. 엔트로피 가중치를 적용한 클래스 결정
Fig. 5. Determining the class to which the entropy weight is applied

(1)과 (2)가 하나의 클래스로 결정되는 쏠림 현상이 발생한다. 엔트로피 가중치를 적용한 히스토그램을 살펴보면, 보편적으로 많이 발생하는 Visual word의 영향력이 감소하였음을 알 수 있다. 또한 드물게 발생하는 Visual word에 큰 가중치가 곱해지므로 두 이미지가 다른 클래스로 최종 분류됨을 확인할 수 있다. 클래스가 균일하게 분류되어 검색속도가 클래스에 상관없이 동일하게 향상되며, 특정 클래스에서 오식별이 증가하는 문제가 개선되었다.

### 2.4 SQL 최적화

생성된 핑거프린트는 DBMS(DataBase Management System)을 사용해서 저장한다. DBMS는 SQL (Structured Query Language)을 사용해서데이터를 효율적으로 질의하고 수정할 수 있도록 하며, 허락되지 않은 사용자로부터 정보를 보호하는 보안 기능을 가진 시스템이다. 중복을 최소화하여 데이터를 관리할 수 있으며, 어플리케이션 프로그램과 독립적이기 때문에 손쉽게 시스템을 변경할 수 있다. DBMS중에 시스템과 연동이 손쉬운 정형 데이터베이스인 MySQL을 사용하여 데이터베이스 서버를 구축하였다.

서버 DB 테이블을 S, 클라이언트 DB 테이블을 C이며 유사도 계산 함수를 Function_distance()라 할 때, 웹툰 식별에 사용하는 질의문은 다음과 같다.

"SELECT Function_distance() FROM C, S
WHERE C.class = S. class;"

여러 단계의 nested 질의문 없이 간단한 검색 질의만으로 저작물의 식별이 가능하다. WHERE문에서 클래스를 우선 검색하여 다른 클래스의 핑거프린트는 유사도 연산 과정에서 제외되므로 식별 효율성이 향상 되었다.

### 2.5 핑거프린트 인덱싱(Indexing)

질의를 효율적으로 지원하기 위해서 DBMS의 인덱스 구조를 적절히 사용한다. 데이터 베이스의 특정 값을 매칭하기 위해, 모든 데이터 값을 순차적으로 검색하면 데이터 수가 늘어날수록 검색 시간이 크게 증가할 것이다[10].

인덱스는 [“탐색 키,” “레코드에 대한 포인터”]로 구성된다. 검색하려는 키를 인덱스를 통해서 찾으면, 저장된 정보를 포인터로 바로 연결시켜준다. 인덱스 생성 시, 탐색 키 값이 오름차순으로 정렬되어 저장되기 때문에, 이진 탐색을 통해 디스크 접근 횟수가 줄어들어 응답 시간이 크게 단축 된다. 본 시스템에서는 “클래스” 값에 인덱스를 생성해 두어, 데이터베이스 검색 시간을 감소시켰다.

### 2.6 Diskless 분산 시스템 구조

diskless 시스템은 분산 가상 데스크탑 환경의 일종으로, 서버와 클라이언트의 구조로 구성되어 있다. 일반적인 시스템은 저장 장치에 있는 운영체제의 부트로더를 통해 부팅된다. diskless 구조의 핵심은 클라이언트인 개별 노드 컴퓨터의 부팅과정에서 로컬 저장장치 대신 서버의 가상 원격 저장장치를 사용한다는 점이다. 운영체제가 부팅 완료될 경우, 노드들은 각자 소유한 가상의 원격 저장장치를 마치 자신의 로컬 저장장치 인 것처럼 사용할 수 있다. 이러한 각 개별 노드들의 가상 저장장치는 diskless 서버에 일종의 이미지 형식으로 저장되어 있다. 따라서 노드들은 상호간에 파일이나 데이터의 추가, 수정, 삭제에 있어 제한 없이 각자 소유한 저장 공간을 자유롭게 이용할 수 있다. 그림 6은 diskless 시스템의 구조를 보여준다. diskless의 서버에서 구성되어야 하는 기능들은 동적 호스트 구성 프로토콜(Dynamic Host Configuration Protocol, DHCP) 서버, TFTP(Trivial File Transfer Protocol) 서버, iSCSI(internet Small Computer System Interfaces) 서버가 필요하다.

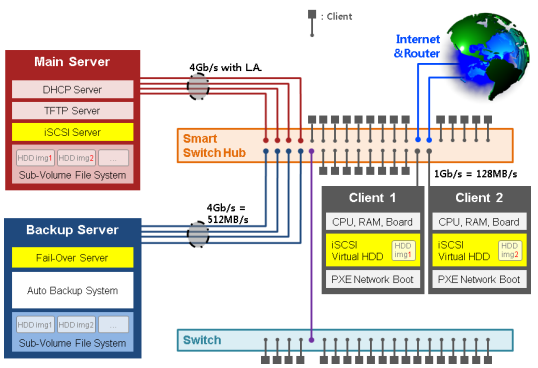



그림 6. Diskless 시스템 구조
Fig. 6. Diskless system structure

diskless의 클라이언트인 노드에서 구성되어야 하는 기능들은 iSCSI 클라이언트, PXE(Preboot eXecution Environment) 펌웨어 및 네트워크 부팅 환경이다. 이러한 구성이 서로 유기적으로 연동되어 프로토콜의 송수신이 가능하게 되면, 노드에서 서버에 할당된 가상의 이미지를 자신의 로컬 저장장치 인 것처럼 인식하여 결과적으로 로컬 저장장치 없이 운영체제의 부팅이 가능하며, 또한 마치 자신의 로컬 저장공간처럼 자유롭게 활용할 수 있게 된다.

#### 2.6.1 사전 부팅 환경(Pre eXecution Environment) 및 펌웨어

사전 부팅 환경은 노드 컴퓨터의 바이오스 단에서 로컬 저장장치 없이 포스트(post) 이후의 과정을 수행할 수 있도록 iSCSI 프로토콜, TFTP 프로토콜, DHCP 프로토콜 등을 포함시켜놓은 일종의 펌웨어이다. 바이오스에서 LAN(Local Area Network)를 통한 부팅을 활성화시키면, 포스트 이후의 부팅 과정은 자연스럽게 PXE 펌웨어가 넘겨받음으로써 필요한 최소한의 드라이버와 프로토콜들을 로드한 후 부팅과정을 수행하게 된다. 본 논문에서는 UNDI(Universal Network Device Interface)의 펌웨어를 필요한 프로토콜과 드라이버를 넣어서 빌드하여 사용함으로써, NIC(Network Interface Card)의 종류나 드라이버에 상관없이 표준적으로 호환되는 드라이버를 통해 노드의 하드웨어 환경에 제한받지 않는 사전 부팅 환경을 조성하였다. 노드 컴퓨터가 부팅과정을 수행할 시, 포스트 과정 이후에는 diskless 서버의 DHCP 서버로 접근을 수행, 응답을 기다린 후 PXE펌웨어를 다운받아서 다음 부팅 과정을 진행하게 된다.

#### 2.6.2 동적 호스트 구성 프로토콜(DHCP) 서버

동적 호스트 구성 프로토콜은 각 개별 노드들이 클라이언트로써 서버에 접근하여 IP(Internet Protocol) 주소를 응답받아 자신의 IP 주소로써 사용할 수 있게 통신하는 프로토콜이다. diskless 서버는 DHCP 서버로써 IP 주소들의 풀과 클라이언트 설정 파라메터를 관리한다. 노드 컴퓨터로부터 요청을 받으면 서버는 노드 컴퓨터의 MAC(Media Access Control) 주소를 소유하고 있는 IP-MAC 매칭 테이블에서 검색하여 적절한 IP 주소를 대여 기간과 함께 응답한다. 이 후 노드 컴퓨터는 받은 IP 주소를 통해 네트워크에 접속하

게 되고 이후 서버에 ACK(acknowledgement)를 전송하여 요청된 설정을 확인 응답하게 된다. DHCP 서버가 ACK를 수신하면, TFTP 프로토콜을 이용하여 PXE 펌웨어를 해당 노드에게 전송하여 다음 부팅과정에 대한 권한을 넘겨주게 된다. 이후 PXE 펌웨어는 포함된 드라이버와 프로토콜들을 이용하여 부팅을 수행하게 되는데, 본 diskless 구조에서는 iSCSI 프로토콜을 이용하여 서버로부터 노드 자신의 가상 저장장치를 마운트하는 과정이 일어나게 된다.

#### 2.6.3 iSCSI(internet Small Computer System Interfaces) 서버

iSCSI 프로토콜은 네트워킹 환경에서 데이터 스토리지 시설을 이어주는 IP 기반의 스토리지 네트워킹 표준으로, 실제 노드 컴퓨터가 로컬 저장장치를 이용할 때 사용되는 하드웨어 명령어 셋을 네트워크 프로토콜 수준으로 가상화를 시켜놓은 프로토콜이다. iSCSI 프로토콜은 SCSI 하드웨어 명령어 셋을 IP망을 통해 전달함으로써 네트워크를 통해 데이터 전송을 쉽게 하고 원거리에서 스토리지를 접근하거나 관리하는데 사용된다. 따라서 iSCSI 서버 - 클라이언트로 연결이 성공적으로 된다면, 마치 노드 컴퓨터는 자신에게 설치된 로컬 저장소의 SCSI 하드웨어 명령어를 사용하는 것과 같아 실제 로컬 저장장치 인 것처럼 동작하게 된다. 노드 컴퓨터에서 PXE 펌웨어가 iSCSI 서버에 접근을 시도한 후, 연결이 성공적으로 생성되면 이후 부팅 과정은 iSCSI 서버의 가상 저장 공간에 설치되어 있는 운영체제의 부트섹터를 읽은 후, 부트로더 프로그램이 실행되어 운영체제의 부팅을 마무리하게 된다.

### 2.7 분산 검색 시스템 플로우

대용량 웹툰 핑거프린트 데이터를 효율적으로 저장 및 관리하기 위해, diskless시스템을 활용하여 데이터를 독립적으로 나누고 병렬 처리하는 시스템 플로우를 설계하였다.

그림 7과 같이 이미지 핑거프린트 데이터를 여러 CPU에 분산하여 병렬적으로 매칭하는 시스템을 제안한다. 제안하는 분산 데이터베이스 시스템은, 물리적 저장 공간을 공유하며 다수의 프로세서를 이용하여 동시에 병렬적으로 매칭 연산을 수행하는 시스템이다. 각 클라이언트 노드에서 독립적으로 연산기능을 수행하기 때문에 다양한 알고리즘에 확장이 용이하다는 장점이 있다.

diskless 시스템은 클라이언트 노드에 저장 공간이

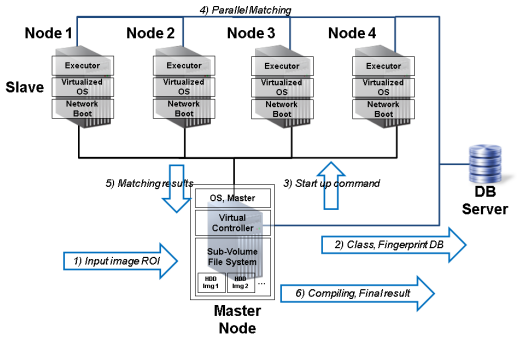



그림 7. 분산 검색 시스템 플로우
Fig. 7. Distributed retrieval system flow

없이 서버 노드에서 가상 컨트롤러를 통해 OS 이미지를 관리하며, 물리적 하드디스크(HDD)를 공유한다. 따라서 전체 서버 DB를 한 공간에 저장해두고, 각 노드에서 병렬적으로 검색을 수행한다.

분산 검색 및 매칭 시스템의 동작 단계는 다음과 같다.

1) 모든 클라이언트 노드는 대기상태로 서버 노드의 명령을 기다린다. 질의 이미지가 들어오면 Visual word 클러스터링을 통해 클래스를 결정하고, 주파수 영역 핑거프린트를 추출한다.
2) 클래스와 핑거프린트 정보를 공용 DB의 데이터베이스의 클라이언트 테이블에 저장한다.
3) 입력 질의 이미지에서 클래스 결정과 핑거프린트 추출이 완료되면 서버 노드에서 각 클라이언트 노드에 TCP(Transmission Control Protocol)를 사용하여 업무 수행 명령을 내린다.
4) 업무 수행 명령을 받은 각 클라이언트 노드는 SQL 문을 실행하여, 각 노드에 분산되어 할당된 데이터베이스의 핑거프린트 벡터와 입력 핑거프린트 벡터와의 유사도 연산을 수행한다.
5) 각 노드에서 TCP 프로토콜을 사용해서 가장 유사도가 높은 웹툰 저작물 ID와 유클리디안 거리를 서버 노드에 전송한다.
6) 서버 노드에서는 모든 클라이언트 노드에서 수신한 결과정보를 취합해서 가장 유사도가 높은 웹툰 ID를 출력한다.

만일 각각의 저장 공간을 분담하여 관리하는 분산처리 시스템이라면, 대량의 데이터를 노드 간에 전송하거나 불필요한 복제 작업이 요구된다. 그러나 본 diskless 시스템에서는 HDD를 공유하며 각 노드에서는 연산을 맡아서 처리하기 때문에 노드 간에 대량의

데이터가 전송하는 과정이 없다. 서버와 클라이언트 노드 간에는 작업의 시작 명령과 최종 결과 값만 전송되기 때문에 노드 간 통신에 사용되는 비용이 매우 적으므로 효율적이다.

### 2.8 GUI(Graphic User Interface) 구현

사용자는 GUI를 통해 보다 편리하게 식별 결과를 확인할 수 있다. 사용자가 입력한 이미지에서 추출된 프레임을 디스플레이 하고, 각 프레임에 따른 최종 식별 결과와 후보군 20개의 유사를 그래프로 확인할 수 있는 웹 GUI를 설계하였다. 웹에서 사용자가 확인할 수 있는 정보는 데이터베이스에서 참조한다. 데이터베이스에 서버노드에서 최종 취합한 정보가 저장될 결과(result) 테이블을 생성하였다. result 테이블은 추출된 이미지가 저장된 경로, 후보군 20개 핑거프린트 벡터와 입력 벡터의 유클리디안 거리와 웹툰 ID, 최종 식별 결과 여부로 구성된다.

추출된 프레임을 화면에 보여주기 위해 데이터베이스에 저장된 디렉토리 경로를 참조한다. 웹 브라우저가 이미지를 사용자에게 보여주기 위해서는, 웹페이지 HTML(HyperText Markup Language) 태그의 이미지(img) 속성을 사용한다. img 속성에서 이미지의 경로를 입력하는 부분에 데이터베이스의 이미지 경로를 매핑해 준다.

그림 8과 같이 최종 식별 결과는 20개의 후보군을 바이너리로 데이터베이스에 저장한다. Top column에는 식별된 웹툰 ID에 1이 저장되며, 나머지 19개의 핑거프린트 벡터는 모두 0이 저장된다. 웹서버에서 top column이 1인 row의 웹툰 ID를 선택한다. 해당하는


그림 9. 웹툰 식별 결과 GUI
Fig. 9. GUI of Webtoon identification result

웹툰 ID의 메타 정보를 출력하기 위해 웹툰 ID와 메타 정보가 저장된 테이블과 LEFT JOIN 질의를 수행하여, 식별된 웹툰의 이름과 회 차 정보를 웹브라우저에 함께 표시한다.

후보군 20개의 유사도 거리를 그래프로 나타내기 위해 HTML의 캔버스(canvas) 속성을 사용하여 직사각형을 순서대로 그려준다. 직사각형 하나의 너비는 전제 canvas할당 영역의 1/20로 설정한다. 직사각형의 높이는 데이터베이스에서 유클리디안 거리를 참조하여 그려준다. 유클리디안 거리에 조건을 설정하여 일정 유클리디안 거리 이하의 직사각형의 색상을 다르게 표시하는 인덱싱 시각화를 적용하였다. 각 직사각형에 onClick() 함수를 적용하여, 사용자가 마우스로 해당 직사각형을 클릭할 때, 후보 webtoon ID가 팝업

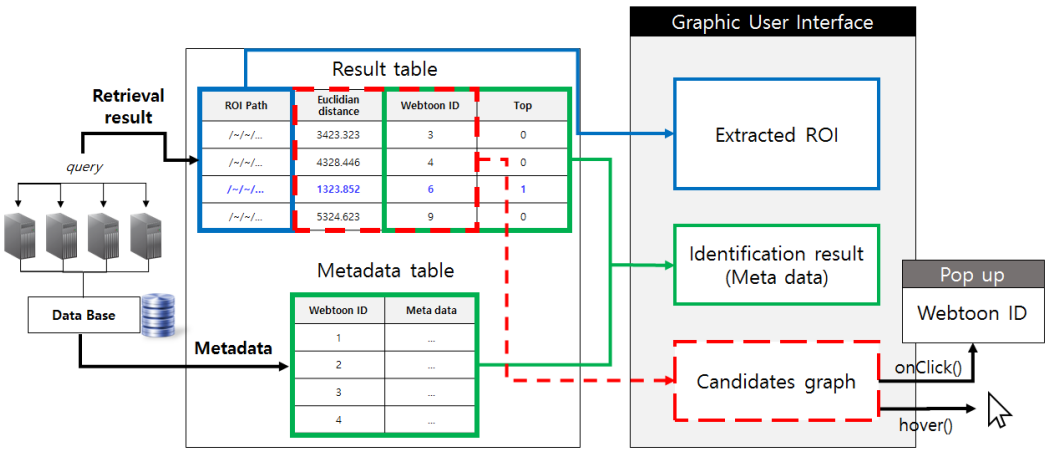



그림 8. 데이터베이스 테이블을 통한 데이터 검색 결과와 GUI 연동
Fig. 8. GUI connection with data retrieval result through database table

되도록 설정하였다. hover() 함수를 적용할 경우, 사용자가 해당 직사각형에 마우스를 올렸을 때 색상이 변하는 등의 설정을 추가 할 수 있다. 만들어진 GUI의 모습은 그림 9와 같이 그래프의 특정 바를 클릭하였을 때 해당 유클리디안 거리를 가지는 웹툰의 ID가 팝업창으로 나타나는 것을 확인할 수 있다.

## Ⅲ. 실험 결과

### 3.1 실험 환경

실험 시 사용한 CPU 사양 및 개발 환경은 다음과 같다. 서버 노드 1대와 클라이언트 노드 4대, 총 5대 동일 사양 노드를 사용하였다.

CPU : Intel core i5-6600 CPU @ 3.30GHz
RAM : 8GB
운영 체제 : Windows 10 64비트
개발 환경 : Microsoft Visual Studio 2013
영상 처리 라이브러리 : OpenCV 2.4.13
데이터 베이스 서버 : MySQL Server 5.7

실험에 사용한 웹툰 이미지 테이터 수와 파라메터는 표 1과 같다. 테스트 이미지는 원본에서 행 또는 열이 일부 손실된 이미지 10,000장을 사용하였다.

식별 성능을 최적화하기 위해 식별에 영향을 미치는 파라미터를 조절하며 성능을 측정하였다. Visual word 클러스터 개수(K), 분산 노드 수($N_n$), Visual word 엔트로피 가중치 여부를 조절하며 최적의 성능을 나타내는 값을 탐색하였다.

표 1. 실험에 사용된 웹툰 데이터 및 파라미터
Table 1. Webtoon data and parameter setting for simulation

| Webtoon Data | |
|---|---|
| No. of webtoon IDs in DB | 810 |
| No. of webtoon images in DB | 17,166 |
| Testing images | 10,000 |
| Parameters | |
| No. of clusters | K = 64 |
| No. of nodes | $N_n$ = 4 |

### 3.2 Visual word 클러스터링에 따른 식별 성능 분석

Visual word 클러스터링을 사용하지 않고 데이터베이스에 저장된 모든 핑거프린트를 전수조사 한 경우, 시간이 59.19 초로 정확도는 91.7%로 나타났다.

표 2. 클러스터 수에 따른 식별 성능
Table 2. Identification performance according to the number of clusters

| K | 0 | 27 | 64 | 125 | 216 |
|---|---|---|---|---|---|
| Accuracy (%) | 91.7 | 93.8 | 94.4 | 93.6 | 93.2 |
| Time (sec/ROI) | 59.19 | 2.37 | 1.14 | 1.09 | 1.03 |

클러스터 수 K를 증가시킴에 따라 초기에 검색해야할 핑거프린트수가 줄어들기 때문에 식별 시간도 짧아지고 정확도도 증가하였다.

클러스터의 수가 계속 증가하면 클러스터 중심점 사이의 간격이 줄어들어 클러스터가 오분류된 확률이 증가하며, 클러스터 중심점 사이의 거리를 계산하는데 소모되는 비용이 증가한다. 최적 클러스터의 개수는 특징점의 개수와 분포 형태 등에 따라 달라지며, 효율적인 검색을 위하여 최적의 군집개수를 능동적으로 바꿔주는 작업이 필요하다.

클러스터 수가 적은 경우 한 번에 검색해야하는 핑거프린트 수가 많기 때문에 식별 시간이 증가하며, 원본 데이터 외에 다른 웹툰 저작물의 핑거프린트 일치 확률도 함께 증가한다. 표 2와 같이 실험에 따라 클러스터 수를 조절하여 64개의 클러스터에서 식별 시간은 1.14초, 정확도는 94.4%로 최적의 성능을 나타냄을 확인하였다.

### 3.3 분산 노드 수에 따른 식별 성능 분석

분산 노드에서 핑거프린트를 여러 노드에 분배하여 검색을 동시에 처리하는 방식으로 시스템 고속화를 구현하였다. 표 3과 같이 노드 수 $N_n$를 증가시키면서 식별에 소요되는 시간을 측정해 보았다.

노드 수가 2개 일 때 1.05배, 3개일 때 1.12배로, 4개일 때 1.16배로 노드 수가 늘어남에 따라 병렬 매칭 연산의 효과로 식별시간이 감소했다. 본 논문에서는 최대 4대의 분산 노드를 사용하였지만, 노드를 추가하는 방식으로 시스템을 확장한다면 식별 시간을 더욱 감소시킬 수 있을 것이다.

표 3. 분산 노드 수에 따른 식별 성능
Table 3. Identification performance according to the number of nodes

| $N_n$ | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| Time (sec/ROI) | 1.32 | 1.26 | 1.18 | 1.14 |

### 3.4 Visual word 엔트로피 가중치 적용 유무에 따른 식별 성능 분석

Visual word 히스토그램을 사용하여 클래스를 결정한다. 히스토그램의 Visual word 카운트 값에 발생 빈도를 반영하는 엔트로피 가중치를 추가하여 식별 성능을 향상 시켰다.

그림 10은 가중치 유무에 따른 Visual word의 분포를 나타낸다. 가중치는 Visual word의 쏠림을 방지하기 때문에 가중치 적용 전에 비해 Visual word가 골고루 분포되었음을 확인할 수 있다. Visual word의 고른 분포는 결국 웹툰 이미지 클래스의 고른 분포를 의미한다. 표 4와 같이 클래스에 따른 엔트로피 계산 결과, 가중치를 적용한 경우 그렇지 않은 경우보다 엔트로피가 크게 나타남으로 클래스가 골고루 분배 되는 것을 확인할 수 있다. 클래스의 적절한 분배는 특정 클래스에 데이터가 집중적으로 저장되어 식별 정확도가 감소하는 것을 방지한다. 따라서 가중치 적용 결과 식별 정확도도 93.7%에서 94.4%로 향상됨을 알 수 있다.

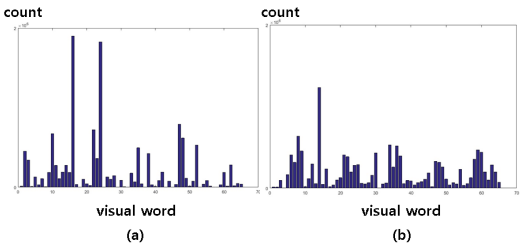



그림 10. 가중치 유무에 따른 Visual word 분포 (a) 가중치 적용 전 (b) 가중치 적용 후
Fig. 10. Visual word distribution according to weights (a) non-weight (b) weight

표 4. 가중치 유무에 따른 식별 성능
Table 4. Identification performance with or without weighting

| | Non-weight | weight |
|---|---|---|
| Accuracy (%) | 93.7 | 94.4 |
| Entropy | 4.68 | 5.36 |

## Ⅳ. 결 론

제안하는 diskless 분산 데이터베이스 프레임워크는, 각각의 클라이언트 노드가 HDD를 공유하는 DBMS를 활용하여 유사도 연산을 병렬적으로 수행함으로써 핑거프린트 검색 속도를 향상시켰다. Visual word를 사용한 클러스터링을 통해 검색량을 제한함으로써 웹툰 이미지 저작물 식별 성능을 고속화 하였다. 실험 결과를 통해 기존 전수조사 방식에 비해 검색 효율성이 증대되어, 크게 향상된 식별 정확도와 검색 시간 단축을 확인할 수 있었다.

클러스터링 클러스터 64개, 4개 분산 노드를 사용하였을 때 식별률 94.4% 식별시간 1.14초로 최적의 성능을 나타냈으며. 본 논문에서 제안하는 시스템으로 실제 웹상의 대용량 데이터를 매칭하는데 실제 시스템에 적용 가능한 성능임을 확인하였다.

diskless 시스템은 각각의 노드에서 병렬 연산이 독립적으로 진행되기 때문에 전체 일을 수량에 따라 배분하기 용이하며, 검색 알고리즘뿐만 아니라, 각 노드에 감지 역할을 분담함으로써 실시간으로 감지를 해야 하는 감시 시스템에도 효과적으로 활용할 수 있다. 또한, diskless 시스템의 하드웨어적인 이점을 활용한다면, 분산 처리 시스템의 클라이언트를 손쉽게 추가할 수 있으며 이를 통해 양적으로 시스템의 성능을 향상시킬 수 있다.

본 논문에서는 웹툰 이미지에서 추출된 프레임 하나를 식별 단위로 하는 시스템이다. 추후에 여러 개의 프레임에서 취합된 결과를 출력하면 보다 향상된 식별 결과를 얻을 수 있을 것이다.

## References

[1] H. Kim, *Copyright status of domestic cartoon industry*(2015), Retrieved Oct. 22, 2018, from www.copyright.or.kr.

[2] F. Chen, et al., “Data mining for the internet of things: literature review and challenges,” *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, p. 431047, 2015.

[3] M. Liu, X. Jiang, and A. C. Kot, “Efficient fingerprint search based on database clustering,” *Pattern Recognition*, vol. 40, no. 6, pp. 1793-1803, 2007.

[4] J. Yang, et al., “Evaluating bag-of-visual-words representations in scene classification,” *Proceedings of the international workshop on Workshop on multimedia information retrieval*, pp. 197-206, Augsburg, Germany, Sep. 2007.

[5] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” *European*

*conference on computer vision*, pp. 404-417, Berlin, Heidelberg, May 2006.

[6] O. A. Abbas, "Comparisons between data clustering algorithms," *The Int. Arab J. Inf. Technol.*, vol. 5, no. 3, pp. 320-325, Jul. 2008.

[7] A. Likas, N. Vlassis, and Jakob J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition*, vol. 36, no. 2, pp. 451-461, 2003.

[8] M. Muja and David G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 11, pp. 2227-2240, 2014.

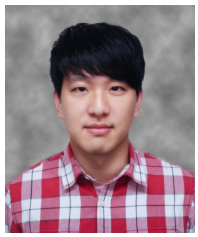
[9] D. Nister and Henrik Stewenius, "Scalable recognition with a vocabulary tree," *IEEE computer society conference on Computer vision and pattern recognition*, vol. 2, pp. 2161-2168, New York, USA, Jun. 2006.

[10] Tobin J. Lehman and Michael J. Carey, "A study of index structures for main memory database management systems," in *Proc. 12th International Conference on Very Large Data Bases*, vol. 1, pp. 294-303, Kyoto, Japan, Aug. 1986.

[11] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley, M. J. Franklin, S. Shenker, and I. Stoica, "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing," *Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation*, pp. 2-2, San Jose, USA, Apr. 2012.

[12] D. Kim, et al., "Robust fingerprinting method for webtoon identification in large-scale databases," *IEEE Access*, vol. 6, pp. 37932-37946, Jul. 2018.

**김 도 영** (Doyoung Kim)



2014년 2월 : 연세대학교 전기전자공학부 졸업

2014년 3월~현재 : 연세대학교 전기전자공학과 석·박사 통합과정

<관심분야> 인공 지능, 행동 인지, 영상 처리

**박 현 우** (Hyunwoo Park)

2014년 8월 : 연세대학교 의용전자공학과 졸업

2015년 3월~현재 : 연세대학교 전기전자공학과 석사과정

<관심분야> 영상 처리, 데이터베이스, 분산 시스템

**박 종 화** (Jonghwa Park)

2014년 8월 : 연세대학교 전기전자공학부 졸업

2014년 9월~현재 : 연세대학교 전기전자공학과 석사과정

<관심분야> 인공 지능, 데이터베이스, 분산 시스템

**권 혁 주** (Hyuck-Joo Kwon)

2009년 2월 : 세종대학교 컴퓨터공학과 졸업

2011년 2월 : 세종대학교 컴퓨터공학과 석사 졸업

2016년 8월 : 세종대학교 컴퓨터 공학과 박사 졸업

2016년 9월~현재 : 연세대학교 전기전자공학과 박사후 과정 연구원

<관심분야> 3차원 랜더링 프로세서 구조 설계, 하드웨어 가속, 실시간 레이캐스팅, 모바일 GPU

**이 상 훈** (Sanghoon Lee)

1989년 2월 : 연세대학교 전기공학과 졸업
1991년 2월 : 한국과학기술원 전기공학과 석사 졸업
2000년 1월 : 텍사스 오스틴 대학교 전기공학과 박사 졸업
2003년 3월~현재 : 연세대학교 전기전자공학과 교수

<관심분야> 인공 지능, 영상 처리, 분산 시스템