Journal of The Korea Society of Computer and Information
Vol. 24 No. 1, pp. 57-65, January 2019
www.ksci.re.kr
https://doi.org/10.9708/jksci.2019.24.01.057

An Efficient Comparing and Updating Method of Rights Management Information for Integrated Public Domain Image Search Engine

Il-Hwan Kim*, Deok-Gi Hong*, Jae-Keun Kim*, Young-Mo Kim*, Seok-Yoon Kim*

Abstract

In this paper, we propose a Rights Management Information(RMI) expression systems for individual sites are integrated and the performance evaluation is performed to find out an efficient comparing and updating method of RMI through various image feature point search techniques. In addition, we proposed a weighted scoring model for both public domain sites and posts in order to use the most latest RMI based on reliable data. To solve problem that most public domain sites are exposed to copyright infringement by providing inconsistent RMI(Rights Management Information) expression system and non-up-to-date RMI information. The weighted scoring model proposed in this paper makes it possible to use the latest RMI for duplicated images that have been verified through the performance evaluation experiments of SIFT and CNN techniques and to improve the accuracy when applied to search engines. In addition, there is an advantage in providing users with accurate original public domain images and their RMI from the search engine even when some modified public domain images are searched by users.

▸Keyword: Public Domain Image, RMI, dHash, Average Hash, CNN, Weighted Scoring Model

I. Introduction

현재 공유마당, 플리커, 픽사베이와 같은 국내·외 이미지 저작물 서비스 플랫폼으로부터 10억 개가 넘는 자유이용저작물들이 기업과 개인 사용자들에게 제공되고 있다. 특히 개인 및 소호 사업자와 같은 1인 기업, 개발자 및 디자이너 등이 저작권으로부터 자유롭게 사용 가능한 자유이용저작물에 큰 관심을 보이고 있다. 이처럼 공유저작물에 대한 수요는 계속 증가하고 있으나 저작권에 대해 정확한 이해를 하지 못한 채 공유저작물을 사용하여 다양한 저작권 이슈에 노출되어 있다[10].

이는 저작물에 대한 권리관리정보(Rights Management Information)의 비 현행화, 공유저작물 제공 사이트들의 일관되지 못한 권리관리정보 표현체계 등으로 인해 발생한다. 이러한 문제점을 해결하기 위해서 각기 다른 권리관리정보 표현체계를 효과적으로 관리하는 통합방안과 항상 최신의 저작물의 권리관리정보를 제공하는 지능정보 기반 공유저작물 검색엔진 개발 연구가 선행되고 있다[1][9]. 하지만 사용자가 원본 공유저작물이 아닌 수정된 공유저작물을 검색할 경우 정확한 정보를 제공하지 못한다는 문제점과 검색된 저작물이 여러 출처로부터

• First Author: Il-Hwan Kim, Corresponding Author: Young-Mo Kim
*Il-Hwan Kim (kih3159@naver.com), Dept. of Computer Science and Engineering, Soongsil University
*Deok-Gi Hong (dukihong13@gmail.com), Dept. of Computer Science and Engineering, Soongsil University
*Jae-Keun Kim (jaekeun0310@gmail.com), Dept. of Computer Science and Engineering, Soongsil University
*Young-Mo Kim (ymkim828@ssu.ac.kr), Dept. of Computer Science and Engineering, Soongsil University
*Seok-Yoon Kim (ksy@ssu.ac.kr), Dept. of Computer Science and Engineering, Soongsil University
• Received: 2018. 12. 03, Revised: 2018. 12. 30, Accepted: 2019. 01. 01.
• This work was supported by Ministry of Culture, Sport and Tourism (MCST) and Korea Copyright Commission in 2018(2017-SHARE-9500).

중복된 문제가 발생한 경우 어떤 저작물을 기준으로 갱신 및 비교를 하는지에 대한 문제가 발생한다.

따라서 본 논문에서는 이를 해결하기 위해 사이트별 권리관리정보 표현체계를 통합하고 다양한 이미지 특징점 비교검색기법을 통해 효율적인 방법을 알기 위해 실험 성능평가를 수행한다. 또한, 신뢰성 있는 데이터를 기준으로 최신의 권리관리정보로 현행화하기 위하여 공유저작물 사이트와 게시글별 Weighted Scoring Model을 제안한다.

본 논문의 구성으로는 2장에서는 이미지 해시를 이용한 이미지 유사도 기법과 이미지 특징점을 활용한 검출방법에 대해 살펴보고 3장에서는 본 논문에서 제안하는 RMI 통합방안과 지능정보기반 공유저작물 검색엔진 아키텍처, Weighted Scoring Model에 대하여 자세히 기술하고 4장에서는 결론으로 논문을 마친다.

II. Preliminaries

1. dHash & Average Hash

파이썬의 라이브러리 중의 하나인 dhash는 이미지의 행과 열에 대한 hash 정보를 생성하여 결합하는 방식으로 우선 원본 이미지를 흑백(Grayscale)으로 변환하여 픽셀값을 0~255로 만들어준다. 흑백으로 변환된 이미지를 공통 크기를 줄여 강도(Intensity)만으로 판단할 수 있게 한다. 이러한 dHash를 사용시 공유저작물 사이트별로 이미지의 크기가 상이하게 등록되어도 같은 이미지로 찾는 것이 가능하다[2].

Average hash는 이미지를 비교 가능한 해시값으로 나타낸 것이다. 해시 함수는 MD5, SHA256 등은 이미지의 데이터 값을 간단한 해시값으로 변환하여 비교검색을 할 수 있다는 장점이 있지만, 이미지의 밝기, 기울기, 크기, 해상도 등 여러 요인으로 인하여 이미지의 변경이 이루어지면 유사한 이미지로 검출할 수 없는 단점이 있다. 반면 Average Hash는 이미지의 Binary 값을 통해 Hamming Distance를 계산하여 유사한 이미지를 검출할 수 있는 장점이 있다[3].

2. SIFT

SIFT(Scale-Invariant Feature Transform) 알고리즘은 대표적인 특징점 기반 인식 알고리즘으로 이미지의 크기 및 회전에 불변하는 특징을 추출한다. SIFT 알고리즘은 Scale-space extrema detection, Keypoint localization, Orientation assignment, Keypoint Descriptor의 4단계로 이루어져 있다.

첫 번째 단계인 Scale-space extrema detection은 특징의 크기와 위치를 결정하는 것으로 Scale과 Orientation에 불변한다고 추측되는 관심 영역을 추출한다. 이를 위해 Fig. 1. 과 같은 Gaussian 피라미드가 필요하며 Dog(Difference of Gaussian)를 이용한다[3].

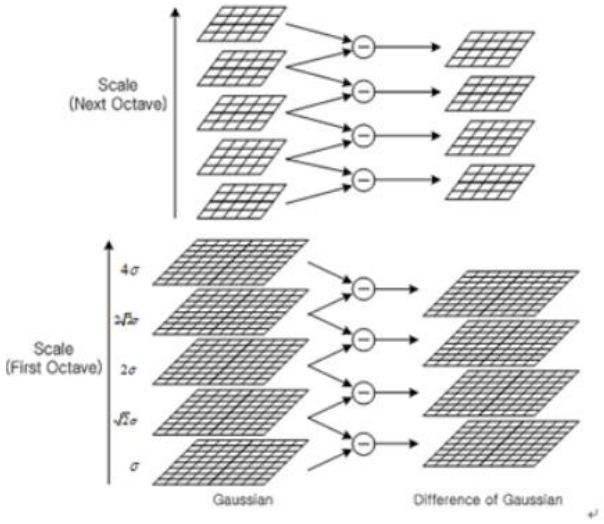

Fig. 1. Gaussian Pyramid

두 번째 단계인 Keypoint Localization에선 Keypoint 중 활용도가 떨어지는 것들을 제거하고 Taylor 급수를 사용하여 Keypoint를 Integer Domain이 아닌 연속 공간에 위치시킨다. 이때 근사 된 영상은 D 값에 다른 극점을 가지게 되고 이 차이를 이용해 조정할 수 있다. 그다음 단계인 Orientation assignment는 keypoint 주변 픽셀의 방향을 아래의 (Expression 1)을 이용하여 계산하고 Fig. 2. 과 같은 Orientation histogram을 만든다[4].

$$m(x,y)=\sqrt{\left(L(x+1,y)-L(x-1,y)\right)^2+\left(L(x,y+1)-L(x,y-1)\right)^2}$$

$$\theta(x,y)=\tan^{-1}\left(\frac{L(x,y+1)-L(x,y-1)}{L(x+1,y)-L(x-1,y)}\right)$$

..... (Exptression 1)

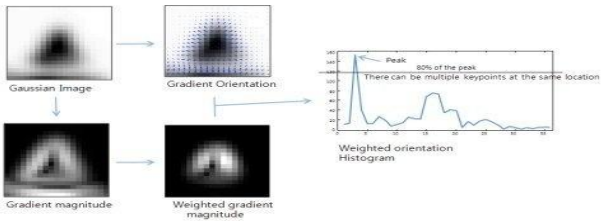

Fig. 2. Orientation Histogram

마지막 단계인 Keypoint Descriptor는 Keypoint를 중심으로 주변에 있는 Gradient 값들의 방향을 구하며 구하여진 값들은 Gaussian Window를 사용하여 가중치를 적용한다. 이때 Orientation Invariant를 위하여 Gradient 값을 구할 때 사용하는 좌표를 Keypoint 방향으로 회전시켜 Sampling하고 Illumination 불변을 위하여 Orientation 값을 정규화한다[4].

3. CNN

합성곱 신경망(Convolutional Neural Network)은 이미지 인

식 분야에서 기초적인 구조로서 완전 연결 계층(Fully Connected Layer) Fig. 4. 와 달리 합성곱 계층(Convolutional Layer)과 풀링 계층(Pooling Layer)이 더해진 구조 Fig. 3. 이다. Fig. 4.를 보면 출력에 가까운 층에서는 'Affine-ReLU' 구성을 사용할 수 있으며, 마지막 출력 계층에서는 'Affine-Softmax' 조합을 그대로 사용한다[5][6].

CNN에서는 합성곱 계층의 입출력 데이터를 Feature Map이라 하며 합성곱 계층의 입력 데이터를 Input Feature Map, 출력 데이터를 Output Feature Map이라 한다[5][6].

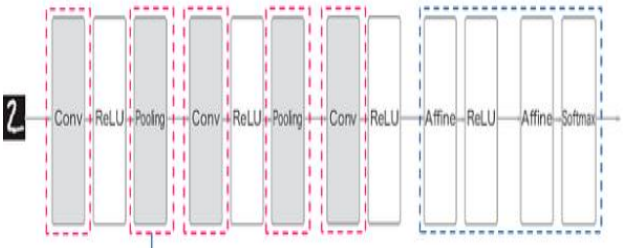

Fig. 3. Network with CNN

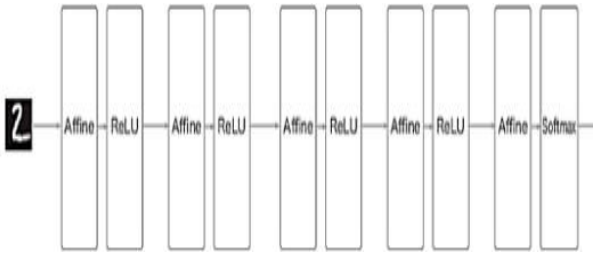

Fig. 4. Network Consisting of A complete Connection Layer (Affine layer)

본 논문에서 적용하는 특징점 Numpy Matrix 합성곱 연산은 Fig. 5.과 같이 진행한다. Fig. 5.와 같이 입력 데이터에 필터를 적용하며 입력 데이터는 세로, 가로 방향의 형상을 가지고 있으며 필터 역시 가로세로 방향의 차원을 갖게 된다. 합성곱 연산의 필터는 Window를 일정 간격으로 Sliding 하며 입력 데이터에 Fused Multiply Add 연산을 수행하게 되며 이 과정을 전부 수행하게 되면 합성곱 연산 출력이 완료된다.

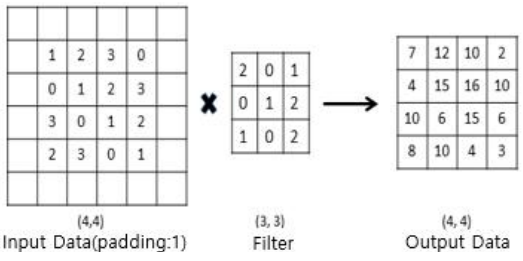

Fig. 5. Example of Convolution Operation

III. Comparison and Update Method of Integrated Public Domain Image Search Engine

본 논문에서는 이미지 공유저작물 사이트로부터 필요한 이미지 공유저작물과 메타데이터를 Clawler 기술을 이용해 수집하여 HDFS 기반 Crawler_DB에 저장한다. Feature Point Extraction 모듈을 통해 dHash, Average Hash, SIFT Algorithm, CNN 특징점 추출 성능평가를 수행하여 결과를 저장한다. 그리고 Search Engine_DB의 공유저작물 데이터와 매칭을 수행하며 최신정보로 Update Process를 수행하도록 한다. 수행 도중 두 곳 이상의 플랫폼에서 같은 공유저작물이 존재할 경우 본 연구에서 제시한 Weighted Scoring Model로 신뢰도가 높은 결과를 기준으로 Search Engine_DB에 갱신하도록 한다.

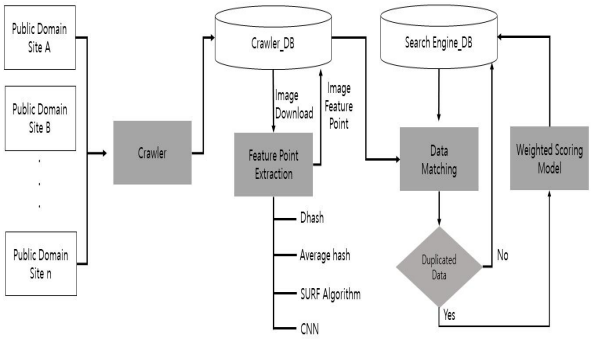

Fig. 6. Public Domain Image Search Engine Architecture

1. RMI Integration

공유저작물 사이트별 상이한 권리관리정보 표현으로 인한 이용자들의 혼란감소와 Search Engine_DB 관리 용이성을 위해 각각 사이트들의 권리 표현을 하나로 통합하였다. 공유저작물 사이트들에서 권리 표현 방법으로 주로 이용되는 CCL 6형을 기본으로 통합을 하였으며, 만료저작물, 고아저작물 등의 자유롭게 이용할 수 있는 저작물에 한해선 자유이용 저작물로 결합하여 아래 Table 1.처럼 스키마로 정의하였다.

Table 1. RMI Database Scheme

RMI_ code	RMI_ eng	Explanation
rmi_01	CC BY	Share — copy and redistribute the material in any medium or format Adapt — remix, transform, and build upon the material for any purpose, even commercially.
rmi_02	CC BY-NC	Share — copy and redistribute the material in any medium or format Adapt — remix, transform, and build upon the material
rmi_03	CC BY-ND	Share — copy and redistribute the material in any medium or format for any purpose, even commercially.

rmi_04	CC BY-SA	Share — copy and redistribute the material in any medium or format Adapt — remix, transform, and build upon the material for any purpose, even commercially.
rmi_05	CC BY-NC-SA	Share — copy and redistribute the material in any medium or format Adapt — remix, transform, and build upon the material
rmi_06	CC BY-NC-ND	Share — copy and redistribute the material in any medium or format
rmi_07	FREE USE	The author can give up his rights or use it without any restriction over the copyright term.

아래의 Table 2, Table 3은 공유마당, Flickr, Europeana 각각 통합된 RMI를 보여주고 있으며 기존의 CCL 형식을 제외한 저작물을 이용하는 데 어떠한 제약도 존재하지 않는 RMI에 한해서 위 Table 1에서 제안한 rmi_07인 'FREE USE' 자유이용 저작물로 통합하였다. 또한, 사이트별 이미지 형식의 저작물이 아니거나 저작권자가 모든 저작권을 가지고 있어 이용자가 자유롭게 이용할 수 없는 저작물은 제하였으며 저작권 정보를 정확하게 알 수 없어 이용자가 이용하는데 분쟁이 일어날 가능성이 있는 저작물 또한 Metadata 통합 단계에서 제외하였다.

Table 2. Integrated RMI Schema of 'Gong-U Madang'

Gong-U Madang RMI	Integrated RMI	Remarks
CC-BY	rmi_01	
CC-BY-NC	rmi_02	
CC-BY-ND	rmi_03	
CC-BY-SA	rmi_04	
CC-BY-NC-SA	rmi_05	
CC-BY-NC-ND	rmi_06	
Expired Assets	rmi_07	
Donation (Free use)	rmi_07	
Donation (Permission)	rmi_07	
Gong-Gongnuli 1 Type	X	Not image format
Gong-Gongnuli 2 Type	X	Not image format
Gong-Gongnuli 3 Type	X	Not image format
Gong-Gongnuli 4 Type	X	Not image format

Table 3. Integrated RMI Schema of Flikr

Flickr RMI	Integrated RMI	Remarks
CC-BY	rmi_01	
CC-BY-NC	rmi_02	
CC-BY-ND	rmi_03	
CC-BY-SA	rmi_04	
CC-BY-NC-SA	rmi_05	
CC-BY-NC-ND	rmi_06	
No known Copyright Restrictions	rmi_07	
Public Domain	rmi_07	
All Rights Reserved	X	Copyright Protected
U.S. Government Works	X	Copyright Protected

2. Image Comparison Search Technique

2.1. Average Hash & dHash Similarity Detection

Average Hash의 결과값으로 Numpy Matrix 산출 과정은 Fig. 7. 과 같다. 공유저작물 이미지들을 16*16 Numpy Matrix를 구하기 위해 8*8 크기로 Resizing을 진행한다. 그리고 Gray Scale로 변환하여 이미지 각 픽셀에 대한 데이터 평균을 계산하여 벡터 당 평균 밝기보다 낮으면 0 높으면 1로 이진화하여 Numpy Matrix로 추출하게 된다. 본 논문에서는 여러 이미지 중에서 유사한 이미지를 검색하기 위해 표본 이미지로 캘리포니아 공과대학에서 머신러닝을 위해 배포하고 있는 'Computational Vision at CALTECH'를 사용하였다.

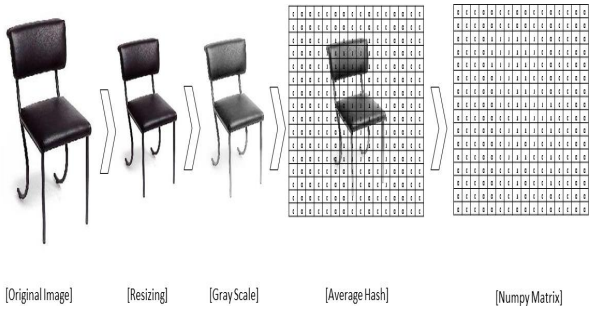

Fig. 7. Average Hash

검색하려는 이미지와 유사한 이미지를 검출하기 위해서는 Numpy Matrix의 벡터값들을 Hamming Distance로 계산하였으며. Caltech 101 이미지 데이터를 통해 기준 이미지와 비슷한 이미지를 검색하였을 때 75%의 유사도를 가졌다. 검출한 실행 결과는 Fig. 8. 와 같으며 출력한 결과는 Fig. 9. 과 같다.

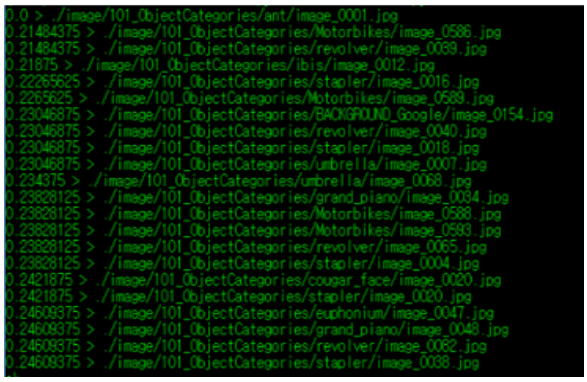

Fig. 8. Average Hash Execution Result

Fig. 9. Similar Image Search Results using Average Hash

Fig. 9. 는 결과 기준이 되는 이미지와 동일한 이미지와 비슷

한 이미지도 검출이 되었지만, 전혀 상관이 없는 이미지도 검출되어 Overfitting이 발생하였다. 따라서 Average Hash 방법을 통한 유사 이미지 검출방법은 처리 속도와 이미지 색상, 형식, 해상도에 구분 없이 검출할 수 있다는 장점이 있지만, 이미지 유사도에 따른 강인성이 다소 낮다는 것을 확인할 수 있다.

dHash 라이브러리는 다른 이미지 검출 라이브러리인 aHash, pHash와 거의 같지만 aHash는 이미지 평균 픽셀값에 중점을 두고, pHash는 Color Frequency를 계산하지만 dHash는 Image Gradient를 계산하므로 훨씬 정확하다는 장점이 있다.

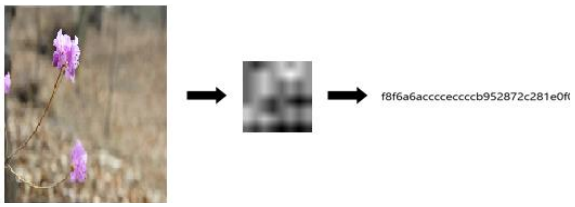

Fig. 10. dHash

dHash 알고리즘 작동 방식은 Fig. 10. 과 같다. 크기와 종횡비의 관계없이 비슷한 그림을 검색할 수 있도록 이미지를 일정한 크기로 축소한다. 색상을 Gray Scale로 변환하여 해시가 72픽셀에서 총 72색으로 변경하여 인접한 픽셀 차이를 계산하여 Gradient 방향을 정하며 행 당 9픽셀의 인접한 8개의 픽셀 차이를 계산하여 32자리 16진수 해시값을 출력한다.

2.2. SIFT & CNN Image Similarity Detection

이미지 공유저작물 비교검색을 위해 이미지의 회전, 질감 변경, 밝기, 노이즈를 추가한 후 매칭 정확도 산출을 위해 SIFT 알고리즘을 사용하여 Fig. 11. 와 같이 실험을 진행하였다.

```
FLANN_INDEX_LSH = 6
index_params = dict(algorithm=FLANN_INDEX_LSH, table_number=6, key_size=12, multi_probe_level=1)
search_params = dict(checks=50)
# FLANN_INDEX_KDTREE = 0
# index_params = dict(algorithm=FLANN_INDEX_KDTREE, trees=5)
# search_params = dict(checks=100)

flann = cv2.FlannBasedMatcher(index_params,search_params)
matches = flann.knnMatch(des1,des,k=2)

good = []
for m, n in matches:
    if m.distance < 0.7 * n.distance:

        good.append(m)

img1 = cv2.rectangle(img1, (0, 300), (211,0), (0,0,255), 3)
gray = cv2.drawMatches(img1,kp1,img,kp,good, None,(0, 255, 0), flags=0)
msg1 = 'Feature Count %d' % (fn)
msg = 'there are %d good matches' % (len(good))
font = cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(gray, msg1, (230, 260), font, 0.5, (0, 0, 0), 1, cv2.LINE_AA)
cv2.putText(gray, msg, (190, 280), font, 0.5, (0, 0, 0), 1, cv2.LINE_AA)
cv2.imwrite('matching2.jpg', gray)
```

Fig. 11. SIFT Algorithm

실험결과 Fig. 12. 와 같이 도출되었다. 밝기 및 질감이 변경됐을 경우, 총 90개의 특징점에서 42개의 특징점이 매칭되었으며 이미지를 90° 회전하였을 경우 총 90개의 특징점에서 82개의 특징점을 매칭하였다. 실험결과 이미지의 Color Gradation이 변경된 것보다 이미지 반전에서 매칭률이 높은 것을 확인하였다.

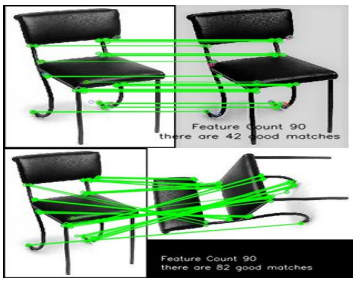

Fig. 12. SIFT Algorithm Result

앞선 dHash, Average Hash, SIFT 알고리즘을 통해 추출한 데이터로 이미지 공유저작물을 비교검색을 수행했을 때 약간의 수정된 이미지 검출과 속도에서 장점이 있지만, 빅데이터 기반 검색엔진에서 사용할 만큼 정확한 데이터를 도출하지 못한다는 단점이 있다. 더욱 정확도를 높이기 위해 CNN 모델을 구축하여 합성곱층, 활성화 함수(ReLU), 맥스 풀링층을 더하여 학습 정밀도를 높였다[7][8]. Overfitting 발생을 줄이기 위해 'Computational Vision at CALTECH' 데이터 셋 학습과 MNIST 데이터 셋을 더하여 TensorFlow + Keras 조합으로 CNN을 테스트하였다.

Fig. 13. 와 Fig. 14. 같이 판정 정밀도를 높이기 위해 이미지의 각도, 수평 및 수직 반전을 통해 학습하여 predict() 메서드로 여러 개의 데이터를 예측할 수 있게 매개변수로 배열을 지정하여 92% 정확도를 산출하였다.

```
# Model Architect --- (※2)
def build_model(in_shape):
    model = Sequential()
    model.add(Convolution2D(32, 3, 3,
        border_mode='same',
        input_shape=in_shape))
    model.add(Activation('relu'))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
    model.add(Convolution2D(64, 3, 3, border_mode='same'))
    model.add(Activation('relu'))
    model.add(Convolution2D(64, 3, 3))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    model.add(Dropout(0.25))
    model.add(Flatten())
    model.add(Dense(512))
    model.add(Activation('relu'))
    model.add(Dropout(0.5))
    model.add(Dense(nb_classes))
    model.add(Activation('softmax'))
    model.compile(loss='binary_crossentropy',
        optimizer='rmsprop',
        metrics=['accuracy'])
    return model
```

Fig. 13. CNN Model Construction

```
# Model Training --- (※3)
def model_train(X, y):
    model = build_model(X.shape[1:])
    model.fit(X, y, batch_size=32, nb_epoch=30)
    # Model Saving --- (※4)
    hdf5_file = "./image/test-model.hdf5"
    model.save_weights(hdf5_file)
    return model
# Model Evaluation --- (※5)
def model_eval(model, X, y):
    score = model.evaluate(X, y)
    print('loss=', score[0])
    print('accuracy=', score[1])
if __name__ == "__main__":
    main()
```

Fig. 14. CNN Learning

2.3. Image Similarity Detection Performance Evaluation

본 논문에서는 사이트별 RMI 표현체계를 통합하고 다양한 이미지 특징점 비교검색기법을 통해 효율적인 방법을 알기 위해 실험 성능평가를 하였다. 6000개의 이미지 검색 실험결과는 Table 4. 와 같이 도출되었다. 검색속도는 이미지 해시를 이용한 유사도 검출방식이 속도와 정확성이 높은 것을 확인하였다. 하지만 원본이 아닌 약간의 수정된 이미지를 인식하는 확률이 낮음을 볼 수 있다. 반대로 SIFT와 CNN을 활용한 특징점 비교 방식은 이미지 해시를 이용한 방식보다 처리 속도가 느리지만 이미지 인식의 정확성과 수정된 이미지로 원본 이미지를 매칭하는 인식률이 높은 것을 확인하였다.

Table 4. Results of Image Feature Point Comparison Search Performance

	Processing speed	accuracy	변경점 인식
dHash Similarity Detection	4m 32s	97%	21%
Average Hash Similarity Detection	6m 1s	95%	44%
SIFT Similarity Detection	9m 43s	91%	61%
CNN Similarity Detection	15m 16s	92%	87%

3. Weighted Scoring Model

매칭되는 데이터가 여러 공유저작물 사이트에서 중복 데이터로 검출될 경우 어떤 사이트를 가장 정확한 데이터로 판별할 것인지에 대한 문제가 발생하게 되고, 해결방안으로 Weighted Scoring Model을 설계하여 가장 신뢰도가 높다고 판단되는 데이터를 기준으로 갱신을 진행하게 된다[8]. 데이터 신뢰성을 판별하는 Weighted Scoring Model 기법으로 본 논문에선 Table 5.와 같은 항목들을 평가할 것을 제안한다.

Table 5.의 평가항목표는 사이트의 신뢰도를 측정하는 평가항목과 사이트 내의 게시글의 신뢰도를 측정하는 평가항목 두 가지로 이루어져 있다. 사이트의 평가표에는 운영단체, 최근 한 달 내의 사이트 방문횟수 및 게시물 등록 수, 총 게시물 등록 수, 등록 방법 따라 가중치를 다르게 하였다. 총 게시물 등록 수 및 최근 한 달 내의 사이트 방문횟수와 같은 평가항목은 공유저작물 사이트 중 최댓값을 구해 그 최댓값으로부터 점수를 차등 분배하였다.

Table 5. Evaluation Items of Weighted Scoring Model

Site Reliability	Posts Reliability
Number of Recent Visits	Number of Views
Number of Recent Posts Registered	Download Count
Total Posts Registration	Meta Information Count
Upload Type	Update Date

마지막으로 개인 등록 여부란 플리커와 같이 저작자 개인이 직접 사이트에 업로드하는 방식과 공유마당과 같이 관리자의 승인을 받아 업로드하는 두 방식이 있다. 관리자의 승인을 받는 경우 중간에 권리관리정보 변경 신청을 했을 경우, 관리자의 처리 시간으로 인해 반영하는 것이 다소 늦어질 수 있지만, 개인이 직접 사이트에 업로드하는 방식은 언제나 즉시 수정이 가능하므로 보다 높은 가중치를 산정하였다. 이때 개인이 직접 업로드하는 방식은 자신이 자신의 저작물을 게시할 경우에만 한한다.

게시글 신뢰도의 평가항목의 경우 조회 수, 다운로드 수, 업데이트날짜. 저작물에 대한 메타 정보의 수 등의 항목들에 대해 가중치를 다르게 하여 평가표를 작성했다. 조회 수, 다운로드 수의 경우 중복으로 검출된 저작물 사이트들의 게시물을 서로 비교하여 점수를 분배하였으며 저작물에 대한 메타 정보를 얼마나 제공하느냐에 따라 점수를 분배하였다. 그중 업데이트날짜가 가장 최신일 경우 신뢰도 점수에 있어 가장 큰 가중치를 두었다. 두 평가항목표의 각각의 항목들은 그 중요도에 따라 가중치를 서로 다르게 주었으며 각 항목에 매겨진 가중치는 Table 6.와 같다. Table 6.을 기반으로 각 사이트 신뢰도와 게시글 신뢰도 에 대한 점수를 나타내는 (Expression 2)와 같다.

Table 6. Rate this Item by Weight of the Weighted Scoring Model

Site Reliability		Posts Reliability	
Category	Weight	Category	Weight
Number of Recent Visits (nrv)	30%	Number of Views (nv)	10%
Number of Recent Posts Registered (nrp)	20%	Download Count (dc)	25%
Total Posts Registration (tpr)	20%	Meta Information Count(mic)	15%
Upload Type (ut)	30%	Update Date (date)	50%

$$
\begin{aligned}
&Site\,Reliability \\
&= 30\% * nrv + 20\% * nrp + \\
&\quad 20\% * tpr + 30\% * ut \qquad \cdots\cdots\cdots\cdots \\
&Article\,Reliability \qquad \text{(Expression 2)} \\
&= 10\% * nv + 25\% * dc + \\
&\quad 15\% * mic + 50\% * date
\end{aligned}
$$

Fig. 15. 이미지에 대한 갱신 작업을 시행하였을 때 A 사이트와 D 사이트에서 동일한 이미지가 검출되었다. 이러한 경우에서 먼저 각 사이트에 대한 평가항목들에 대한 실제 데이터를 수집하고 항목별로 점수를 계산하여 각 사이트에 대한 신뢰도를 구한다. Table 7.은 각 사이트의 평가항목에 대한 수치를 나타내며 Table 8.은 사이트별 신뢰도 점수를 나타낸 표이다.

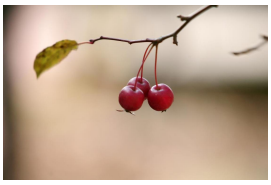

Fig. 15. Duplicated Images of Sites

각 항목에 대한 점수는 최대 50점까지 산정하였으며, 계산식에 따라 점수에 가중치를 적용한 수치를 나타내는 표는 Table 8. 이며 마지막 항목인 업로드 방식 항목으로 저작자가 직접 저작물을 등록/수정하는 B와 D 사이트에 가산점 15점을 산정하였다.

Table 9.은 중복된 이미지가 검출된 A와 D 사이트의 게시글에 대한 수치를 나타내며 Table 10.는 수치를 이용한 게시판의 신뢰도를 구한 표이다. 평가항목별 점수 계산식은 사이트 신뢰도를 구하였던 계산식과 같으며 업데이트날짜가 가장 빠른 사이트에만 가산점을 부여하였다.

Table 7. Results of Evaluation by Public Domain Site

	Site A	Site B	Site C	Site D
Number of Recent Visits	9,876	17,554	2,350	6,286
Number of Recent Posts Registered	1,241	3,211	890	638
Total Posts Registration	14,855,462	485,321.540	9.345.621	13,669,748
Upload method	Admin Approval	Direct Registration	Admin Approval	Direct Registration

Table 8. Results of Reliability Calculation by Site

	Site A	Site B	Site C	Site D
Number of Recent Visits (weight 30%)	8.43	15.00	2.00	5.37
Number of Recent Posts Registered (weight 20%)	3.86	10.00	2.77	1.98
Total Posts Registration (weight 20%)	0.30	10.00	0.19	0.28
Upload method (weight 30%)	0.00	15.00	0.00	15.00
Total	12.59	50.00	4.96	22.63

Table 9. Results of Evaluation by Duplicate Image Article

	Site A	Site D
Number of Views	145	170
Download Count	65	50
Meta Information Count	8	8
Update Date	2016-01-24	2017-06-18

Table 10. Results of Duplicate Image Reliability Calculation

	Site A	Site D
Number of Views (weight 10%)	4.26	5.00
Download Count (weight 25%)	12.5	9.61
Meta Information Count (weight 15%)	7.50	7.50
Update Date (weight 50%)	0	25.00
Total	24.26	47.11

앞서 산출한 사이트 신뢰도와 게시판 신뢰도를 이용하여 두 사이트의 신뢰도를 비교한다. 이때 사이트 신뢰도에는 가중치를 40%, 게시판 신뢰도에는 가중치를 60%씩 주었다. A 사이트의 경우 신뢰도의 총점이 19.58이고 D 사이트의 경우 신뢰도의 총점이 37.31이므로 D 사이트의 데이터가 더 신뢰성이 높다고 판단하여 D 사이트 데이터를 기준으로 갱신 작업을 실행한다.

Table 11. Site A, D Reliability Comparison

	Site A	Site D
Site Reliability(weight 40%)	5.03	9.05
Board Reliability(weight 60%)	14.55	28.26
Total	19.58	37.31

따라서 이미지 공유저작물이 다른 권리관리정보를 가지고 상이한 공유저작물 사이트에 업로드되었을 때 아래 Fig. 16. 과 같이 게시글 신뢰도 기준에 따라 권리관리정보가 rmi_04에서 rmi_02로 갱신되는 것을 확인할 수 있다.

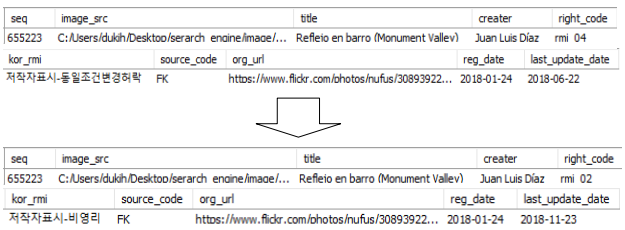

Fig. 16. Weighted Scoring Model-based update results

IV. Conclusions

본 논문에서는 지능정보기반 공유저작물 검색엔진에서 사용자가 정확한 이미지와 권리관리정보를 확인할 수 있도록 RMI 통합과 이미지 특징점 비교검색 성능평가, 검색엔진 DB에서 중복 데이터가 발생할 시 설계한 Weighted Scoring Model을 이용하여 높은 기준으로 갱신하는 아키텍처를 제안하였다.

이미지 특징점 비교검색 성능평가에서는 6000개의 이미지를 대상으로 하는 실험결과에서는 검색속도는 hash를 이용한 방식이 비교적 빠르지만 약간의 수정된 이미지를 원본 이미지

로 인식하는 확률이 낮음을 볼 수 있다. 반대로 SIFT와 CNN을 활용한 특징점 비교방식은 수정된 이미지를 인식하는데 높은 인식률을 보인 것을 확인하였다.

또한, 지능정보기반 공유저작물 검색엔진에서 이미지 검색을 통해 올바른 권리관리정보를 제공하기 위하여 제안한 Weighted Scoring Model을 통해 사용자가 수정된 이미지를 검색하더라고 원본 이미지와 최신의 권리관리정보를 검색엔진에서 제공하도록 하였다.

따라서 본 연구를 통해 신뢰도 높은 공유저작물 사이트와 게시글을 기준으로 갱신하여 중복문제에 대한 검색엔진의 정확성을 향상할 수 있으며, 약간의 수정된 저작물을 사용자가 검색하여도 검색엔진에서 정확한 원본 저작물과 권리관리정보를 사용자에게 제공함으로써 정확한 저작물 사용에 도움이 될 것이라 기대된다. 향후 추가적인 연구로는 CNN을 활용한 이미지 특징점 학습 속도의 개선을 위한 데이터셋 분류, 개선된 딥러닝 알고리즘 적용연구와 제안된 Weighted Scoring Model보다 신뢰성 있는 데이터 제공을 위해 평가항목 선정에 관한 지속적인 구가 필요할 것으로 사료된다.

REFERENCES

[1] H. Deok-Gi, K. Il-Hwan, K. Youngmo and K. Seok-Yoon "A Rights Management Information Updating Technique Using Image Feature Points," Proceedings of the Korean Society of Computer Information Conference. Vol. 26, No. 2, pp. 462-464, July 2018.

[2] HackerFactor , *http://www.hackerfactor.com/*

[3] K. hikodukue, "*Python NI YORU SCRAPING & KIKAIGA KUSHU KAIHATSU TECHNIQUE*" Japan-Press, pp. 301-320, 2016.

[3] D.G Lowe, "Object recognition from local scale-invariant features,"Proceedings of the Seventh IEEE International Conference on Computer Vision. Vol. 2, p 1150, September 1999.

[4] bskvision, *http://http://bskyvision.com/21*

[5] A. Krizhevsky, I. Sutskever and G, E, Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Proceedings of the NIPS'12 Proceedings of the 25th International Conference on Neural Information Processing Systems, Vol 1, pp. 1097-1105, 2010.

[6] S. goki, "*Deep Learning from Scratch*" Japan-Press, pp.227-235, 2016.

[7] J. SangYup, K. HoYeon and C. Jae-Soo, "Estimation of Non-trained Category in Image Classification Model based on Deep-Learning," Journal of Institute of Control, Robotics and Systems, Vol. 24, No. 9, pp. 793-801, 2018.

[7] K. MyungJoon and P. Byeonghwa, "A Study on Big Data Application using Scoring Model," Jounal of Actuarial Science, Vol. 7, No. 2, pp. 3-22, 2015

[8] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, No. 6, pp. 1137-1149, 2017.

[9] EUN-JI SEO, Il-Hwan Kim, Deok-Gi Hong, Youngmo Kim and Seok-Yoon Kim, "Integrated Multilayer Metadata Based on Intellectual Information Technology for Customization Services of Public Domain Images with Different Usage Permission of License," 28th Journal of Theoretical and Applied Information Technology, Vol. 96, No. 4, pp. 1048-1058, 2018.

[10] N. Hyun Woo, "Research on the Improvement Plan for the Protection and Use Activation of Image Copyright," KOREA SCIENCE & ART FORUM, Vol. 15, pp. 233-246, 2014.

Authors

Il-Hwan Kim received the B.S degrees in Computer Science and Engineering for Dong-Seoul University Korea, in 2016 respectively He is currently a M.S Student in the Department of Computer Science and Engineering, Soongsil University. He is interested in Image AI technique, Deep Learning, Public Domain, Rights Management Information

Deok-Gi Hong received the B.S. degrees in Computer Science and Engineering from Hoseo University, Korea, in 2012 and 2017, respectively He is currently a M.S. student in the Department of Computer Science and Engineering, Soongsil University. He is interested in Copyright Technology and Intelligence Information Technology.

Jae-Keun Kim received the B.S. degree in Computer Science and Engineering from Soongsil University, Korea, in 2018 respectively. He is currently a M.S Student in the Department of Computer Science and Engineering, Soongsil University. He is interested include Embedded Systems, DRM(Digital Right Management), Deep Learning, and Object manipulation.

Young-Mo Kim received his Ph.D degree in Computer Engineering from Daejeon University, Daejeon, Korea in 2011. He is currently adjunct professor in Soongsil University. He is also working on several standardization activities and national project. His research interests are security, computer forensics, DRM(Digital Right Management), fingerprint.

Seok-Yoon Kim received the B.S degree in electrical engineering from Seoul University in 1980. He received the M.S and Ph.D degrees in ECE from University of Texas at Austin, in 1990 and 1993, respectively. He is currently a Professor in the Department of Computer Science and Engineering, Soongsil University. He is interested in Computer Systems(Embedded Systems), VLSI/SoC, Design Automation and Copyright Protection Technology.