

텍스트 이미지 실험을 통한 이미지 저장용량의 절감방안

*윤정준 **이경준 ***정제창

*한양대학교 공과대학 융합전자공학부

*bearcubs@nate.com **kjlee888@naver.com ***ijeong@hanyang.ac.kr

The way of data reduction using text images

*Yun, Jung-Jun **Lee, Kyung-Jun ***Jeong, Je-Chang

*Department of Electronic Engineering, Colege of Engineering, Hanyang University

요약

고해상도 이미지는 거의 대부분 유용하지만 저장용량을 많이 차지한다는 점은 그 유용함에 대한 제약이다. 이를 위한 수 많은 용량절감 방식이 있지만 화질저하의 한계점은 육안으로 보기에 원본과 가장 비슷하게 보이는 지점이다. 텍스트 파일은 그 형태가 높은 수준으로 고정되어 있다는 점에 착안하여 육안으로 원본과 비슷하게 보이지 않는 지점까지 손실 저장하더라도 다른 텍스트들과 비교되는 지점까지 가능하다.

1. 서론

오늘날 영상의 생성량이 증가하고 있다. 증가하는 영상의 양만큼 이에 필요한 저장공간 또한 증가하고 있다. 확보할 수 있는 저장공간은 무한하지 않기 때문에 저장 용량을 줄이는 방법은 다른 의미로서의 저장공간의 확보라고 할 수 있다. 텍스트 이미지 파일을 이용한 실험을 통해 저장 용량의 절감 방안을 알아보려고 한다.

2. 준비과정

2.1. 텍스트 이미지의 준비.

실험에 사용할 텍스트 이미지 파일을 검색을 통하여 구하기가 쉽지 않았다. 알파벳 이미지 파일을 검색해보면 일부 알파벳만 검색 되는 경우도 있고 모든 알파벳이 있더라도 그 크기가 매우 작은 경우도 있었다. 플래쉬 카드 이미지나 개인이 작성한 캘러그래피 이미지들 또한 콘텐츠 생산자가 설정한 저작권 허용범위를 알 수 없었다.

워드프로세서를 실행하여 알파벳을 타이핑한 후 글자체 크기를 조정 한 후 512 by 512 사이즈로 스크린샷 저장을 얻은 텍스트 이미지들 실험에 사용하였다. 글자 색상은 검은색으로 설정하였으므로 이들 이미지의 RGB 매트릭스원소는 0이나 255값을 가지게 된다.

2.2. 텍스트 이미지의 축소.

이전 단계에서 얻은 이미지 파일을 MATLAB 프로그램에 각 이미지 파일에 해당하는 변수명으로 할당하였다. 할당된 이미지들의 크기를 축소하기 위하여 홀수행과 홀수열에 해당하는 원소만 선택한 행렬을 새롭게 생성하는 방식을 사용하였다. 새롭게 생성된 이미지는 기존의 이미지에 비해 절반으로 줄어든 행과 열을 가지는 이미지이다. 홀수행과 홀수열을 선택하는 방식 대신 홀수행과 짝수열을 선택하는 방식, 짝수행과 홀수열을 선택하는 방식 그리고 짝수행과 짝수열을 선택하

는 방식도 가능하다.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \rightarrow \begin{pmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{pmatrix}$$

그림 1 . 홀수행과 홀수열을 선택하여 축소
Fig. 1. The reduction by odd cumluns and odd rows

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix} \rightarrow \begin{pmatrix} a_{22} & a_{24} \\ a_{42} & a_{44} \end{pmatrix}$$

그림 2 . 짝수행과 짝수열을 선택하여 축소
Fig. 2. The reduction by even cumluns and even rows

또 다른 방법은 4개 중 하나를 선택하는 것이 아닌 4개의 평균값을 계산하여 0이나 255에 가까운 쪽으로 화소값을 결정하는 방식이 있다. 이번 실험에 사용되는 텍스트 이미지 파일은 0이나 255 두 가지 값만을 가지기 때문에 4개의 매트릭스 원소의 0의 개수가 없는 경우에서부터 4개인 경우까지 모두 5가지의 경우의 수가 있다. 0인 원소의 개수가 255인 원소의 개수보다 많은 경우에는 4개의 매트릭스 원소의 대푯값을 0으로 결정하고 0인 원소의 개수가 255인 원소의 개수보다 적을 때는 대푯값을 255로 결정하였다.

$$\begin{pmatrix} 0 & 0 \\ 255 & 0 \end{pmatrix} \rightarrow 0 \begin{pmatrix} 255 & 0 \\ 255 & 255 \end{pmatrix} \rightarrow 255$$

그림 3 . 많은 화소값을 선택하여 축소
Fig. 3. The reduction by mosf value

절반으로 줄어든 이미지를 또 다시 홀수행과 홀수열에 해당하는 원소만 선택한 행렬을 새롭게 생성하면 처음 스크릿샷을 통해 얻은 이미지에 비해 행과 열이 $\frac{1}{2}$ 로 줄어든 이미지를 얻을 수 있다. 한번 더 반복하여 시행하면 행과 열은 $\frac{1}{8}$ 로 줄어든 이미지를 얻고 계속 반복하면 $\frac{1}{16}$, $\frac{1}{32}$ 등등을 거쳐 $\frac{1}{512}$ 까지 행과 열이 줄어든 이미지까지 가능하다.

2.3. 축소된 텍스트 이미지의 확대.

축소된 이미지를 다시 확대하는 이유는 처음에 얻은 원본과 비교하기 위해서 해상도를 맞추기 위함이다. MATLAB의 내장함수인 imresize를 적용하여 행과 열이 $\frac{1}{2}$ 로 줄어든 이미지 파일은 2배 확대, $\frac{1}{4}$ 로 줄어든 이미지 파일은 4배 확대하고 이를 반복하여 $\frac{1}{512}$ 로 줄어든 이미지 파일은 512배 확대를 적용한다. imresize를 이용하여 확대를 하면 원본이 0이나 255값만을 가진 것과는 다르게 그 사이 값을 가진 화소가 나타나게 된다. 이 값들은 0보다 255값에 가까울 경우엔 255로 화소값을 변경하고 0에 가까울 경우엔 0으로 화소값을 변경하여 선명하게 하였다.

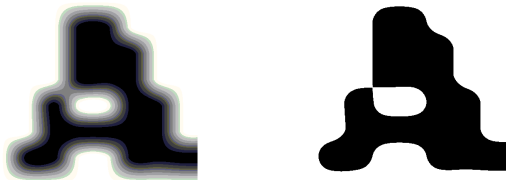


그림 4. 근접한 화소값을 선택하여 선명화
Fig. 4. The sharpening by the nearest value

3. 비교 및 결정 과정

3.1. 원본 이미지와의 유사성.

이미지의 비교 방법에는 여러 가지 방법이 있지만, 실험에 사용된 텍스트 이미지는 0과 255 두 개의 화소값만 가지기 때문에 화소값 빈도수 비교를 하게 될 경우 형태는 다르지만 255인 화소 개수가 비슷한 경우 텍스트 이미지 결정에 오류가 발생할 수 있다. 따라서 이 경우에는 원본 이미지 파일과 축소되었다 확대된 이미지 파일의 매트릭스 원소값을 일대일 비교하는 방식을 사용하였다. 원본 이미지 파일에 특정 위치의 매트릭스 원소값과 축소되었다 확대된 매트릭스 원소값이 같다는 것은 화소값의 분포는 물론 형태까지도 유사하다는 의미가 된다.

3.2. 다른 원본 이미지와의 비교.

축소되었다 확대된 이미지는 해당되는 이미지와의 형태 유사성도 중요하지만 해당되지 않는 텍스트 이미지들과 형태 불일치도 중요하다.

해당되는 이미지와 형태가 유사하지만 해당되지 않는 이미지와도 형태가 해당되는 텍스트 이미지보다 더 유사할 경우 다른 텍스트로 결정될 수 있기 때문이다. 따라서 축소되었다 확대된 이미지는 해당되는 텍스트를 제외한 나머지 51개의 텍스트 파일과도 비교하여야 한다. 해당되는 텍스트와의 유사성이 나머지 51개의 텍스트와의 유사성보다 클 경우에만 해당되는 텍스트로 결정될 수 있다.

4. 비교 및 결정 결과

4.1. 텍스트 유사성의 한계점.

축소되었다 확대된 이미지는 행과 열이 $\frac{1}{2}$ 로 줄어든 이미지, $\frac{1}{4}$ 로 줄어든 이미지에서 $\frac{1}{512}$ 로 줄어든 이미지까지 9가지가 있다. $\frac{1}{32}$ 로 줄어든 이미지까지는 해당되는 텍스트와의 유사성이 가장 높았다.

표 1 화소값일치율 (행과 열 1/32 축소)
Table 1 The pixel value equality rate(1/32 scale)

	해당텍스트 화소값일치율	그 외 최고 화소값일치율
홀수행 홀수열 선택 축소	89.71 %	79.61 %
홀수행 짝수열 선택 축소	72.53 %	65.11 %
짝수행 홀수열 선택 축소	89.20 %	79.85 %
짝수행 짝수열 선택 축소	88.75 %	78.69 %
대푯값 선택 축소(동수 백색)	91.40 %	80.91 %
대푯값 선택 축소(동수 흑색)	93.26 %	79.53 %

(대문자 A, 소수점 셋째자리에서 반올림)

$\frac{1}{64}$ 로 줄어든 이미지부터는 해당되지 않는 텍스트와의 유사성이 더 높은 경우와 낮은 경우가 혼재되어 나타났다.

표 2 화소일치율 (행과 열 1/64 축소)
Table 2 The pixel value equality rate(1/64 scale)

	해당텍스트 화소값일치율	그 외 최고 화소값일치율
홀수행 홀수열 선택 축소	83.13 %	78.45 %
홀수행 짝수열 선택 축소	79.38 %	78.75 %
짝수행 홀수열 선택 축소	52.95 %	51.22 %
짝수행 짝수열 선택 축소	77.99 %	79.38 %
대푯값 선택 축소(동수 백색)	83.45 %	83.89 %
대푯값 선택 축소(동수 흑색)	86.72 %	81.80 %

(대문자 A, 소수점 셋째자리에서 반올림)

$\frac{1}{28}$ 로 줄어든 이미지부터는 해당되지 않는 텍스트와의 유사성이 모든 경우에서 높아 텍스트 결정에 적합하지 않았다.

표 3 화소일치율 (행과 열 1/128 축소)
Table 3 The pixel value equality rate(1/128 scale)

	해당텍스트 화소값일치율	그 외 최고 화소값일치율
홀수행 홀수열 선택 축소	62.73 %	66.45 %
홀수행 짝수열 선택 축소	65.39 %	68.73 %
짝수행 홀수열 선택 축소	29.09 %	30.69 %
짝수행 짝수열 선택 축소	64.40 %	68.67 %
대푯값 선택 축소(동수 백색)	74.30 %	85.72 %
대푯값 선택 축소(동수 흑색)	70.39 %	66.82 %

(대문자 A, 소수점 셋째자리에서 반올림)

4.2. 용량 절감.

다른 텍스트 이미지들과 구분되는 가장 축소된 이미지는 행과 열이 $\frac{1}{32}$ 로 줄어든 이미지이다. 행과 열이 $\frac{1}{64}$ 로 줄어든 이미지 파일은 화소의 개수가 $\frac{1}{32} \times \frac{1}{32} = \frac{1}{1024}$ 로 줄어든다. 따라서 용량도 $\frac{1}{1024}$ 으로 줄어들게 되며, $\frac{1}{1024}$ 은 원본 용량의 0.097%에 해당한다.

5. 향후 과제

5.1. 추정할 수 있는 원본 이미지의 확대.

실험에 지금까지 사용된 텍스트 이미지 파일은 알파벳 대소문자 52 가지이다. 한글은 이번 실험에 포함되지 않았다. 텍스트 이미지 파일의 특성을 더욱 세밀히 파악하여 이번엔 포함되지 않은 텍스트들도 포함한 실험이 이루어진다면 적용범위가 더욱 넓어질 것이다.

5.2. 특수기호와 구두점 등.

특수기호는 알파벳과 형태가 유사한 경우가 종종있다. 구두점 중 쉼표와 마침표 콜론과 세미콜론은 그 형태가 매우 유사하여 축소 이후 확대된 이미지에서 구별할 수 있는 수준에서의 용량 절감이 미비할 수도 있다. 이런 형태적 유사성은 여러 문장에서 문맥 패턴을 확보하여 결정하는 것이 형태적 유사성을 극복할 수 있는 방안 중 하나가 될 것이라 추측한다.

5.3. 향상된 축소, 확대 알고리즘 발견과 적용

이번 실험에 향상된 알고리즘이 적용된다면 $\frac{1}{64}$ 로 줄어든 경우에

서도 해당 텍스트가 다른 텍스트보다 낮은 일치율을 보이는 경우들을 제거할 수 있을 것이다.

향상된 축소 알고리즘과 확대 알고리즘은 축소과정에서 더욱 원본에 유사한 형태를 가질 수 있고 확대과정에서도 원본에 더욱 근접한 형태를 얻을 수 있다. 원본 보존이나 추정에 성능이 뛰어난 알고리즘의 적용은 이미지의 용량 절감의 폭을 넓혀줄 것이다.

감사의 글

이 논문은 2015년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(NRF-2015R1A2A2A01006004)

참고문헌

- [1]SuHeeLee, GuangHoCha, Performance Evaluation of Dimension Reduction Techniques for the Efficient Content-based Image Retrieval, The HCI Society of Korea, 711-716, 2005.1
- [2]HeeJeong Wi, KyHyun Um, An Extracting and Indexing Scheme of Compressed Medical Images, Korea Multimedia Society, 328-331, 2000.5
- [3]Won-jung Kim, Mi-hwa Jang, The Avata Construction System for Image Lossless Scaling, The Korea Institute of Information and Communication Engineering, 181-189, 2002.4
- [4]Rafael C. Gonzalez, Richade E. Woods, *Digital Image Processing Third ed*, Pearson Education, Inc., Upper Saddle River, New Jersey 07258, 2010
- [5]Rafael C. Gonzalez, Richade E. Woods, Steven L. Eddins, translated by Hyun-Jung You and Tae-Woo Kim, *Digital Image Processing using MATLAB*, ITC, 85-3 3rd floor YeokChon-Dong EunPyung-Gu Seoul, 2004
- [6]Anaseair McAndrew, Jung-Hua Wang, Chun-Shun Tseng, translated by Tae-Hyo Kim and Young-Man Kwon and Myung-Guen Jeon, INTRODUCTION TO DIGITAL IMAGE PROCESSING with MATLAB, CENGAGE Learning Inc, Suite 1801 Seokyo Tower Building 353-1, 22 Seokyo-Dong Mapo-Gu Seoul 121-837 Korea, 2011
- [7]Chung-Won Kim, Hyun-sook Jung, Hui-Duk Yang, *Digital Image Processing*, Saeng Neung Inc., 143, Gwanginsa-gil, Paju-si, Gyeonggi-do 10881, 2016
- [8]Stormy Attawa, translated by Young-Gu Yu, Byung-Wan Jo, Seung-Jae Moon, *A Practical INTRODUCTION TO PROGRAMING AND PROBLEM SOLVING*, Ajjin, 201 Han-mi Building, 50, Hakdong-ro 20-gil, Gangnam-gu, Seoul 06113, 2012