

# 딥 러닝을 이용한 화재 감지 알고리즘

뉴엔 만 둥\*, 노 승 환<sup>o</sup>

## Smoke Detection Algorithm Using Deep Learning

Nguyen Manh Dung\*, Soonghwan Ro<sup>o</sup>

요 약

화재는 신체 상해 및 재산상 손해를 초래하는 치명적인 사고이다. 따라서 신속한 화재 탐지 및 경고는 화재 피해를 줄이는 가장 좋은 방법이다. 조기 화재 발생에서 연기를 감지하는 것은 화재 경보 시스템에서 가장 중요한 단서이다. 본 논문에서는 카메라 감시 시스템과 영상 처리 기술에 의한 자동 연기 감지 알고리즘을 제안하고자 한다. 이 알고리즘에서는 연기를 감지하고 움직이는 물체로 추적 한 다음 컨볼루션 신경 회로망(CNN : Convolutional Neural Network)을 사용한 캐스케이드(cascade) 분류 모델을 사용하여 연기와 연기가 없는 물체를 구분한다. 본 연구의 결과는 기존의 연구와 비교했을 때 잘못된 경보를 줄이는 것뿐만 아니라 연기를 탐지하는데 좋은 결과를 보여준다.

**Key Words** : smoke detection, deep learning, CNN(Convolutional Neural Network), Alexnet CNN, GMM

### ABSTRACT

Fire is a fatal accident that causes personal injury and property damage. Therefore, rapid fire detection and warning are the best ways to reduce fire damage. Detecting smoke from an early fire is the most important clue in a fire alarm system. In this paper, we propose an automatic smoke detection algorithm based on camera surveillance system and image processing technology. In this algorithm, smoke is detected and traced to a moving object, and then a cascade classification model using CNN (Convolutional Neural Network) is used to distinguish smoke from non-smoke objects. The results of this study show good results in detecting smoke as well as reducing false alarms when compared with existing studies.

### I. 개요 및 관련 연구

화재는 많은 인명 피해와 재산 손실이 발생시킨다. 그러나 조기에 화재 발생이 감지가 된다면, 대부분의 화재는 사전에 예방하거나 발생하더라도 손실을 줄일 수 있다. 따라서 자동 화재 경보 시스템은 화재 예방에 매우 중요하다.

화재 발생 초기에 연기를 감지하는 것은 화재를 감지하는 가장 중요한 단서가 되므로, 화재 초기에 연기를 감지하는 것은 매우 중요하다. 기존의 연기 감지

시스템에서는 센서를 많이 이용하였다. 그러나 연기 센서를 사용하는 시스템의 정확도를 높이기 위해서는 센서를 조밀하게 배치해야 하며, 특히 실외 환경과 같은 넓은 지역이면 시스템 설치 자체가 어려울 수 있다.

최근 디지털 카메라는 보안 감시 분야에서 급속도로 발전하고 있다. 보안 카메라의 장점은 센서에 비해서 쉽게 설치를 할 수 있으며 개방된 넓은 지역을 모니터링 할 수 있다. 특히 최근에는 감시 카메라 시스템과 영상 분석 기술을 이용한 연기 감지 시스템이 센서 기반 감지 시스템을 대체하고 있는 추세이다.

\* First Author : Kongju National University, manhdungbk45@gmail.com, 학생회원

<sup>o</sup> Corresponding Author : Kongju National University, rosh@kongju.ac.kr, 종신회원

논문번호 : KICS2017-06-169, Received June 8, 2017; Revised June 13, 2017; Accepted July 20, 2017

연기를 감지하기 위한 다양한 영상 처리 알고리즘이 제안되었으며, 대부분의 제안된 알고리즘은 연기를 움직이는 물체로 간주하고, 연기가 발생할 때 배경의 외관을 변경한다고 가정한다<sup>2,3,5,6</sup>. 이러한 알고리즘의 첫 번째 단계로 배경이 변화하는 것을 탐지하는 것을 연기가 발생하는 후보 영역으로 감지하고 연기가 없는 정지된 물체를 제거하는 방법이 주로 사용된다. 배경이 변화하는 것을 탐지하는 가장 효과적인 기술로는 배경 모델링<sup>11</sup>, 배경 제거 및 광 흐름(optical flow) 방법 등이 있다.

배경이 변화하는 것을 감지하는 것은 단지 연기가 나는 후보 영역을 탐지 하는데 도움이 되지만 인간, 차량 또는 배경 조명 변화와 같은 움직이는 물체와 연기를 구분 할 수는 없다. 따라서 탐지된 객체가 연기라는 것을 검증하기 위해서는 추가적으로 분석 단계가 필요하다.

색은 연기를 분류하는 데 사용되는 일반적인 특성이며<sup>2,3,5,6</sup>, 연기의 색은 회색, 밝은 회색, 흰색 그리고 어두운 회색이 될 수 있다. 그러나 실제 환경에는 비슷한 색을 갖는 많은 물체가 있으며, 그리고 어떤 경우에는 연기가 반투명하여 배경색에 영향을 받을 수도 있다. 따라서 색은 연기를 감지하는 신뢰할 만한 단서가 될 수 없다.

그리고 연기 영역의 크기 변화<sup>21</sup>, 연기 윤곽의 거친 정도<sup>4,8</sup>, 연기 영역의 증기<sup>8</sup>, 상승 영역의 검출<sup>8</sup> 등과 같은 방법들이 연기를 감지할 때 잘못된 경보를 제거하기 위해 제안되었지만, 그 들 모두 여전히 특정 상황에서 잘못된 경보를 유발할 수 있다.

연기를 구분하는 또 다른 방법은 웨이블릿(wavelet) 기반 분석이다<sup>4,5</sup>. 연기가 발생하면 배경 화면의 전체 또는 일부분이 연기에 의해 흐려질 수 있다. 이것은 연기 영역의 경계에서 고주파 성분과 선명도를 떨어뜨린다. 웨이블릿 에너지의 감소를 계산하는 것은 연기를 감지하는데 매우 중요한 단서이다. 그러나 이러한 특성이 항상 정확하지는 않으며, 예를 들어 연기가 매끄러운 배경의 경계를 증가시키거나, 부드럽고 넓은 면적을 갖는 연기가 배경의 선명도와 결계를 감소시킬 수 있다.

최근에는 로컬 화상 특징(HOG(Histogram of Oriented Gradients), SIFT(Scale Invariant Feature Transform)) 기반의 화상 분류 알고리즘을 사용하여 Bag of Visual Words를 구성하고 통계 분류기를 실행하여 많은 수의 객체 범주로 화상을 분류한다<sup>7</sup>. 이러한 접근법은 좋은 결과를 보여 주므로 연기와 연기가 없는 대상을 분류하는 방법으로 사용될 수 있다. 그러

나 분류기는 visual word에 의존하며, 객체의 컨텍스트를 표현할 때 패치(patch) 간의 공간적 관계를 무시하고 배경 정보를 혼동하며, 이 알고리즘은 시각적 또는 규모적인 관점에서의 불변성이 충분히 시험되지 않았다. 기계 학습(machine learning)에 있어서 딥 러닝(Deep Learning)은 새로운 경향이며, 컨벌루션 신경 회로망(CNN : Convolutional Neural Network)<sup>19</sup>이라는 화상 학습 알고리즘은 화상 분류를 포함한 컴퓨터 비전에서 최근 많은 성과를 보여주고 있다. 그러나 이 방법의 단점은 화상 화면에서 물체의 위치를 지정할 수 없으며, 계산 비용이 매우 높다는 것이다. 그러나 후보 연기의 위치를 확인하기 위해서 배경을 제거하거나 고속의 하드웨어 가속 장치를 사용하여 계산 속도를 향상시킴으로써 이러한 문제를 해결할 수 있다. 또한 연기가 없는 피사체를 대부분 제거하기 위해서 고속의 캐스케이드 모델(cascade model)을 여러 개 사용하고, 다음에 최종적으로 연기를 인식하기 위해 딥 러닝 분류기를 사용함으로써 계산 시간을 줄일 수 있다.

본 논문에서는 빠르고 안정적인 연기 감지 알고리즘을 제안하며, 2장에서 알고리즘과 구현에 대해서 설명하고, 3장에서는 알고리즘의 정확성과 성능에 대한 실험적 평가를 제시하고 보여준다. 그리고 마지막으로 4장에서 결론과 향후 연구에 대해서 논의한다.

## II. 비디오 기반의 연기 감지

그림 1은 본 연구에서 제안한 비디오 기반의 연기 감지 알고리즘 흐름 도를 보여준다. 제안된 알고리즘은 세 단계로 구성되며, 첫 번째 단계에서는 후보 연기영역을 감지하며, 두 번째 단계에서는 연기영역과 아닌 영역으로 구분하고, 세 번째 단계에서는 최종 결정을 위한 시간 분석(temporal analysis)을 한다.

연기 영역의 감지 : 제안된 알고리즘에서는 가우시안 혼합 모델링 (GMM)<sup>11</sup>을 사용하여 배경 픽셀의 변화를 감지하고, 연결된 픽셀들을 클러스터링 하여 후보 연기 영역으로 분류한다.

연기 분류기(classifier) : 연기와 아닌 영역을 분류하기 위하여 캐스케이드 모델(cascade model)이라고 하는 분류기의 집합이 사용된다. 각 분류기는 하나의 계층이며, 후보 연기 객체는 캐스케이드 모델의 모든 계층을 통과하면 실제 연기로 분류된다. 캐스케이드 모델의 상위 단계에서는 색상, 임의의 크기 변화, 경계 에너지 등을 이용하여 연기 영역이 아닌 영역을 제거한다. 그러나 이 때 오직 연기가 아닌 객체만 제거되도록 임계값을 선택해야 한다. 이 임계값으로 인해 잘

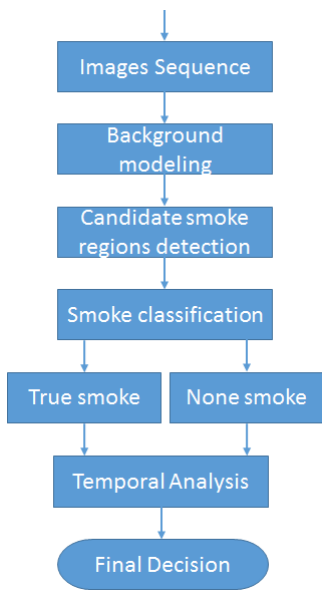


그림 1. 비디오 기반 연기 감지 알고리즘의 흐름도  
Fig. 1. Video based Smoke detection flow

못된 정보가 많이 발생할 수 있지만, 분류기의 각 계층을 통과 후에 오류는 감소하게 된다. 캐스케이드 모델의 최종 단계는 연기 영역을 검증하기 위한 딥 러닝 화상 분류기이다. 시간 분석(temporal analysis) 단계는 최종 결정의 신뢰성을 높이기 위해 연기 신호의 시간적 흐름을 분석하는 단계이다.

### 2.1 후보 연기 영역의 감지

연기가 발생하면 배경 화상의 모양이 바뀐다. 따라서 배경 화상의 변화를 감지함으로써 연기가 발생했다는 것을 감지할 수 있다. 연기 감지를 위해서 많은 알고리즘이 있으며, GMM(Mixture of Gaussian Background Modeling)<sup>[1]</sup>은 가장 많이 사용되는 알고리즘이다. GMM에도 여전히 부분적 또는 전체적인 빛의 변화(illumination change)에 따른 문제가 있지만, 매우 효율적으로 사용되고 있다. 따라서 GMM을 적용하여 후보 연기 영역을 탐지하고 나머지 문제는 이 후의 처리 절차에 의해 해결될 것이다.

가장 먼저 GMM은 화상 픽셀을 배경 픽셀(background pixel) 및 전경 픽셀(foreground pixel)의 2개의 클래스로 분류한다. 그 후에 연결된 전경 픽셀들을 블롭(blob)으로 묶을 것이고, 각 블롭은 하나의 후보 연기 영역 일 것이다. 그림 2에서는 후보 연기 영역을 추출하는 단계를 보여주며, 하나 또는 여러 개의 후보 연기 영역이 감지 될 수 있다.

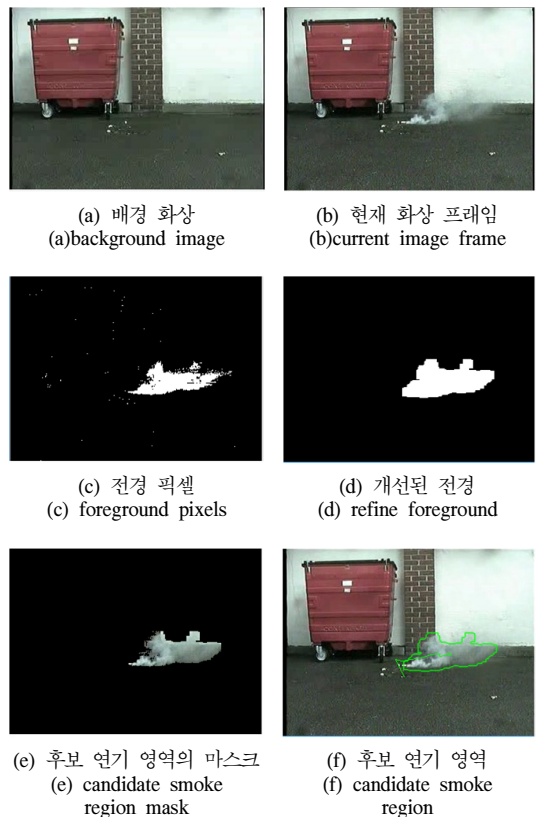


그림 2. 후보 연기 영역 감지  
Fig. 2. Candidate smoke region detection

### 2.2 연기 분류를 위한 캐스케이드 모델

이 모델은 연기 분류기 집합이며 각 계층은 하나의 분류기이다. 각 연기 영역은 이 모델의 모든 계층을 통과했을 때에만 실제 연기로 분류된다. 그림 1은 모델 아키텍처를 보여주며, 그림 3은 연기 분류를 하기 위한 캐스케이드 모델을 보여준다.

#### 2.2.1 색상 분류

일반적으로 연기는 어두운 회색, 회색, 밝은 회색, 흰색이다. 따라서 다음 조건을 만족하면 하나의 화상 픽셀은 연기 픽셀로 분류된다.

$$C_{max} - C_{min} < th_C \tag{1}$$

$$I_{min} < I < I_{max} \tag{2}$$

식 (1)에서  $C_{max}$ ,  $C_{min}$ 은 화상 픽셀의 (R, G, B)의 최대 값과 최소 값이며,  $th_C$ 는 임계 값이며, 실험에서  $th_C$ 의 범위가 5에서 25까지임을 보여준다.  $I$ 는 화상

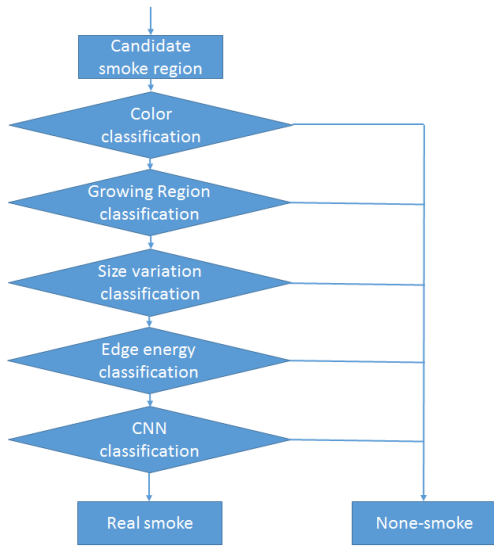


그림 3. 연기 분류의 캐스케이드 모델  
Fig. 3. Cascade model smoke classification

픽셀 강도(pixel intensity)를 나타내며, 연기 픽셀  $80 < I < 220$ 에 대해 그림 4와 그림 5는 연기가 없는 색상이 제거되기 전과 제거된 후의 연기 영역을 각각 보여준다.

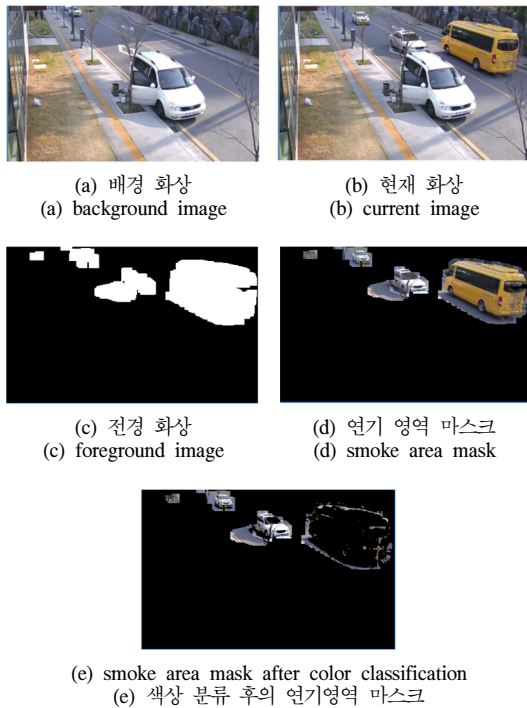


그림 4. 연기가 없는 화면에서의 연기 색상의 분할  
Fig. 4. smoke color segmentation in none smoke scene

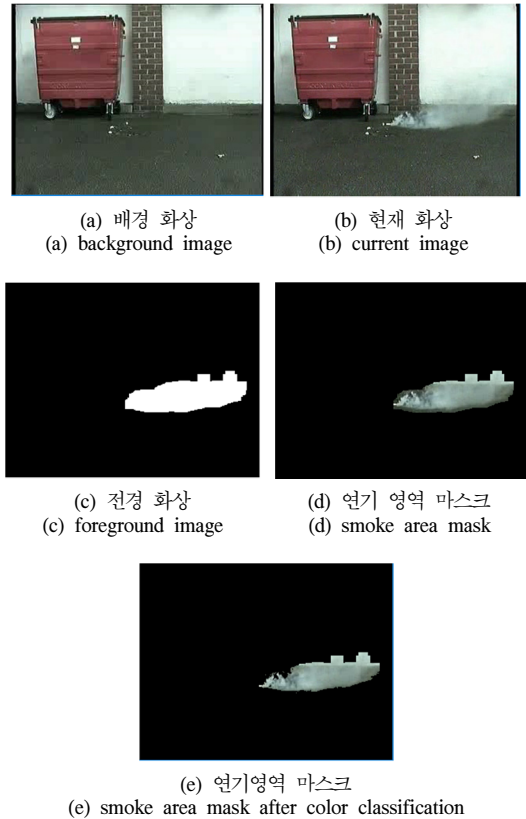


그림 5. 연기가 있는 화면에서의 연기 색상의 분할  
Fig. 5. smoke color segmentation in smoke scene

식 (3)은 색상 확률에 의한 후보 연기 영역이 사실 이 될 확률을 계산하기 위한 식이다.

$$F_C = n_C / S \tag{3}$$

식 (3)에서  $F_C$ 는 색상 확률을 나타내며,  $n_C$ 는 연기 색상 픽셀로 분류된 영역 내에 있는 픽셀의 수를 나타 내고,  $S$ 는 영역의 면적을 나타낸다. 연기 후보 영역에서 연기 영역이 아닌 것으로 분류된 영역은 제거될 것이다.  $th_{pc}$ 는 실험에 의해 결정되며, 이 단계에서  $th_{pc}$ 는 연기 영역이 아닌 영역은 배제되도록 한다.

$$F_C < th_{pc} \tag{4}$$

### 2.2.2 확산 영역 분류

초기에 연기가 발생하면 빠르게 확산하며, 일정 시 간 동안에 연기 영역이 지속적으로 확장된다. 그림 6 은 연기 영역이 확장하는 것을 보여준다.

확장하는 영역의 분석은 다음과 같다.  $n_g$ 를 후보



그림 6. 연기 영역의 확장  
Fig. 6. Growing of smoke region

연기 영역의 성장 인자라고 정의하고, 0으로 초기화한다. 두 연속 프레임  $t-1$ 과  $t$  사이의 연기 영역의 크기가 특정 임계 값  $th_{ds}$ 를 초과하면 성장 인자가 1만큼 증가한다. 이 때  $S_t, S_{t-1}$ 를 각각 시간  $t$ 와  $t-1$ 에서 후보 연기 영역의 크기라고 하면,

$$n_g = n_g + 1 \quad \text{if } S_t - S_{t-1} > th_{ds} \quad (5)$$

그  $n$ 을 분석 프레임의 수로 가정하면, 식 (6)을 사용하여 후보 연기 영역이 연기 확장 영역( $F_g$ )이 될 확률을 식 (5)와 같이 계산할 수 있다.

$$F_g = \frac{n_g}{n} \quad (6)$$

만일

$$F_g < th_g \quad (7)$$

이면 후보 연기 영역이 연기 영역이 아닌 것으로 분류되고 그 다음 절차에서 제외될 것이다.

### 2.2.3 크기 변화 분류

공기 흐름 때문에 연기 영역의 크기는 임의로 변한다. 그것은 또한 연기 영역과 연기가 아닌 영역을 구분하는 좋은 특징이 될 수 있다. 크기 변화 특성을 사용하여 연기 영역과 연기가 아닌 영역을 분류하기 위해 다음과 같은 파라미터를 계산해야 한다.

시간  $t$ 에서 연기 영역의 크기 변화  $\Delta S$ 는 식 (8)과 같다.

$$\Delta S_t = |S_t - S_{t-1}| \quad (8)$$

정규화된 영역의 크기 변화  $\Delta A$ 는 식 (9)와 같다.

$$\Delta A_t = \frac{\Delta S_t}{S_t} \quad (9)$$

시간  $t$ 에서 최근  $n$  프레임에서의  $\Delta S$ 에 대한 표준 편차는 식 (10)과 같다.

$$\Delta S^{t_{std}} = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (\Delta S_{t-1} - \overline{\Delta S})^2} \quad (10)$$

만일 식 (11)과 식 (12) 조건을 만족하면 후보 연기 영역은 다음 절차를 위해서 통과된다.

$$\Delta A_t > th_{\Delta A} \quad (11)$$

$$\Delta S^{t_{std}} > th_{\Delta S} \quad (12)$$

### 2.2.4 경계 에너지 분류

연기가 발생하면 배경 화상이 흐려지므로 연기 영역 화상 픽셀의 경계가 선명하지 않게 되므로 경계의 크기가 감소한다. 연기 영역의 경계 크기가 감소하는 것을 확인하기 위해 잃어버린 경계 픽셀의 수  $n_{lose}$ 와 얻은 경계 픽셀 수  $n_{gain}$ 을 추정한다. 연기의 경우에  $n_{gain}$ 은  $n_{lose}$ 보다 작다.

$n_{gain}$ 과  $n_{lose}$ 를 추정하기 위해서 첫 번째로 식 (13)~(15)를 사용하여 배경 화상 픽셀에 대한 경사도 크기(gradient magnitude)를 예측한다.

$$dx_{x,y}^{Bt} = B_{x+1,y}^t - B_{x-1,y}^t \quad (13)$$

$$dy_{x,y}^{Bt} = B_{x,y+1}^t - B_{x,y-1}^t \quad (14)$$

$$G_{x,y}^{Bt} = \sqrt{dx_{x,y}^{Bt \ 2} + dy_{x,y}^{Bt \ 2}} \quad (15)$$

식 (13)~(15)에서  $G_{x,y}^{It}, dx_{x,y}^{It}, dy_{x,y}^{It}, I_{x,y}^{It}$ 는  $(x, y)$  지점에서 배경 화상에 대한 경사도 크기, 수직 경사도, 수평 경사도, 그리고 화상의 강도(intensity)를 각각 나타낸다.

마찬가지로 현재의 화상 픽셀에 대한 경사도 크기는 식 (16)~식 (18)와 같다.

$$dx_{x,y}^{It} = I_{x+1,y}^t - I_{x-1,y}^t \quad (16)$$

$$dy_{x,y}^{It} = I_{x,y+1}^t - I_{x,y-1}^t \quad (17)$$

$$G_{x,y}^{It} = \sqrt{dx_{x,y}^{It \ 2} + dy_{x,y}^{It \ 2}} \quad (18)$$

식 (16)~식(18)에서  $G_{x,y}^{It}, dx_{x,y}^{It}, dy_{x,y}^{It}, I_{x,y}^{It}$  은 (x, y) 지점에서 현재 프레임에 대한 경사도 크기, 수직 경사도, 수평 경사도, 그리고 화상의 강도 (intensity)를 각각 나타낸다.

만일 식 (19)를 만족하면 하나의 픽셀은 손실된 경계 크기 픽셀(lost edge magnitude pixel)로 간주되거나

$$G_{x,y}^{Bt} - G_{x,y}^{It} > th_{mag} \quad (19)$$

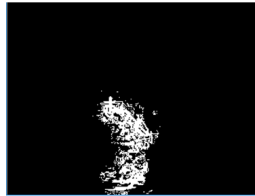


(a) 배경 화상  
(a) background image

(b) 현재 화상  
(b) current image



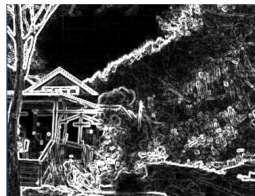
(c) 배경 화상 경계  
(c) background image edge



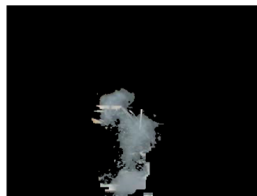
(d) 현재 화상 경계  
(d) current image edge



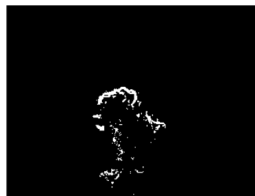
(e) 전경 화상  
(e) foreground image



(f) 연기 마스크  
(f) smoke mask



(g) 손실 경계 크기 픽셀  
(g) lost edge magnitude pixels



(h) 이득 경계 크기 픽셀  
(h) gained edge magnitude pixels

그림 7. 연기 영역에 대해 손실 된 경계 크기 픽셀 맵과 얻어진 경계 크기 픽셀의 맵

Fig. 7. Lost edge magnitude pixels map and gained edge magnitude pixels map for true smoke region

식 (20)을 만족하면 얻어진 경계 크기 픽셀(gained edge magnitude pixel)로 간주된다.

$$G_{x,y}^{It} - G_{x,y}^{Bt} > th_{mag} \quad (20)$$

식 (19), (20)에서  $th_{mag}$ 는 잡음을 제거하기 위해 사용되는 임계값이다.

그림 7은 연기 영역에 대해 손실 된 경계 크기 픽셀 맵과 얻어진 경계 크기 픽셀의 맵을 보여 주며, 얻어진 경계 크기 픽셀 수가 손실 된 경계 크기 픽셀 수보다 훨씬 적다는 것을 쉽게 알 수 있다. 이러한 특성을 이용하여 식 (21) 조건을 만족한다면 연기 영역을 식별 할 수 있다.

$$\frac{n_{gain}}{n_{lose}} < th_e \quad (21)$$

식 (21)에서  $th_e$ 은 결정 임계값이다.

### 2.2.5 컨벌루션 신경 회로망(CNN : Convolutional Neural Network)을 이용한 딥 러닝 화상 분류

캐스케이드 모델의 최종 계층은 딥 러닝 알고리즘인 CNN을 사용하여 연기 영역과 연기가 아닌 영역을 분류하는 것이다. 최근 이러한 프레임 워크(frame work)<sup>[9,10]</sup>가 자연 화상을 하위 객체 범주로 분류하는데 매우 좋은 결과를 보여주었다.

CNN은 유형 분류를 위한 기계 학습 알고리즘의 한 분야이다. 기계 학습을 사용하는 유형 분류는 학습(training) 및 예측(prediction)의 2 단계로 이루어진다. 학습 단계에서는 화상과 해당 클래스 레이블로 구성된 데이터 집합을 사용하여 알고리즘을 학습한다. 예측 단계에서는 보지 않았던 화상의 레이블을 예측하기 위해 학습 된 모델을 사용한다.

CNN은 입력 계층(input layer), 컨벌루션 계층(convolution layer), 풀링 계층(pooling layer) 및 완전 연결 계층(fully-connected layer)으로 구성되었다. 일반적으로 CNN은 이러한 계층들의 시퀀스이며 각 계층은 서로 다른 기능을 통해 활성화 볼륨(volume of activation)을 다른 볼륨으로 변환한다.

- 입력 계층 : 화상
- 컨벌루션 계층 : 이 계층은 화상을 공간적으로 이동시킬 수 있는 학습 가능한 필터 세트에 의해 구성된 필터 항목과 입력 화상 사이의 내적(product)을 계

산한다.

- 풀링(pooling) 계층 : 풀링 계층은 공간 차원을 따라 다운 샘플링 연산(down sampling operation)을 수행한다.
- 완전 연결 계층 : 이 계층에서는 클래스 점수를 계산한다.
- ReLU(Rectified Linear Unit) : 이 계층에서는 0 지점에서  $\max(0, x)$  임계치리(thresholding)와 같은 요소 단위 활성화 함수를 적용한다. 이 계층에서는 볼륨의 크기가 변경되지 않는다.

가장 간단한 컨벌루션 신경 망은 입력 계층(화상)로 시작해서 컨벌루션 계층과 풀링 계층의 시퀀스가 이어지고, 완전 연결 계층으로 끝난다. 일반적으로 컨벌루션 계층은 하나의 ReLU 활성화 함수 계층이 뒤 따른다. 컨벌루션, 풀링 및 ReLU 계층은 학습 가능한 특징 추출기(feature extractor)의 역할을 하지만, 완전 연결된 계층은 기계 학습 분류기의 역할을 한다. 또한 네트워크의 초기 계층은 화상의 일반 패턴을 인코딩(encoding)하고 나중의 계층은 화상의 세부 패턴을 인코딩한다. 컨벌루션 계층과 완전 연결된 계층에만 가중치가 있으며, 이 가중치는 훈련 단계(training phase)에서 학습이 된다. CNN에 대한 자세한 내용은 [14]에 설명되어 있다. 그림 8은 CNN 구조의 간단한 예를 보여준다.

컨벌루션 신경 회로망은 많은 양의 데이터 세트와 계산 시간이 필요하므로 다른 카테고리의 훈련 된 모델을 재교육하거나 업데이트하고자 할 때 문제가 발생할 수 있다. 전달 학습(transfer learning)<sup>[10-13]</sup>은 이러한 문제들을 해결할 수 있다. 망을 처음부터 교육하는 대신, 전달 학습은 다른 데이터 세트에서 숙련된 모델을 활용하고 새로운 분류자를 학습하기 위해 적응시킨다.

전달 학습에는 다음과 같은 두 가지 접근법이 있다.

- 훈련 된 모델을 고정된 특징 추출기(feature extractor)로 사용 : 이 접근법에서는 훈련된 모델에서 마지막으로 완전 연결 계층을 제거하고 나머

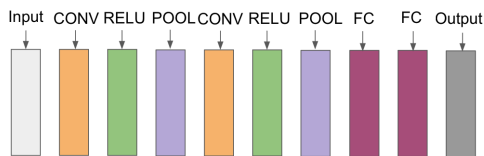


그림 8. 간단한 CNN 구조  
Fig. 8. A simple CNN architecture

지 계층의 가중치를 고정한 다음, 나머지 계층의 출력에 대한 머신 러닝 분류기를 학습(train)시킨다.

- 훈련 된 모델의 미세 조정 : 이 접근법에서는 역 전파(back propagation)를 계속하여 새 데이터 세트에서 훈련 된 모델을 미세 조정한다. 전체 망을 미세 조정하거나 일부 계층을 고정시킬 수 있다.

BVLC CaffeNet 모델은[9]에서 설명한 Alexnet 모델의 복사본으로 약간의 차이가 있다. 이 모델은 1000 개의 객체 범주를 구분하기 위해서 수백만 개의 화상을 갖는 Image Net Database<sup>[16]</sup>에서 훈련되었다. 본 논문에서는 이 모델을 사용하고 모델을 훈련하기 위해 미세 조정 접근법을 사용한다.

그림 9는 5개의 컨벌루션 계층으로 구성된 훈련 된 모델의 원래 아키텍처를 보여주며, 5개의 계층 뒤에는 2개의 완전 연결 계층(fully-connected layer)과 최종 1,000개 방향의 소프트맥스(1000-way softmax) 분류 계층이 뒤 따른다.

CNN의 분류 계층은 ImageNet 데이터 세트에 대한 1,000 개 클래스 확률을 계산한다. 미세 조정 절차를 하기 위해서는 이 계층을 수정해야 한다. 새로운 분류 계층은 연기와 연기가 아닌 두 가지 다른 부류만을 갖는 화상 데이터 세트를 사용하는 역 전파(back propagation) 알고리즘을 사용하여 처음부터 훈련을 받는다. 화상 데이터 세트에서 연기가 없는 화상은 사람, 차량 또는 단순히 배경 화상이다. 본 논문에서 훈련을 위해서 사용된 데이터 세트에는 10,000개의 연기 화상과 10,000개의 연기가 아닌 화상, 그리고 평가를 하기 위한 2,000 개의 연기 화상과 2,000 개의 연기가 아닌 화상이 사용되었다. 연기가 아닌 객체가 화상은 PETA 데이터 세트의 보행자 화상, 차량의 Cars 데이터 세트 및 배경 및 기타 움직이는 물체의 PASCAL 데이터 세트와 같은 다양한 소스에서 수집되었다. 또한 IVS 프로젝트<sup>[17]</sup>에서 기록 된 감시 영상으로부터 수동으로 연기가 아닌 객체를 분류하였다.

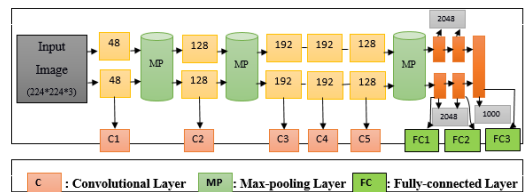


그림 9. Alexnet CNN 모델 구조  
Fig. 9. Alexnet CNN model architecture

그리고 연기 객체 화상은 IVS 프로젝트<sup>[17]</sup>, YouTube의 연기 비디오 및 인터넷에서의 연기 비디오 (<http://signal.ee.bilkent.edu.tr/VisiFire>)에서 기록된 비디오에서 수동으로 세그먼트하였다.

학습을 위해서 0.01의 학습율(learning rate)로 미세 조정을 시작하고 2,000회 반복마다 학습율을 10배로 낮췄다. 이미 훈련된 CNN 가중치가 상대적으로 양호하다는 가정 하에 미세 조정 된 CNN 가중치에 대해 더 작은 학습율을 사용하였다. 본 실험에서 필터 파라미터(filter parameter)가 빨리 크게 왜곡되지 않게 하기 위해서 학습 속도와 학습 속도 감쇄를 매우 작게 유지하였다. 연기 발생 여부에 대한 최종 결정의 신뢰성을 높이기 위해 일정 시간 동안 화상 프레임들의 이력을 유지한다. 예로 가장 최근 N개의 프레임에서 연기 객체를 포함하는 프레임 수를 Nsmoke라고 가정한다. 이 때 Nsmoke와 N 사이의 비율이 특정 임계값을 초과하면 화재가 발생한 것으로 간주하고 모든 사람에게 화재 경보를 발령한다. 본 실험에서는 분석 프레임 수 N을 20 프레임으로 설정하고 임계값은 0.8로 정했다.

실험에서 최적화 과정은 최대 50,000 회 실행되었으며, 그림 10의 훈련 곡선은 훈련 된 모델이 97.2%의 정확도를 보이며, 훈련된 모델의 false positive 확률은 1.8%, false negative 확률은 1.2%를 나타냈다.

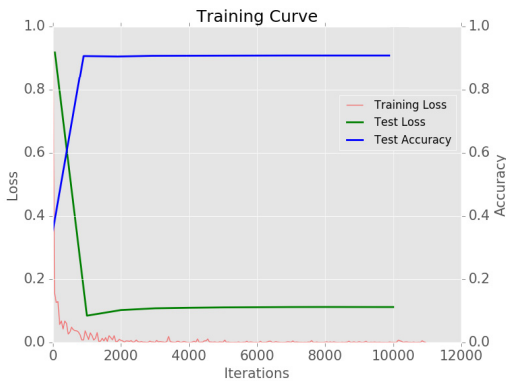


그림 10. 훈련 곡선  
Fig. 10. Training curve

### III. 테스트 및 평가

본 논문에서 VS2013과 오픈 소스 라이브러리인 OpenCV 2.4.10, NVIDIA CUDA SDK, Caffe Deep Learning을 기반으로 제안한 알고리즘을 구현 하였다.



(a) 배경의 낮은 색상 대비와 (b) 연기와 유사한 이동 물체  
경계 정보 부족  
(a) low background contrast (b) similar moving object  
and poor with smoke edge information

그림 11. 연기 감지가 어려운 경우  
Fig. 11. hard case of smoke detection

알고리즘은 Intel® core™ i7-4790, windows10 및 NVIDIA GeForce GTX 750에서 실행되었으며, IVS Project<sup>[17]</sup>의 데이터 세트, 개방형 데이터베이스 (<http://signal.ee.bilkent.edu.tr/VisiFire/>) 및 인터넷 비디오가 사용되었다.

연기를 감지하기 위해서는 높은 임계값을 정해야 하지만, 높은 임계값으로 인해서 연기를 감지하지 못하거나, 잘 못 감지하는 경우가 발생한다. 그림 11은 높은 임계값으로 인해서 연기를 잘 못 감지하는 경우를 보여준다. 그림 11(a)는 배경 화상의 색상대비(contrast)가 낮고 경계 정보가 좋지 않아서 연기 감지가 어려운 경우이다. 이러한 경우에 에너지 분석 기술을 사용하면 연기를 분류하기가 어렵다. 그림 11(b)는 움직이는 물체가 연기와 유사하게 밝은 회색의 매끄러운 표면을 갖는 경우를 보여주며, 이러한 경우에도 잘못된 경보를 발생시킬 수 있는 문제가 발생한다.

제안 된 알고리즘에서는 마치 연기처럼 이동하는 것처럼 보이는 물체의 확률을 제거하기 위한 임계값을 선택하였으며, 우수한 성능의 CNN 분류기를 사용함으로써, 제안된 알고리즘의 감지율이 향상된 것과 잘못된 감지가 감소하였다는 것을 알 수 있었다. 그림 12는 실험의 일부 예를 보여주며, 제안된 알고리즘은 연기가 포함 된 짧은 비디오에서 연기를 감지하고 잘못된 경보는 발생하지 않았다. 그림 12에서 빨간 경계 영역은 연기 영역으로 분류되며, 녹색 경계영역은 연기가 아닌 영역으로 분류된다.

표 1은 상세한 실험 결과를 보여준다. 표 1에서는 평가를 위해서 15 개의 비디오를 사용했으며, 이 가운데 10개는 연기 화상을 포함하며, 5개는 연기 화상을 포함하지 않는다. 프레임 간 분석에서는 약간의 False Positive(화재가 아닌데 화재로 분석하는 경우)가 발생했으며, 그 이유는 본 알고리즘이 연기 지역의 특징을 분석하는 데 약간의 시간이 필요하기 때문이었다. 그





그림 12. 연기 감지 결과  
Fig. 12. Smoke detection results

표 1. 실험결과  
Table 1. Experiment result

Video No.	Description	Number of Frame	Frame by fram Analysis(%)		Smoke Alarm
			False Positive	False Negative	
01	Smoke	1,499	6.5	3.8	Yes
02	Smoke	599	8.5	2.7	Yes
03	Smoke	898	2.6	2.3	Yes
04	Smoke	689	1.3	0.5	Yes
05	Smoke	574	4.5	4.9	Yes
06	Smoke	3,129	1.7	3.3	Yes
07	Smoke	1,820	5.8	4.1	Yes
08	Smoke	1,767	4.3	3.5	Yes
09	Smoke	901	6.1	1.2	Yes
10	Smoke	716	0.7	0.7	Yes
11	Non-Smoke	296	NA	0.0	No
12	Non-Smoke	1,942	NA	1.6	No
13	Non-Smoke	298	NA	0.0	No
14	Non-Smoke	340	NA	0.0	No
15	Non-Smoke	4,000	NA	0.0	No
Average			3.55	1.99	

그러나 False Negative(화재를 화재가 아닌 경우로 분석하는 경우)는 CNN 분류의 신뢰성 때문에 매우 낮았다. 대부분의 False Negative는 연기가 배경에 반투명할 때 발생했다. 특히 인간 또는 차량과 같이 규칙적으로 움직이는 물체에서 발생하는 False Negative의율이 매우 낮았다. 표 1의 결과에서 보여주듯이 시간 분석(temporal analysis)후의 최종 결정은 매우 정확하다는 것을 보여준다.

NVIDIA 컴퓨팅 가속화 하드웨어를 사용함으로써 제안된 알고리즘은 매우 빠른 처리 성능을 보여주었으며, 초당 40 프레임을 처리 할 수 있으므로 실시간 응용에 적합하다는 것을 알 수 있었다.

#### IV. 결 론

화재를 자동으로 감지하는 것은 매우 중요하다. 신속하게 화재를 경보함으로써 사람들이 대피할 수 있을 뿐만 아니라 인명과 재산 피해를 줄일 수 있다.

본 논문에서 조기 화재 경보 시스템을 위한 비디오 기반 연기 감지 알고리즘에 대한 알고리즘을 제안하였다. 배경 모델링에 기반을 둔 알고리즘은 배경의 변화를 탐지하며, 연기 영역의 분류를 위해서는 캐스케이드 모델을 사용하였다. 캐스케이드 모델은 다중 특징(multi-feature)의 장점을 보다 쉽게 활용할 수 있게 하며, 화재 분류에서 딥 러닝 컨벌루션 신경 회로망은 연기를 감지하는데 높은 정확성을 제공한다. 제안된 알고리즘은 다른 알고리즘과 비교할 때 크게 개선되었다. 실험 결과 빠르고 신뢰할 수 있는 결과를 보여주며, 실시간 감시 시스템에 가장 적합한 것을 보여주었다.

추후 연구로는 제안된 모델의 정확성을 향상시킬 수 있는 좋은 특징을 찾는 것이다. 또한 향상된 화재 경보 시스템을 개발하기 위해서 화염 탐지 모델로 확장을 할 예정이다.

#### References

[1] C. Stauffer and W. E. L. Grimson, "Adaptive background mixture models for real-time tracking," *IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognition*, vol. 2, 1999.

[2] T. X. Tung and J.-M. Kim, "An effective four-stage smoke-detection algorithm using video images for early fire-alarm systems," *Fire Safety J.*, vol. 46, no. 5, pp. 276-282, Jul.

2011.

[3] W. Zheng, W. Xingang, A. Wenchuan, and C. Jianfeng, "Target-tracking based early fire smoke detection in video," *ICIG '09*, pp. 172-176, Sept. 2009.

[4] B. U. Toreyin, Y. Dedeoglu, and A. Enis Cetin, "Contour based smoke detection in video using wavelets," *14th Eur. Signal Process. Conf.*, pp. 1-5, Sept. 2006.

[5] C.-Y. Lee, C.-T. Lin, C.-T. Hong, and M.-T. Su, "Smoke detection using spatial and temporal analysis," *Int. J. Innovative Comput., Inf. and Contr.*, vol. 8, no. 7(A), Jul. 2012.

[6] B. C. Ko, J. Y. Kwak, and J. Y. Nam, "Wildfire smoke detection using temporal-spatial features and random forest classifiers," *Opt. Eng.*, vol. 51, no. 1, Feb. 2012.

[7] A. Bosch, A. Zisserman, and X. Munoz, "Image classification using random forests and ferns," *ICCV 2007*, pp. 1-8, Oct. 2007.

[8] A. Genovese, R. D. Labati, V. Piuri, and F. Scotti, "Wildfire smoke detection using computational intelligence techniques," *CIMSA*, pp. 1-6, Sept. 2011.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet classification with deep convolutional neural networks," *NIPS'12*, Dec. 2012.

[10] I. J. Kim, "Deep Learning : 기계학습의 새로운 트렌드," *KICS Inf. and Commun. Mag.*, vol. 31, no. 11, pp. 52-57, Nov. 2014.

[11] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," *2014 IEEE CVPR*, pp. 1717-1724, Jun. 2014.

[12] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," *UTLW'11 Proc. 2011 Int. Conf. Unsupervised and Transfer Learning workshop*, vol. 27, pp. 17-37, 2012.

[13] A. K. Reyes, J. C. Caicedo, and J. E. Camargo, "Fine-tuning deep convolutional networks for plant recognition," *Working*

*Notes of CLEF 2015-Conf. and Labs of the Evaluation Forum CLEF 2015*, Sept. 2015.

[14] Convolutional Neural Networks, <http://cs231n.github.io/convolutional-networks/>

[15] BAIR/BVLC CaffeNet Model, [https://github.com/BVLC/caffe/tree/master/models/bvlc\\_reference\\_caffenet](https://github.com/BVLC/caffe/tree/master/models/bvlc_reference_caffenet)

[16] ImageNet, <http://www.image-net.org/>

[17] Intelligent video surveillance system (IVS) Project, <http://ivstech.co.kr/?lang=en>

뉴엔 만 등 (Nguyen Manh Dung)



2005년 7월 : Hanoi Univ. of Science and Tech. Dept. of Electronic and Telecommunication(공학사)  
 2009년 2월 : 공주대학교 정보통신과(공학석사)  
 2017년 3월~현재 : 공주대학교 정보통신과 박사과정

<관심분야> 영상처리, 임베디드 시스템

노 승 환 (Soonghwan Ro)



1987년 8월 : 고려대학교 전자공학과 졸업(공학사)  
 1989년 8월 : 고려대학교 전자공학과 석사(공학석사)  
 1993년 8월 : 고려대학교 전자공학과 박사(공학박사)  
 1994년 3월~현재 : 공주대학교 교수

<관심분야> 이동성 관리프로토콜, 임베디드 시스템