

심층신경망을 리용한 실시간차번호판령역검출의 한가지 방법

리충일, 김순돌, 최철

경애하는 최고령도자 김정은동지께서는 다음과 같이 말씀하시였다.

《우리는 가까운 앞날에 전반적인 과학기술분야에서 세계를 디디고 올라설수 있다는
배심을 가지고 첨단돌파의 기적들을 련이어 창조하여야 합니다.》

심층신경망(DNN: Deep Neural Network)은 화상인식뿐만아니라 대상검출에서도 높은 성
능을 보여주고있으며 그것의 층깊이는 보다 더 깊어지고있다.

Fast R-CNN에 이어 Faster R-CNN 등 심층신경망에 의한 대상검출방법[1]들이 차례로
제안되면서 대상검출의 속도와 정확도가 높아지고있다.

그러나 심층신경망방법들을 대상검출의 한 종류에 속하는 차번호판령역검출에 적용
하는 경우 실시간성보장에서 문제가 제기된다.

론문에서는 실시간차번호판령역검출을 위한 심층신경망의 구조를 설계하고 그에 기
초한 차번호판령역검출방법을 제안하였다.

1. 문 제 설 정

차번호인식은 차번호판령역추출과 차번호문자인식으로 구성되며 여기서 차번호판령
역검출의 성능을 높이는것이 선차적인 문제로 나선다.

차번호문자인식의 성능이 아무리 좋다고 해도 차번호판령역검출에서 실패하면 차번
호인식자체가 실패로 끝난다.

또한 주행차량들에 대한 차번호인식체계에서는 실시간성보장이 중요한 요구로 제
기된다.

차번호인식에서 실시간성을 보장하자면 적어도 100ms이내에 인식이 진행되어야
한다.

그러나 심층신경망을 리용한 선행한 대상검출방법들은 GPU(도형처리장치)를 리용하
는 경우에는 고속이지만 CPU(연산처리장치)를 리용하는 경우에는 실시간성을 보장하지
못한다.

현실적으로 GPU를 내장한 컴퓨터가 많지 못하고 장치비용이 높기때문에 CPU에서
실시간성을 보장하는것이 중요한 문제로 나선다.

GoogLeNet의 Inception, ResNet[2] 등 망의 성능을 유지하면서 파라미터용량과 계산량
을 줄이기 위해 1×1 중첩층을 리용하여 심층신경망의 층수를 깊게 하는 방법들이 제안되
고있다.

론문에서는 선행방법들의 우점을 반영하여 CPU에서도 차번호판령역검출의 실시간성
을 보장할수 있는 한가지 심층신경망방법을 제안하였다.

2. 차번호영역검출을 위한 심층신경망의 구조설계

1) 중첩연산의 분해에 의한 망파라미터의 축소

중첩연산의 원리에 의해 입력특징지도에 5×5 려파기를 한번 중첩시킨것은 3×3 려파기를 두번 중첩시킨것과 같게 된다. 이때 층수는 2배 늘어나지만 파라미터수는 25로부터 18로 줄어들게 된다. 결과 학습을 효율적으로 진행하게 하고 학습문제를 계층적으로 분해할수 있게 한다.

이로부터 최신심층신경망들은 크기가 5이상인 려파기들을 3×3 려파기들로 교체하고 층수를 늘이는 방향으로 설계되고있다.[2]

한편 3×3 중첩연산전에 1×1 중첩연산을 삽입하는 방법, 3×3 중첩연산을 3×1 중첩연산과 1×3 중첩연산으로 분해하는 방법 등 파라미터크기를 줄이는 여러가지 방법들이 제안되고있다.

본문에서는 중첩층의 려파기조를 2개의 려파기조로 분해하여 파라미터크기를 축소하는 방법을 제안하였다.

입력특징지도 F 의 너비, 높이, 통로수를 (W, H, M) , 출력특징지도 G 의 너비, 높이, 통로수를 (W, H, N) 이라고 하면 $k \times k$ 려파기들의 조 K 의 크기는 $((k \times k) \times M) \times N$ 이다.

이때 걸음값(stride)이 1이고 채우기값(padding)이 $[k/2]$ 일 때 표준중첩연산은 다음과 같다.

$$G_{k,l,n} = \sum_{i,j,m} K_{i,j,m,n} \cdot F_{k+i-1,l+j-1,m} \quad (1)$$

식 (1)의 표준중첩연산의 계산량(적합연산의 개수)은 $k \cdot k \cdot W \cdot H \cdot M \cdot N$ 이다.

려파기조 K 를 그림과 같이 분리할수 있다.

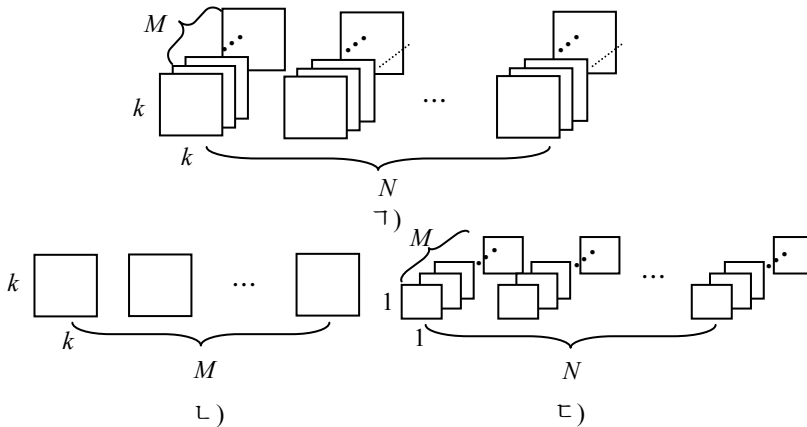


그림. 려파기조의 분리

$\Gamma)$ 표준 $k \times k$ 려파기조 K , $\Upsilon)$ 입력통로방향의 $k \times k$ 려파기조 K' ,

$\Upsilon)$ 출력통로방향의 1×1 려파기조 K''

즉 입력특징지도에 대하여 그림 1의 $\Gamma)$ 와 같은 려파기조 K 를 적용할대신에 $\Upsilon)$ 의 려파기조 K' 를 먼저 적용하고 얻어진 중간특징지도 $G'_{k,l,m}$ 에 대하여 $\Upsilon)$ 의 려파기조 K'' 를 적용한다.

식 (1)을 중첩연산기호 \otimes 를 리용하여 표기하면 다음과 같다.

$$G_n = \sum_m K_{m,n} \otimes F_m \quad (2)$$

입력특징지도 F 는 입력특징지도화상 F_m 들의 모임 ($F = \{F_m | m = \overline{0, M-1}\}$)이고 출력특징지도 G 는 출력특징지도화상 G_n 들의 모임 ($G = \{G_n | n = \overline{0, N-1}\}$)이며

$$K = \{K_{m,n} | m = \overline{0, M-1}, n = \overline{0, N-1}\}$$

은 $k \times k$ 려파기조이다.

그림 1과 같이 려파기조 K 를 입력통로방향의 려파기조 K' 와 출력통로방향의 려파기조 K'' 로 분해하면 다음과 같다.

$$G'_m = K'_m \otimes F_m \quad (3)$$

$$G_n = \sum_m K''_{m,n} \otimes G'_m \quad (4)$$

K' 는 M 개의 $k \times k$ 려파기조이며 K'' 는 M 개의 1×1 려파기조가 N 개 들어있는 려파기조 묶음이며 $M \times N$ 행렬로 실현된다.

식 (3), (4)에 의한 중첩연산을 진행하는 경우 계산량은 $k \cdot k \cdot M \cdot W \cdot H + M \cdot N \cdot W \cdot H$ 로 된다.

결국 파라메터 축소비율 R 는

$$R = \frac{k \cdot k \cdot M \cdot W \cdot H + M \cdot N \cdot W \cdot H}{k \cdot k \cdot M \cdot N \cdot W \cdot H} = \frac{1}{N} + \frac{1}{k^2} \quad (5)$$

이다.

따라서 $k=3$ 이고 N 이 128 또는 256과 같이 큰 경우 축소전에 비해 약 9배의 계산량 축소효과를 가진다.

2) 차번호령역검출을 위한 심층신경망의 구조설계

파라메터를 축소하기 전 망층들의 배치와 크기를 표 1에 보여주었다.

표 1. 축소전 망층들의 배치와 크기

층의 유형	입력특징지도의 크기	려파기조의 크기
conv1/mp2	448×448×3	3×3×3×16
conv2/mp2	224×224×16	3×3×16×32
conv3/mp2	112×112×32	3×3×32×64
conv4/mp2	56×56×64	3×3×64×128
conv5/mp2	28×28×128	3×3×128×256
conv6	14×14×256	3×3×256×256
conv7/mp2	14×14×256	3×3×256×512
conv8	7×7×512	3×3×512×1 024
conv9	7×7×1 024	1×1×20
avgpool	7×7×20=980	

파라메터를 축소한 후 망층들의 배치와 크기를 표 2에 보여주었다.

표 1, 2에서 mp2는 2×2 크기의 max-pooling연산을 의미하며 conv(id), conv(od)는 각각 입력통로방향, 출력통로방향의 려파기조에 의한 중첩연산을 의미한다.

표 2. 축소후 망층들의 배치와 크기

층의 유형	입력특징지도의 크기	려파기조의 크기
conv1/mp2	448×448×3	3×3×3×16
conv2(id)	224×224×16	3×3×16
conv3(od)/mp2	224×224×16	1×1×16×32
conv4(id)	112×112×32	3×3×32
conv5(od)/mp2	112×112×32	1×1×32×64
conv6(id)	56×56×64	3×3×64
conv6(od)/mp2	56×56×64	1×1×64×128
conv7(id)	28×28×128	3×3×128
conv8(od)/mp2	28×28×128	1×1×128×256
conv9(id)	14×14×256	3×3×256
conv10(od)	14×14×256	1×1×256×256
conv11(id)	14×14×256	3×3×256
conv12(od)/mp2	14×14×256	1×1×256×512
conv13(id)	7×7×512	3×3×512
conv14(od)	7×7×512	1×1×512×1 024
conv15(fc)	7×7×1 024	1×1×20
avgpool	7×7×20=980	

망의 출력은 마지막전련결층의 출력에 3×3 크기의 average pooling 연산을 거쳐 얻는다.

망의 출력은 대상검출층(DL: Detect Layer)을 거쳐 대상영역검출을 진행하며 이 층에서의 대상영역검출방법[3]은 다음과 같다.

대상검출층의 입력은 7×7×20 크기의 텐소르이고 출력특징지도 7×7의 매 요소에 4개의 고정4각형(default box)이 적용되며 검출대상클래스는 1개이기때문에

$$4 \times (\text{직4각형 자리표값}(4) + \text{클래스개수}(1)) = 20$$

의 예측이 진행된다.

4개의 고정4각형들의 크기 $Box(w, h)$ 는 입력화상을 448×448 크기로 정규화한 화상에서 출현하는 차번호판의 너비, 높이들을 참고하여 다음과 같이 정한다.

$$Box(w, h) = \{(37, 37), (49, 45), (72, 68), (93, 85)\}$$

3) 망의 학습

고정4각형의 개수를 N 이라고 할 때 대상추출층의 1개 특징요소에서의 손실함수는 선행방법[1]에서와 유사하게 다음과 같이 계산한다.

$$L(x, c, p, g) = \frac{1}{N} (L_{cls}(x, c) + L_{loc}(x, p, g)) \quad (6)$$

검출손실 함수

$$L_{loc}(x, p, g) = \frac{1}{N} \sum_{i \in Pos} \sum_{m \in \{cx, cy, w, h\}} x_{ij} S_{L1}(p_i^m - \hat{g}_j^m) \quad (7)$$

에서 p 는 예측4각형, g 는 정답4각형이며

$$\hat{g}_j^m = \{\hat{g}_j^{cx}, \hat{g}_j^{cy}, \hat{g}_j^w, \hat{g}_j^h\}$$

이다. 여기서

$$\begin{aligned}\hat{g}_j^{cx} &= (g_j^{cx} - d_i^{cx}) / d_i^w, \quad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy}) / d_i^h \\ \hat{g}_j^w &= \log(g_j^w / d_i^w), \quad \hat{g}_j^h = \log(g_j^h / d_i^h)\end{aligned}$$

이다.

$x_{ij} = \{1, 0\}$ 은 p_i 와 g_j 의 겹침면적이 렉값 $I_{ou}=0.5$ 보다 크면 1, 작으면 0이다. 그리고 S_{L1} 은 다음과 같다.

$$S_{L1}(x) = \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & |x| \geq 1 \end{cases} \quad (8)$$

클라스손실 함수는

$$L_{cls}(x, c) = - \sum_{i \in Pos} x_{ij} \log(c_i) - \sum_{i \in Neg} \log(c_i) \quad (9)$$

이다.

3. 실험 및 결과분석

제안방법의 유효성을 판정하기 위하여 1 920×1 080 해상도의 촬영기로 차번호판이 보이는 도로상의 차량을 촬영한 화상 1 200개를 준비하고 짝수번째 화상을 학습용, 홀수번째 화상을 검사용으로 리용하였다.

학습과 검사는 GTX-980이 장비된 Corei7 3.6GHz컴퓨터에서 Tensorflow프레임워크로 진행되었다.

4만회의 학습을 진행하여 얻은 모형들에 의한 성능평가를 표 3에 보여주었다.

표 3. 성능평가

분 류	화상모임	화상수/개	검출수/개	검출시간		검출률/%	모형의 크기/MB
				CPU/ms	GPU/ms		
파라미터	학습	600	590	300	75	98.3	27.5
축소전방법	검사	600	575	300	75	95.8	
제안방법	학습	600	590	70	25	98.5	3.2
	검사	600	574	70	25	96.1	

표 3은 제안방법이 파라미터축소전에 비해 인식정확도의 손실이 없이 기억용량을 8.5배 축소하는것과 동시에 인식시간을 4배 단축함으로써 100ms(CPU시간)이하의 실시간 차번호판검출을 보장한다는것을 보여준다.

맺 는 말

실시간차번호판인식을 보장하기 위한 심층신경망방법을 제안하고 실험을 통하여 그 유효성을 확증하였다.

참 고 문 헌

- [1] S. Len et al.; IEEE Transaction on Pattern Analysis & Machine Intelligence, 39, 6, 1137, 2017.
- [2] K. He et al.; Computer Vision and Pattern Recognition, 770, 2016.
- [3] W. Liu et al.; European Conference on Computer Vision, 21, 2016.

주체109(2020)년 8월 5일 원고접수

A Method of Real-time Vehicle License Plate Location Using Deep Neural Network

Ri Chung Il, Kim Sun Dol and Choe Chol

In this paper we designed a high-speed deep neural network using filter decomposition for real-time vehicle license plate recognition system on running vehicles and confirmed that the proposed method has a high speed of below 100ms on even CPU by experiments.

Keywords: deep neural network, license plate location, license plate recognition