일반논문 (Regular Paper)
방송공학회논문지 제26권 제2호, 2021년 3월 (JBE Vol. 26, No. 2, March 2021)
https://doi.org/10.5909/JBE.2021.26.2.184
ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

객체 검출을 위한 통계치 적응적인 선형 회귀 기반 객체 크기 예측

권 용 혜[a], 이 종 석[b], 심 동 규[b]‡

Object Size Prediction based on Statistics Adaptive Linear Regression for Object Detection

Yonghye Kwon[a], Jongseok Lee[b], and Donggyu Sim[b]‡

요 약

본 논문은 객체 검출 알고리즘을 위한 통계치 적응적인 선형 회귀 기반 객체 크기 예측 방법을 제안한다. 기존에 제안된 딥 러닝 기반 객체 검출 알고리즘 중 YOLOv2 및 YOLOv3은 객체의 크기를 예측하기 위하여 네트워크의 마지막 계층에 통계치 적응적인 지수 회귀 모델을 사용한다. 하지만, 지수 회귀 모델은 역전파 과정에서 지수 함수의 특성상 매우 큰 미분값을 네트워크의 파라미터로 전파시킬 수 있는 문제점이 있다. 따라서 본 논문에서는 미분 값의 발산 문제를 해결하기 위하여 객체 크기 예측을 위한 통계치 적응적인 선형 회귀 모델을 제안한다. 제안하는 통계치 적응적인 선형 회귀 모델은 딥러닝 네트워크의 마지막 계층에 사용되며, 학습 데이터셋에 존재하는 객체들의 크기에 대한 통계치를 이용하여 객체의 크기를 예측한다. 제안하는 방법의 성능 평가를 위하여 YOLOv3 tiny를 기반으로 제안하는 방법을 적용하여 재설계한 네트워크의 검출 성능과 YOLOv3 tiny의 검출 성능을 비교하였으며, 성능 비교를 위한 데이터셋으로는 UFPR-ALPR 데이터셋을 사용하였다. 실험을 통해 제안하는 방법의 우수성을 검증하였다.

Abstract

This paper proposes statistics adaptive linear regression-based object size prediction method for object detection. YOLOv2 and YOLOv3, which are typical deep learning-based object detection algorithms, designed the last layer of a network using statistics adaptive exponential regression model to predict the size of objects. However, an exponential regression model can propagate a high derivative of a loss function into all parameters in a network because of the property of an exponential function. We propose statistics adaptive linear regression layer to ease the gradient exploding problem of the exponential regression model. The proposed statistics adaptive linear regression model is used in the last layer of the network to predict the size of objects with statistics estimated from training dataset. We newly designed the network based on the YOLOv3tiny and it shows the higher performance compared to YOLOv3 tiny on the UFPR-ALPR dataset.

Keyword: Object Detection, Statistics Adaptive Linear Regression, YOLO

Copyright ⓒ 2021 Korean Institute of Broadcast and Media Engineers. All rights reserved.
“This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (http://creativecommons.org/licenses/by-nc-nd/3.0) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered.”

Ⅰ. 서 론

컴퓨터 비전 분야에서 객체 검출은 영상에서 특정 객체의 위치를 예측하고 종류를 분류하는 작업을 의미한다. 객체 검출은 얼굴 인식[1], 자율 주행[2], 차량 번호판 인식[3] 등 다양한 컴퓨터 비전 어플리케이션에서 요구되는 기술로 필요성이 높기 때문에 많은 연구가 진행되어 왔다[4]. 최근 convolutional neural networks (CNN)을 기반으로, 객체 인식, 객체 검출, 분할 등 다양한 컴퓨터 비전 작업에서 활발한 연구가 이루어지고 있으며, 높은 성능을 보이고 있다. 특히, CNN을 기반으로 한 객체 검출 알고리즘들이 대표적인 벤치마크 데이터 셋인 PASCAL visual object classes[5]와 MS-COCO[6]에서 기존에 제안된 머신러닝 기반의 검출 알고리즘 대비 크게 높은 성능을 보임에 따라 CNN을 기반으로 한 검출 알고리즘들이 활발히 연구되고 있다.

대표적인 CNN 기반의 검출 알고리즘으로는 R-CNN[7], Fast R-CNN[8], Faster R-CNN[9], YOLOv1[10], SSD[11], YOLOv2[12], YOLOv3[13] 등이 제안되었다. CNN 기반의 검출 알고리즘은 크게 2단계 검출 알고리즘, 1단계 검출 알고리즘으로 분류할 수 있다[4]. 2단계 검출 알고리즘은 두 단계의 과정을 거쳐 검출을 수행하며, 첫 단계에서는 객체가 있을 법한 관심 영역을 추출하고, 두번째 단계에서는 관심 영역으로부터 피처(feature)를 추출한다. 이를 이용하여 종류를 분류하고, 객체의 위치를 보정한다. 대표적인 2단계 검출 알고리즘으로 R-CNN, Fast R-CNN, Faster R-CNN 등이 있다. 2단계 검출 알고리즘은 높은 검출 성능을 보이나, 상대적으로 계산 복잡도가 높은 문제점이 있다. 1단계 검출 알고리즘은 2단계 검출 알고리즘과 달리 객체의 위치 검출과 종류의 분류를 동시에 수행하는 알고리즘이다. 대표적인 1단계 검출 알고리즘으로는 YOLOv1, YOLOv2, YOLOv3 등이 제안되었다. 1단계 검출 알고리즘은 2단계 검출 알고리즘과 비교하여 상대적으로 낮은 검출 성능을 보이나, 계산 복잡도가 크게 낮다. 이러한 특징에 기반하여 1단계 검출 알고리즘을 응용으로 하는 어플리케이션 개발과 관련 연구가 활발히 진행되고 있다[14][15][16]. 그 중 YOLOv1[10]은 객체를 검출함에 있어 학습 데이터셋으로부터 추정 가능한 어떠한 통계치도 이용하지 않고 객체를 검출하도록 설계되었다. YOLOv2[12]는 객체 검출의 성능 향상을 위하여 학습 데이터셋에 존재하는 객체의 너비와 높이 값의 통계치를 이용하였다. 이를 통해 YOLOv1대비 네트워크의 학습을 안정시킬 수 있었으며 검출 성능을 향상시킬 수 있었다. YOLOv3[13]는 YOLOv2와 같은 방법으로 통계치를 이용하여 객체의 크기를 예측하며, 추가적으로 스케일 변화에 강인한 객체 검출기를 설계하기 위하여 피쳐 피라미드 네트워크 구조를 바탕으로 설계되었다[17]. YOLOv2 및 YOLOv3는 객체의 크기를 예측하기 위하여 네트워크의 마지막 계층에 통계치 적응적인 지수 회귀 모델을 사용한다. 하지만, 지수 회귀 모델은 역전파 과정에서 지수 함수의 특성상 매우 큰 미분 값을 네트워크의 전체로 전파시킬 수 있는 단점이 있다.

본 논문에서는 미분 값 발산 문제를 완화하기 위하여 객체의 크기를 예측함에 있어 통계치 적응적인 선형 회귀 모델을 제안한다. 제안하는 선형 회귀 모델은 딥러닝 네트워크의 마지막 계층에 사용되며, 학습 데이터셋의 객체들의 너비와 높이에 대한 통계치를 이용하여 객체의 크기를 예측한다. 예측되는 객체의 너비 및 높이 값의 평균값과 표준 편차는 학습 데이터셋의 객체들의 너비 및 높이 값의 평균값과 표준 편차를 따르도록 학습된다. 예측되는 값의 평균 값과 표준 편차가 목푯값의 평균값과 표준편차를 따르도록 제약 조건을 부여함으로써 학습 과정을 보다 안정시킬 수 있으며, 최종적으로 검출 성능을 향상시킬 수 있다. 제안하는 방법의 성능을 평가하기 위해 자동차 번호판 데이터셋인 UFPR-ALPR[14]를 사용하였으며, 비교를 위해 YOLOv3-tiny[13][18] 백본 네트워크를 기반으로 YOLOv2 및 YOLOv3의 통계치 적응적인 지수 회귀 모델과 제안하는

a) 광운대학교 컴퓨터공학과(Dept. of Computer Engineering, Kwangwoon University)
b) ㈜마크에니(MarkAny)
‡ Corresponding Author : 심동규(Donggyu Sim)
E-mail: dgsim@kw.ac.kr
Tel: +82-2-940-5470
ORCID: http://orcid.org/0000-0002-2794-9932
※ 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터지원사업의 연구결과로 수행되었음(IITP-2021-2016-0-00288).
※ This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2021-2016-0-00288) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).
· Manuscript received December 17, 2020; Revised March 2, 2021, Accepted March 12, 2021.

통계치 적응적인 선형 회귀 모델을 사용하였을 때의 검출 성능을 비교한다. 실험을 통해 제안하는 방법과 기존 객체 크기 예측 방법의 average precision(AP)를 비교하였을 때, 보다 높은 성능을 보임을 입증한다.

본 논문의 구성은 다음과 같다. 2장에서 기존에 제안된 검출 알고리즘과 통계치를 이용한 객체 검출 알고리즘에 대하여 설명한다. 3장에서는 제안하는 통계치 적응적인 선형 회귀 모델 기반 객체 크기 예측 방법을 상세히 설명한다. 4장에서 기존 방법과 제안하는 방법의 실험적 성능을 평가하고, 5장에서 결론을 맺는다.

Ⅱ. 기존의 객체 검출 알고리즘

딥 러닝 및 CNN 기반의 객체 검출 알고리즘 연구 이전에 전통적인 객체 검출 알고리즘은 후보 영역 제안, 주요 피쳐 추출, 객체 분류로 크게 3가지 단계로 나뉘어 연구되었다[4]. 후보 영역 제안 기술의 목적은 영상에서 객체가 있을 법한 위치, 즉 관심영역을 제안하는 것이다. 다양한 크기를 갖는 객체들을 검출하기 위하여, 입력 영상은 다른 크기로 리사이즈 되거나, 다양한 크기를 갖는 윈도우를 이용하는 슬라이딩 윈도우 알고리즘이 이용되었다. 두 번째 단계에서 적용되는 기술은 주요 피쳐 추출 기술이다. 여기서 주요 피쳐란 영상내에서 객체를 표현하는 도구로써, 이동, 회전, 크기의 변화가 있더라도 해당 객체를 잘 표현하는 강인성을 갖는 피쳐를 의미한다. 객체 검출을 위해 사용되는 주요 피쳐 추출 기술로는 대표적으로 scale invariant feature transform (SIFT)[19], Haar[20], histogram of gradients (HOG)[21], speeded up robust features (SURF)[22]등이 제안되었다. 마지막으로, 객체 분류 기술의 목적은 주요 피쳐로부터 객체의 클래스를 분류하는 것이다. 객체 분류 기술로는 support vector machine (SVM)[23]이 흔히 이용되었다. 전통적인 객체 검출 알고리즘의 경우 주요 피쳐 추출 단계에 사용되는 피쳐 추출기의 설계 방법에 크게 초점을 맞추어 연구되었으며, 이는 연구자에 의하여 수동적으로 설계되었다. 그러나, 기존에 제안된 피쳐 추출기의 경우 주로 낮은 수준의 시각 단서를 수치화 할 수 있는 피쳐를 추출하도록 설계되었기 때문에 객체의 다양한 변화에 대해 강인성이 높은 피쳐를 추출하는데 한계가 있었다. 또한 각 단계가 부분적으로 설계되었으며, 각각 개별적으로 최적화되기 때문에 전역 최적해를 구하지 못하는 문제가 있었다. CNN 기반의 객체 인식 방법[24]이 큰 성공을 거두었고, 관련 연구[25][26][27]가 활발히 진행됨에 따라 이를 기반으로 한 객체 검출 방법 또한 연구[4] 및 개발되고 있다. CNN 기반의 객체 검출 알고리즘은 학습 데이터로부터 추론 과정과 역전파 알고리즘을 통해 최적의 피쳐 추출기를 학습하게 된다. 또한 객체 검출에 필요한 전체적인 작업 과정이 통합 최적화 (Joint Optimization)될 수 있다. 이러한 특징으로 CNN 기반의 객체 검출 알고리즘은 전통적인 객체 검출 알고리즘의 성능 대비 크게 향상된 결과를 보였다.

본 장의 2.1절에서는 기존 CNN 기반의 객체 검출 알고리즘에 대하여 설명한다. 2.2절에서는 기존의 통계치를 이용한 객체 크기 예측 알고리즘인 YOLOv2[12] 및 YOLOv3[13]에 관하여 자세히 설명한다.

1. CNN 기반 객체 검출 알고리즘

CNN 기반의 객체 검출 방법은 크게 2단계 검출 방법과 1단계 검출 방법으로 나뉜다[4]. 2단계 검출 방법은 먼저 첫 단계에서 객체들이 있을 법한 위치를 찾아 관심 영역을 추출하고, 첫 단계에서 추론해낸 후보 영역에 대해 추가적으로 CNN 기반의 분류기를 이용하여 물체들의 종류를 분류하고, 회귀 모델을 통해 보다 정확한 위치를 계산한다[7]. 대표적인 2단계 검출 알고리즘은 R-CNN[7], Fast R-CNN[8], Faster R-CNN[9]이 있다. 2단계 검출 방법은 전통적인 객체 검출 알고리즘보다 크게 높은 성능을 보이나, 시간 복잡도가 크게 높은 문제점이 있다. 이러한 문제로 실시간 객체 검출 알고리즘이 요구되는 어플리케이션에는 적용이 어려운 문제점이 있었다. 이에 따라 2단계 검출 방법보다 검출 성능은 비교적 낮으나, 낮은 시간 복잡도를 갖는 1단계 검출 방법이 제안되었다. 1단계 검출 방법은 객체의 위치 추정과 클래스 분류를 동시에 수행하는 방법을 의미한다. 대표적인 1단계 검출 방법은 YOLOv1[10], YOLOv2[12], YOLOv3[13]가 있다. YOLOv1 은 1단계 검출 알고리즘의 활발한 연구 및 개발을 이끈 알고리즘으로, 이미지를 그리드 단위로 나누어 객체를 검출하는 방법을 제안하였다. 그

러나, YOLOv1은 기존에 제안된 2단계 검출 알고리즘 보다 크게 낮은 성능을 보였다. YOLOv2, YOLOv3는 객체를 검출함에 있어 SSD[11]와 마찬가지로 앵커 박스 (Anchor Box)를 이용하였다. YOLOv2는 각각의 앵커 박스의 너비 및 높이 값을 정의하기 위한 방법으로, K-means 알고리즘을 통해 학습 데이터셋에 존재하는 객체들을 너비와 높이 값에 따라 K개의 클러스터로 분류하고, 각 클러스터 별로 정의되는 너비와 높이 값에 대한 산술 평균값을 앵커 박스의 너비와 높이 값으로 정의하였다. 이를 통해 YOLOv2는 검출 성능을 향상시킬 수 있었다. 그리고, YOLOv3는 스케일 변화에 강인한 검출기를 설계하기 위하여 피쳐 피라미드 네트워크 구조[17]를 기반으로 네트워크를 설계하여 검출 성능을 크게 향상시킬 수 있었다.

2. 통계치를 이용한 객체 검출 알고리즘

본 절에서는 통계치 적응적인 지수 회귀 모델 기반 객체 크기 예측 방법을 제안한 CNN 기반 객체 검출 알고리즘인 YOLOv2, YOLOv3에 대해 자세히 설명한다. YOLOv2, YOLOv3는 영상을 그리드 단위로 나누고, 나누어진 셀 별로, 객체를 검출할 수 있도록 설계되었다. YOLOv2, YOLOv3의 객체 크기 회귀 과정은 그림 1과 같다. 보다 쉬운 설명을 위해, 검출을 수행하는 피쳐맵의 사이즈 (채널 수, 너비, 높이)를 $(5+C, 2, 2)$로 가정한다. 이 때, C는 검출기에서 분류 가능한 클래스의 개수를 의미한다. 네트워크에 할당된 앵커 박스의 개수는 1개로 가정한다.

먼저, 입력 영상으로부터 컨볼루션 계층을 기반으로 설계된 CNN을 통해 입력 영상 대비 너비 및 높이의 크기가 n배 축소된 피쳐맵을 추출한다. 피쳐맵에서 (x, y)에 위치한 셀은 입력 영상을 기준으로 좌상단의 위치가 (nx, ny), 우하단의 위치가 $(n(x+1), n(y+1))$인 직사각형 패치 영역내에 중심 좌표가 위치하는 객체의 검출을 수행하도록

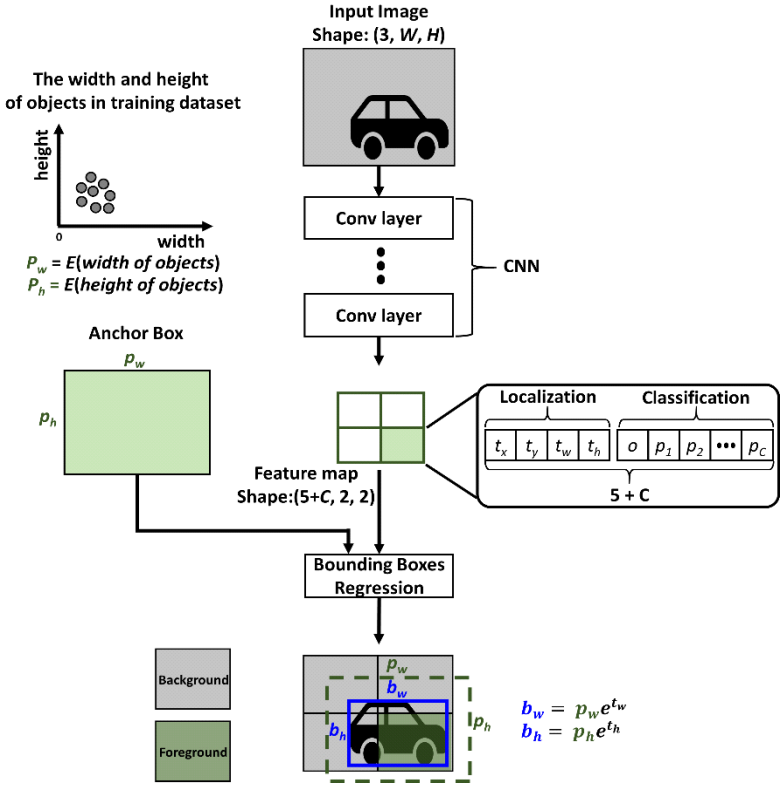


그림 1. YOLOv2, YOLOv3의 객체 크기 회귀 과정
Fig. 1. Object size regression process for YOLOv2 and YOLOv3

설계된다. 그림 1의 피쳐맵에서 색이 칠해지지 않은 셀은 해당 셀에서 객체가 검출되지 않음 (배경으로 판단)을 의미하며, 색이 칠해진 셀은 해당 셀에서 객체가 검출됨을 의미한다. 피쳐맵의 각 셀에서는 객체의 위치 추정과 분류를 수행하기 위한 변수들이 출력된다. 위치 추정을 위한 변수로는 객체의 중점을 검출하는데 사용되는 변수인 t_x, t_y와 객체의 너비와 높이를 예측하는데 사용되는 변수 t_w, t_h가 출력된다. 분류를 위한 변수로는 객체와 배경임을 판단하는 변수인 o와 해당 객체에 대한 클래스별 점수를 의미하는 변수인 $p_1, p_2, \cdots, p_c$가 예측된다. YOLOv2, YOLOv3는 객체의 너비 및 높이를 예측함에 있어 통계치 적응적인 지수 회귀 모델을 제안하였고, 해당 모델은 네트워크의 마지막 계층에 사용되었다. YOLOv2, YOLOv3에서 제안된 통계치 적응적인 지수 회귀 모델은 다음의 수식과 같다.

$$b_w = p_w e^{t_w} \tag{1}$$

$$b_h = p_h e^{t_h} \tag{2}$$

여기서, b_w와 b_h는 네트워크에서 최종적으로 예측되는 객체의 너비 및 높이를 의미한다. p_w와 p_h는 학습 단계 이전에 사전 정의되는 하이퍼 파라미터로써 각 파라미터는 학습 데이터셋에 존재하는 객체들의 너비와 높이의 통계치인 산술 평균값으로 정의되며, 이는 앵커 박스의 크기를 나타낸다. YOLOv2, YOLOv3는 사전 정의된 너비 p_w와 높이 p_h에 대하여 이에 곱해지는 스케일을 적응적으로 예측하도록 네트워크를 설계하였다. 학습 데이터셋에서 추정 가능한 통계치를 이용함으로써, YOLOv2는 학습 단계에서 네트워크의 학습 과정을 보다 안정시킬 수 있었으며, 이를 통해 객체 검출 성능을 향상시킬 수 있었다. 그러나, 객체의 크기를 예측함에 있어 기존에 제안된 통계치 적응적인 지수 회귀 모델은 지수 함수 특성상 역전파 과정에서 매우 큰 미분 값이 네트워크 전체에 전파될 수 있다. 이러한 수치적 안정성 문제를 완화하고자 YOLOv2 및 YOLOv3는 손실 함수를 계산함에 있어 지수 회귀 모델의 출력 단에서 손실 함수를 계산하지 않고, 지수 회귀 모델의 입력 단에서 손실 함수를 계산하고 오차를 전파함으로써 해당 문제를 완화시켰다. 이를 수식으로 나타내면 다음과 같다.

$$g_w = \ln(b_w / p_w) \tag{3}$$

$$g_h = \ln(b_h / p_h) \tag{4}$$

g_w와 g_h는 t_w와 t_h의 목표 값으로, 학습 단계에서 최종적으로 t_w와 g_w 간의 제곱 오차와 t_h와 g_h 간의 제곱 오차를 계산하고 오차를 전파한다.

본 논문에서는 지수 회귀 모델의 미분 값 발산 문제를 해결하기 위하여 객체의 크기를 예측하기 위한 통계치 적응적인 선형 회귀 모델을 제안한다. 제안하는 통계치 적응적인 선형 회귀 모델은 통계치 적응적인 지수 회귀 모델의 문제를 완화시킬 수 있으며, 통계치 적응적인 지수 회귀 모델 보다 안정적인 학습이 가능하며, 최종적으로 검출 성능을 향상시킬 수 있다. 다음 장에서는 제안하는 방법에 대해 자세히 설명한다.

Ⅲ. 통계치 적응적인 선형 회귀 모델 기반 객체 크기 예측 방법

제안하는 방법은 YOLOv2와 YOLOv3와 동일하게 객체의 너비와 높이를 예측함에 있어 학습 데이터셋에서 추정 가능한 통계치를 이용한다. 학습 데이터셋의 통계치를 추정하는 과정은 다음과 같다. 학습 데이터셋에 존재하는 객체들의 너비와 높이 값에 따라 객체를 K개의 클러스터로 분류한 후, 각 클러스터에 속하는 객체들의 너비와 높이 값의 각각의 산술 평균값을 추정한다. 본 논문에서 제안하는 통계치 적응적인 선형 회귀 모델은 각 클러스터에 속하는 객체들의 너비와 높이 값의 표준 편차를 추가적으로 추정한다. 그리고, 네트워크를 통해 예측되는 객체의 너비 및 높이 값의 평균값과 표준 편차가 학습 데이터셋에 존재하는 객체들의 너비 및 높이 값의 평균값과 표준 편차를 따르도록 설계하였다. 학습 데이터셋에서 추정 가능한 통계치를 이용하여 예측 값의 통계치에 대한 제약을 부여함으로써, 학습 단계에서 네트워크를 보다 안정시킬 수 있으며, 검출 성능을 향상시킬 수 있다.

본 장의 3.1절에서는 학습 데이터셋에 존재하는 객체들

의 너비와 높이로부터 객체 크기 예측을 위한 통계치 추정 방법에 대해 설명한다. 3.2절에서는 통계치 적응적인 선형 회귀 모델 기반 객체 크기 예측 방법에 대하여 설명한다.

1. 제안하는 통계치 추정 방법

YOLOv2, YOLOv3는 객체를 검출함에 있어 학습 데이터셋으로부터 추출 가능한 객체들의 너비 및 높이에 대한 산술 평균을 앵커 박스를 통해 사용하였다. 앵커 박스는 너비와 높이가 사전 정의되는 박스로, Faster R-CNN, SSD, YOLOv2, YOLOYv3등 대다수의 객체 검출 알고리즘들이 이를 이용하였다. 기존에 앵커 박스를 이용하는 CNN 기반의 객체 검출 알고리즘은 학습 단계 이전에 네트워크의 설계자가 검출기에 할당되는 앵커 박스의 개수 K를 정해야 하며, 각각의 앵커 박스의 너비 p_w와 높이 p_h가 정의되어야 한다. 제안하는 통계치 추정 방법은 다음과 같다. 먼저, 검출기에 할당될 앵커 박스의 개수 K를 정의한다. 그리고, 최적화 과정을 거쳐 객체들의 너비와 높이에 따라 학습 데이터셋에 존재하는 객체들을 K개의 클러스터로 분류한다. 여기서, 최적화 과정은 K-means 클러스터링 알고리즘의 최적화 과정과 동일하다. 최종적으로 각 클러스터에 속하는 객체들의 너비 값과 높이 값의 산술 평균 값과 표준 편차를 K개의 앵커 박스의 하이퍼 파라미터로써 추가적으로 할당한다.

2. 통계치 적응적인 선형 회귀 모델 기반 객체 크기 예측 방법

본 절에서는 통계치 적응적인 선형 회귀 모델 기반 객체 크기 예측 방법에 대해 설명한다. 제안하는 통계치 적응적인 선형 회귀 모델은 기존에 제안된 객체 검출 알고리즘인 YOLOv2, YOLOv3의 객체 크기 예측을 위하여 마지막 계층으로써 사용되는 통계치 적응적인 지수 회귀 모델을 대체할 수 있다. 기존에 제안되었던 통계치 적응적인 지수 회귀 모델의 경우, 학습 데이터셋에 존재하는 객체들의 너비와 높이의 통계치 중 하나인 평균값을 이용하여 객체의 크기를 예측하였다. 기존에 제안되었던 통계치 적응적인 지수 회귀 모델과 달리, 제안하는 통계치 적응적인 선형 회귀 모델은 학습 데이터셋에 존재하는 객체들의 너비와 높이의 통계치 중 평균 값뿐만 아니라 표준 편차를 함께 이용하여 객체의 크기를 예측한다. 제안하는 모델의 입력 값은 t_w와 t_h이고, 이는 식(1)과 식(2)에서 정의된 변수와 동일하다. 해당 모델로부터 최종적으로 예측되는 값은 객체의 너비 b_w와 높이 b_h로, 해당 값들의 각각의 평균값과 표준 편차는 사전 정의되는 평균값과 표준 편차를 따르도록 설계된다. 먼저, 학습 단계에서는 제안하는 통계치 적응적인 선형 회귀 모델의 입력 값인 t_w와 t_h에 대한 평균값과 표준 편차를 계산한다. 그런데, 학습 과정에서 역전파 과정을 거쳐 네트워크의 파라미터를 업데이트 하게 되는 경우, 네트워크의 아웃풋의 값이 계속해서 변하게 되며 이로 인하여 네트워크의 평균값과 표준편차를 정확하게 추정하는 것이 힘들게 된다. 본 연구에서는 학습 단계에서 t_w와 t_h에 대한 평균값과 표준 편차를 미니 배치 단위로 매번 계산하도록 학습 과정을 설계했다. 미니 배치 단위로 계산되는 t_w와 t_h에 대한 평균값과 표준 편차를 이용하여 t_w와 t_h를 평균값이 0이고, 표준 편차가 1인 확률 분포를 따르도록 정규화한다. 이를 정규화하는 수식은 다음과 같다.

$$\hat{t}_w = \frac{t_w - E(t_w)}{\sqrt{Var(t_w)}} \tag{5}$$

$$\hat{t}_h = \frac{t_h - E(t_h)}{\sqrt{Var(t_h)}} \tag{6}$$

여기서, $E(t_w)$와 $E(t_h)$는 미니 배치로부터 추정한 t_w와 t_h의 평균값이고, $\sqrt{Var(t_w)}$와 $\sqrt{Var(t_h)}$는 미니 배치로부터 추정한 t_w와 t_h의 표준 편차다. $\hat{t}_w$와 $\hat{t}_h$는 t_w와 t_h를 평균값이 0이고 표준 편차가 1인 확률 분포를 따르는 값으로 정규화한 값이다. 추론 단계에서는 학습 단계에서 계산되는 t_w와 t_h의 평균값과 표준 편차에 대한 이동 평균값을 이용하여 t_w와 t_h를 정규화한다. 학습 단계에서 어떠한 데이터가 입력되더라도 정규화 과정을 통해 t_w와 t_h는 평균이 0이고 표준 편차가 1인 확률 분포를 따르도록 정규화된다. 그리고, t_w와 t_h가 예측되는 객체의 너비와 높이 값의 평균값과 표준 편차가 사전 정의되는 평균값과

표준 편차를 따르도록 변환시킬 수 있다. 이를 위한 수식은 아래와 같다.

$$b_w = (\hat{t}_w \times \sqrt{Var(W)}) + E(W) \tag{7}$$

$$b_h = (\hat{t}_h \times \sqrt{Var(H)}) + E(H) \tag{8}$$

여기서, b_w와 b_h는 제안하는 통계치 적응적인 선형 회귀 모델을 통해 네트워크에서 최종적으로 예측되는 객체의 너비와 높이 값으로 각각의 평균이 $E(W)$, $E(H)$이고 표준 편차가 $\sqrt{Var(W)}$, $\sqrt{Var(H)}$인 확률 분포를 따르게 된다. W는 데이터셋에 존재하는 객체들의 너비 값에 대한 확률 변수이며, H는 데이터셋에 존재하는 객체들의 높이 값에 대한 확률 변수이다. 이때, $E(W)$와 $E(H)$는 수식(1), (2)의 p_w와 p_h와 동일한 의미를 갖는다. 이 값은 3.1절에서 설명한 객체 크기 예측을 위한 통계치 추정 방법에 의해 정의되는 값으로, 학습 데이터셋에 존재하는 객체들의 너비와 높이의 분포에 의하여 결정되는 값이다. 이를 통해 네트워크에서 예측되는 객체들의 너비 값 b_w와 높이 b_h가 학습 데이터셋에 존재하는 객체들의 너비와 높이 값에 대한 확률 분포를 따르도록 제약 조건을 부여할 수 있다. 기존에 제안된 YOLOv2, YOLOv3의 객체 크기 예측을 위한 통계치 적응적인 지수 회귀 모델을 수정함에 따라 본 논문에서는 제안하는 객체 크기 예측을 위한 통계치 적응적인 선형 회귀 모델을 학습시키기 위한 손실 함수를 새로이 정의하였으며, 제안하는 손실 함수는 아래와 같다.

$$\hat{g}_w = \frac{g_w - E(W)}{\sqrt{Var(W)}} \tag{9}$$

$$\hat{g}_h = \frac{g_h - E(H)}{\sqrt{Var(H)}} \tag{10}$$

$$Loss_w = (\hat{t}_w - \hat{g}_w)^2 \tag{11}$$

$$Loss_h = (\hat{t}_h - \hat{g}_h)^2 \tag{12}$$

여기서, g_w와 g_h는 예측되어야 하는 목표 값이다. $Loss_w$와 $Loss_h$는 객체의 너비와 높이 예측에 관한 손실 함수이다. $\hat{g}_w$와 $\hat{g}_h$는 g_w와 g_h를 평균값이 0이고 표준 편차가 1인 분포를 따르는 값으로 정규화한 값이다. 이는 학습 과정에서 $\hat{t}_w$와 $\hat{t}_h$의 평균값 및 표준 편차가 $\hat{g}_w$와 $\hat{g}_h$의 평균값과 표준 편차를 따르도록 제약조건을 부여하기 위함이다. 학습 과정의 목표는 $E(Loss_w)$와 $E(Loss_h)$를 최소화하는 것이다. 결과적으로, 이는 예측하는 바운딩 박스의 높이와 너비의 평균과 표준편차가 학습데이터의 평균과 표준편차에 근접하도록 한다. 본 논문에서 제안하는 객체의 너비와 높이 예측에 관한 손실 함수를 포함한 전체 손실 함수 $Loss_{det}$는 수식(13)과 같다.

$$Loss_{det} = Loss_{conf} + Loss_x + Loss_y + Loss_w + Loss_h + Loss_{class} \tag{13}$$

$$Loss_{conf} = -g_o \log(o) - (1-g_o)\log(1-o) \tag{14}$$

$$Loss_x = (t_x - g_x)^2 \tag{15}$$

$$Loss_y = (t_y - g_y)^2 \tag{16}$$

$$Loss_{class} = \sum_{c}^{N} (-g_c \log(p_c) - (1-g_c)\log(1-p_c)) \tag{17}$$

여기서, $Loss_{conf}$는 객체와 배경임을 분류하는데 사용되는 변수 o에 대한 손실함수로 해당 변수의 의미는 2절에서 설명한 o와 같으며, g_o는 목푯값을 의미한다. $Loss_x$, $Loss_y$는 객체의 위치를 예측하도록 사용되는 변수 t_x, t_y에 대한 손실함수로 각 변수의 의미는 바운딩 박스의 위치를 계산하기 위한 오프셋 값이고, g_x, g_y는 바운딩 박스의 목표 위치를 계산하기 위한 오프셋 값을 의미한다. $Loss_{class}$는 클래스를 예측하기 위한 변수인 p_c에 대한 손실함수로 해당 변수의 의미는 2절에서 설명한 p_c와 같으며 g_c는 목푯값, N은 예측해야하는 클래스의 개수를 의미한다. $Loss_x$, $Loss_y$, $Loss_w$, $Loss_h$, $Loss_{class}$는 앵커 박스가 할당된 피쳐맵의 특정 셀에서 객체를 예측하도록 선택된 경우에 계산되는 손실함수이다. 즉, 영상에 객체가 존재하지 않는 경우 해당 손실함수는 계산되지 않는다. $Loss_{conf}$,

$Loss_x$, $Loss_y$, $Loss_{class}$는 YOLOv3[13]와 동일하다.

Ⅳ. 실험 결과

본 논문에서 제안하는 방법의 성능을 평가하기 위해 Intel i7-6700 3.40GHz 프로세서, GTX 1080Ti 11GB GPU, 16.0GB 메모리를 사용하였고, 구현에는 Python 3.6[28]과 PyTorch 1.4.0[29]를 이용하였다. 실험에 사용된 데이터셋은 UFPR-ALPR[14]데이터셋이다. UFPR-ALPR 데이터셋은 총 4,500장의 이미지와 각 이미지별로 차량 번호판 검출을 위한 바운딩 박스 정보를 제공한다. 이중 1,800장은 학습용 데이터셋이며, 900장은 검증용 데이터셋이고, 1,800장은 테스트용 데이터셋이다. 학습 데이터셋 및 테스트 데이터셋의 1,800장의 이미지 중 1,440장의 이미지는 차량 번호판 검출용 이미지이며, 360장의 이미지는 이륜 자동차 번호판 검출용 이미지로 구성 되어있다. 본 논문에서는 학습 및 검출 성능을 측정함에 있어 차량 번호판에 대해서만 검출 성능을 측정하였다. 그림 2는 성능 평가를 위하여 사용한 UFPR-ALPR의 이미지 예시이다. 본 장의 4.1절에서는 제안하는 방법의 성능 검증을 위해 설계된 네트워크 및 학습 방법에 대해 설명한다. 4.2절에서는 검출 성능의 평가 지표로 AP를 이용하여 제안하는 방법과 기존 방법의 검출 성능을 비교한다.

1. 네트워크 및 학습 방법

본 절에서는 성능 비교를 위해 사용된 네트워크와 학습 방법에 대해 자세히 설명한다. 성능 비교를 위해 사용된 네트워크의 백본 네트워크는 경량화된 YOLOv3인 YOLOv3 tiny[13]와 동일하며, 검출이 수행되는 계층의 위치를 일부 수정하였으며, 네트워크에 할당되는 앵커 박스의 수를 1로 정의하였다. 성능을 비교하기 위해 객체의 너비 및 높이를 추정함에 있어 통계치 적응적인 지수 회귀 모델을 마지막 계층으로 사용하여 객체의 크기를 예측하도록 네트워크를 학습시켰을 때와, 제안하는 통계치 적응적인 선형 회귀 모

그림 2. UFPR-ALPR[14] 데이터셋의 샘플 이미지
Fig. 2. Sample images from UFPR-ALPR[14] dataset

델을 마지막 계층으로 사용하여 객체의 크기를 예측하도록 네트워크를 학습시켰을 때의 검출 성능을 비교한다. 그림 3은 실험에 사용된 YOLOv3tiny 네트워크를 도식화한 그림이다. 그림 3에서 BN 계층은 배치 정규화[30] 계층을 의미하고, Upsample 계층은 입력받은 피쳐맵을 최근접 이웃 업샘플링 방법을 이용하여 2배 업샘플링 하는 계층이다.

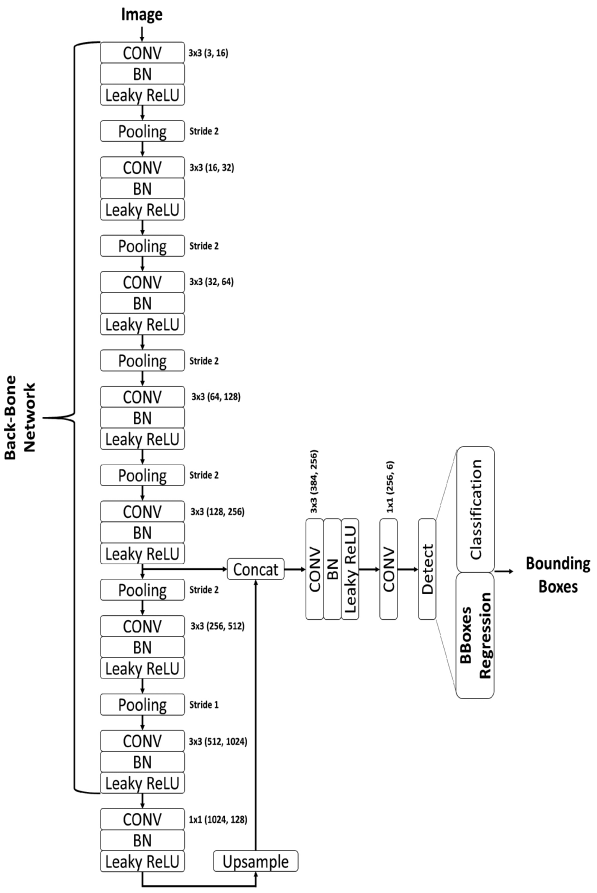


그림 3. 제안하는 방법의 성능을 평가하기 위하여 새롭게 디자인한 YOLOv3-tiny의 네트워크 구조

Fig. 3. The network architecture of newly designed YOLOv3 tiny for experiments

네트워크의 학습을 위한 미니 배치의 크기는 32이다. 네트워크를 학습시킴에 있어 이미지넷[24] 데이터셋을 이용하여 사전 학습된 백본 네트워크를 사용하여 학습시켰을 때의 결과와 사전 학습을 하지 않은 백본 네트워크를 사용하여 학습시켰을 때의 결과를 비교한다. 러닝 레이트의 경우 러닝 레이트 웜업 기법[31]을 사용하여, 6 에폭 동안 러닝 레이트 값을 0.001까지 선형적으로 증가시킨 후, 이후 러닝 레이트 값을 조정함에 있어 코사인 러닝 레이트 하강 기법[32]을 사용하였다. 또한, 네트워크의 학습을 안정시키고 성능을 개선시키기 위해 사용되는 배치 정규화[30] 계층을 네트워크에서 생략하였을 때와 포함하였을 때의 결과를 비교한다. 최적화에는 확률적 경사도 하강 (Stochastic Gradient Descent)기법이 사용되었으며, 총 50 에폭 동안 학습을 진행하였다. 학습 데이터셋의 크기를 증가시키기 위해 데이터 증강 기법을 적용하였으며, 사용된 데이터 증강 기법으로는 이미지 수평 방향 반전, 무작위 영역 샘플링[11], 영상 확대 및 축소, 영상 이동이 사용되었다. 모델의 파라미터를 학습하기 이전에, 학습 데이터셋으로부터 객체의 너비 및 높이의 통계치인 평균값과 표준 편차를 추정함에 있어, 학습 데이터셋에 데이터 증강 기법을 적용하지 않고 추정한 통계치를 사용하였을때와, 데이터 증강 기법을 적용하고 추정한 통계치를 사용하였을 때의 성능을 비교한다. 검출 성능을 비교함에 있어 Pascal VOC 데이터셋에 대한 검출 성능을 측정하기 위해 사용되는 방법[5]을 사용하여 측정된 AP를 비교한다. 이때, 예측된 바운딩 박스에 대하여 True Positive, False Positive를 분류함에 있어 Intersection of union(IoU)값의 경계값을 0.5로 설정하였다.

2. 제안하는 방법의 실험적 분석

본 절에서는 제안하는 통계치 적응적인 선형 회귀 모델을 네트워크의 마지막 계층으로 사용하여 객체의 크기를 예측하도록 학습시킨 네트워크와 YOLOv2 및 YOLOv3에서 제안된 통계치 적응적인 지수 회귀 모델을 네트워크의 마지막 계층으로 사용하여 객체의 크기를 예측하도록 학습시킨 네트워크의 검출 성능을 비교한다. 표 1은 네트워크를 학습시킴에 있어 배치 정규화 계층 (BN.), 사전 학습 기법 (Pre.)을 적용, 적용하지 않았을 때와 객체의 크기를 예측함에 있어 데이터셋 증강 기법이 적용된 데이터셋으로부터 추정한 통계치 (Aug. Prior)를 사용하여 네트워크를 학습시켰을 때의 검출 성능과 데이터셋 증강 기법이 적용되지 않은 데이터셋으로부터 추정한 통계치를 사용하여 네트워크를 학습시켰을 때의 검출 성능을 기록한 표이다. 표에 기록

된 검출 성능은 학습 과정에서 에폭별로 테스트 데이터셋에 대한 검출 성능을 측정하였을 때, 가장 높은 검출 성능을 기록한 것이다. 실험 결과, 배치 정규화 계층을 비포함, 사전 학습기법을 비적용하고 데이터셋 증강 기법이 적용되지 않은 데이터셋으로부터 추정한 통계치를 사용하여 네트워크를 학습시켰을 때, 기존 방법은 75.89에 달하는 AP를 기록하였고 제안하는 방법의 AP는 86.80에 달하는 AP를 기록하였다. 제안하는 방법이 기존 방법 대비 약 10.91 증가된 AP를 보임을 확인할 수 있다. 동일 조건하에 배치 정규화 계층을 포함하여 학습시킨 경우, 기존 방법은 89.21에 달하는 AP를 기록하였고 제안하는 방법은 90.61에 달하는 AP를 기록하였다. 제안하는 방법이 기존 방법 대비 약 1.4 증가된 AP를 보임을 확인할 수 있다. 두 실험의 결과를 비교하였을 때, 기존 방법의 경우 배치 정규화 계층을 생략하였을 때 AP가 약 15% 저하된 반면 제안하는 방법의 경우 AP가 약 5% 저하됨을 확인할 수 있다. 배치 정규화 계층은 학습 과정에서 배치 정규화 계층의 입력 피쳐가 손실 함수를 최소화할 수 있는 평균값과 표준 편차를 따르도록 정규화 하는 역할을 수행한다[30]. 배치 정규화 계층을 생략하였을 때 성능 저하가 적다는 결과는 네트워크의 마지막 계층에서 예측되는 값의 통계치가 목푯값의 통계치를 따르도록 설계하는 것이 네트워크의 최적화를 돕는 것으로 해석할 수 있다. 배치 정규화 계층을 포함, 사전 학습 기법을 적용하고 데이터셋 증강 기법이 적용되지 않은 데이터셋으로부터 추정한 통계치를 사용하여 네트워크를 학습시켰을 때, 기존 방법의 경우 80.27에 달하는 AP를 기록하였고 제안하는 방법의 경우 83.57에 달하는 AP를 기록하였다. 제안하는 방법이 기존 방법 대비 약 3.3 증가된 AP를 보임을 확인하였다. 과적합 문제를 완화하기 위하여 사용되는 사전 학습 기법을 적용하고 학습시켰을 때도 제안하는 방법이 보다 높은 검출 성능을 기록한 것을 확인할 수 있다. 사전 학습 기법을 적용한 경우와 적용하지 않은 경우를 비교하였을 때, 사전 학습 기법을 적용한 경우 검출 성능이 떨어진 것을 확인할 수 있다. 이때, 기존 방법의 경우 AP가 약 10% 저하됐고 제안하는 방법의 경우 AP가 약 7% 저하되었다. 제안하는 방법의 경우 학습 이전에 정의되는 모델의 초기 가중치 값에 보다 강건하게 학습이 될 수 있는 것으로 해석할 수 있다. 배치 정규화 계층을 포함, 사전 학습기법을 비적용하고 데이터셋 증강 기법이 적용된 데이터셋으로부터 추정한 통계치를 사용하여 네트워크를 학습시켰을 때, 기존 방법은 87.26에 달하는 AP를 기록하였고 제안하는 방법의 경우 87.49에 달하는 AP를 기록하였다. 실험 결과로 통계치를 추정함에 있어 데이터셋 증강 기법이 적용되지 않은 데이터셋으로부터 추정한 통계치를 사용하여 네트워크를 학습시켰을 때, 제안하는 방법과 기존 방법 모두 보다 높은 검출 성능을 기록한 것을 확인할 수 있다. 추가적으로, 제안하는 방법은 기존에 제안된 네트워크의 마지막 계층을 대체하는 방법이므로, 학습 및 추론에 대한 소요시간은 크게 변하지 않기 때문에 수행 시간에 대한 비교 실험은 진행하지 않았다.

표 1. 기존 방법과 제안하는 방법의 검출 성능 비교
Table 1. The comparison on UFPR-ALPR test dataset

Method	BN.	Pre.	Aug. Prior	AP
YOLOv3Tiny	X	X	X	75.89
	✓	X	X	89.21
	✓	✓	X	80.27
	✓	X	✓	87.26
Proposed	X	X	X	86.80
	✓	X	X	90.61
	✓	✓	X	83.57
	✓	X	✓	87.49

그림 4는 기존 방법과 제안하는 방법의 객체 검출 결과를 시각화한 것이다. 기존 방법과 제안하는 방법의 검출 결과를 시각화하기 위해 사용된 각각의 모델은 학습 과정에서 배치 정규화 기법과 사전 학습 기법이 적용되지 않았으며, 객체 크기 예측을 위한 통계치는 데이터 증강 기법이 적용되지 않은 데이터셋으로부터 추정한 통계치를 사용하여 학습되었다. 각 모델로 부터 추정된 바운딩 박스 중 바운딩 박스의 클래스 확률값이 0.5 이상인 바운딩 박스만을 시각화하였다. 기존 방법을 통해 추론된 바운딩 박스의 외곽선을 붉은색, 제안하는 방법을 통해 추론된 바운딩 박스의 외곽선을 노란색으로 설정하여 출력하였다.

그림 4. 기존 방법(좌) 및 제안하는 방법(우)의 정성적인 결과
Fig. 4. Qualitative results of YOLOv3 tiny and proposed method

V. 결 론

기존에 제안된 객체 검출 알고리즘의 경우 객체의 크기를 예측함에 있어 통계치 적응적인 지수 회귀 모델을 사용하였다. 하지만, 지수 회귀 모델은 지수 함수 특성상 예측 값에 따라 매우 큰 미분값을 네트워크의 파라미터로 전파할 수 있으며, 이는 역전파 과정에서 네트워크의 학습을 불안정하게 만들 수 있다. 본 논문에서는 네트워크에서 예측되는 객체의 크기를 예측함에 있어 통계치 적응적인 선형 회귀 모델을 제안하였다. 제안하는 통계치 적응적인 선형 회귀 모델의 예측 값은 학습 데이터셋에 존재하는 객체들의 너비와 높이의 평균값과 표준 편차를 따른다. 이를 통해, 네트워크의 학습 과정을 안정시킬 수 있으며, 최종적으로 객체 검출 성능을 향상시킬 수 있다. 본 논문에서 제안하는 통계치 적응적인 선형 회귀 모델의 성능 평가를 위하여 UFPR-ALPR 데이터셋을 사용하였다. 객관적 성능을 평가

하기 위한 지표로 AP를 비교하였을 때, 제안하는 방법이 기존 방법 대비 향상된 검출 성능을 보임을 확인하였다. 본 연구의 한계점으로, 학습 데이터셋에 존재하는 객체의 크기의 통계치와 테스트 데이터셋에 존재하는 객체의 크기의 통계치가 크게 다를 경우 기존 방법과 비교하여 성능이 저하될 수 있다. 향후 계획으로는 이러한 문제점을 완화할 수 있는 방법에 관한 연구를 진행할 예정이다.

참 고 문 헌 (References)

[1] Masi, Y. Wu, T. Hassner, and P. Natarajan, “Deep Face Recognition: A Survey,” in SIBGRAPI Conference on Graphics, Patterns and Images, Parana, 2018.

[2] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, “A Survey on 3D Object Detection Methods for Autonomous Driving Applications,” IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 10, pp. 3782-3795, 2019.

[3] OpenALPR, [Online]. Available: https://www.openalpr.com/

[4] W. Xiongwei, D. Sahoo, and S. C. H. Hoi, “Recent advances in deep learning for object detection,” Neurocomputing, vol. 396, pp. 39-64, 2020.

[5] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, “The PASCAL Visual Object Classes (VOC) Challenge,” International Journal of Computer Vision, vol. 88, no. 2, pp. 303-338, 2010.

[6] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: Common Objects in Context,” in European Computer Vision Conference, Zurich, 2014.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in IEEE Conference on Computer Vision and Pattern Recognition, Columbus, 2014.

[8] R. Girshick, “Fast R-CNN,” in IEEE International Conference on Computer Vision, Santiago, 2015.

[9] 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. 39, no. 6, pp. 1137-1149, 2017.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 2016.

[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single Shot MultiBox Detector,” in European Conference on Computer Vision, Amsterdam, 2016.

[12] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” in IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, 2017.

[13] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” arXiv, 2018.

[14] R. Laroca, E. Severo, L. A. Zanlorensi, L. S. Oliveira, G. R. Gonçalves, W. R. Schwartz, and D. Menotti, “A Robust Real-Time Automatic License Plate Recognition Based on the YOLO Detector,” in International Joint Conference on Neural Networks, Rio de Janeiro, 2018.

[15] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully Convolutional One-Stage Object Detection,” in IEEE/CVF International Conference on Computer Vision, Seoul, 2019.

[16] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, “YOLACT: Real-Time Instance Segmentation,” in IEEE/CVF International Conference on Computer Vision, Seoul, 2019.

[17] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and B. Serge, “Feature Pyramid Networks for Object Detection,” in IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, 2017.

[18] yolov3 tiny, [Online]. Available: https://github.com/pjreddie/darknet/blob/master/cfg/yolov3-tiny.cfg.

[19] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[20] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, 2001.

[21] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, 2005.

[22] H. Bay, T. Tuytelaars, and L. V. Gool, “SURF: Speeded Up Robust Features,” in European Conference on Computer Vision, Graz, 2006.

[23] C. Cortes and V. Vapnik, “Support-vector networks,” Machine Learning, vol. 20, no. 3, pp. 273-297, 1995.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems, Lake Tahoe, 2012.

[25] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” arXiv, 2014.

[26] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, 2016.

[27] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, 2017.

[28] Python, [Online]. Available: https://www.python.org/

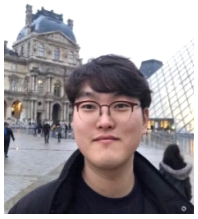
[29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” in Advances in Neural Information Processing Systems, Vancouver, 2019.

[30] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” in International Conference on Machine Learning, Lille, 2015.

저 자 소 개

권 용 혜

- 2018년 2월 : 광운대학교 로봇학부 학사
- 2020년 8월 : 광운대학교 컴퓨터공학과 석사
- 2020년 11월 ~ 현재 : 마크애니 전문연구요원 재직
- ORCID : http://orcid.org/0000-0001-8308-9909
- 주관심분야 : 컴퓨터비전



이 종 석

- 2016년 2월 : 광운대학교 전자공학과 학사
- 2018년 2월 : 광운대학교 전자공학과 석사
- 2018년 3월 ~ 현재 : 광운대학교 컴퓨터공학과 박사과정
- 2020년 2월 ~ 현재 : 디지털인사이트 선임연구원 재직
- ORCID : https://orcid.org/0000-0001-8045-0244
- 주관심분야 : 영상압축, 스파이킹 심층 신경망, 컴퓨터비전, 고해상도 위성영상 처리

심 동 규

- 1993년 2월 : 서강대학교 전자공학과 공학사
- 1995년 2월 : 서강대학교 전자공학과 공학석사
- 1999년 2월 : 서강대학교 전자공학과 공학박사
- 1999년 3월 ~ 2000년 8월 : 현대전자 선임연구원
- 2000년 9월 ~ 2002년 3월 : 바로비젼 선임연구원
- 2002년 4월 ~ 2005년 2월 : University of Washington Senior research engineer
- 2005년 3월 ~ 현재 : 광운대학교 컴퓨터공학과 교수
- ORCID : https://orcid.org/0000-0002-2794-9932
- 주관심분야 : 영상신호처리, 영상압축, 컴퓨터비전