

논문 2022-17-10

# 엣지 컴퓨팅 환경에서 적용 가능한 딥러닝 기반 라벨 검사 시스템 구현

## (Implementation of Deep Learning-based Label Inspection System Applicable to Edge Computing Environments)

배 주 원, 한 병 길\*

(Ju-Won Bae, Byung-Gil Han)

Abstract : In this paper, the two-stage object detection approach is proposed to implement a deep learning-based label inspection system on edge computing environments. Since the label printed on the products during the production process contains important information related to the product, it is significantly to check the label information is correct. The proposed system uses the lightweight deep learning model that able to employ in the low-performance edge computing devices, and the two-stage object detection approach is applied to compensate for the low accuracy relatively. The proposed Two-Stage object detection approach consists of two object detection networks, Label Area Detection Network and Character Detection Network. Label Area Detection Network finds the label area in the product image, and Character Detection Network detects the words in the label area. Using this approach, we can detect characters precise even with a lightweight deep learning models. The SF-YOLO model applied in the proposed system is the YOLO-based lightweight object detection network designed for edge computing devices. This model showed up to 2 times faster processing time and a considerable improvement in accuracy, compared to other YOLO-based lightweight models such as YOLOv3-tiny and YOLOv4-tiny. Also since the amount of computation is low, it can be easily applied in edge computing environments.

Keywords : Edge-Computing System, Deep Learning, Machine Vision, Vision Inspection, Object Detection

### 1. 서 론

제품의 생산과정에서 제품의 표면에 부착되거나 인쇄되는 라벨은 제품의 제조 일자, 분류 번호 등과 같은 정보들을 포함하고 있다. 공정 과정 중 라벨이 잘못 출력된 상태로 제품이 출고되면 불량회수와 같은 큰 물적 손실이 발생하기 때문에 라벨 검사는 생산 공정에서 필수적인 과정이다. 최근 스마트팩토리 분야에서는 머신 비전 기반의 비전 검사를 통해 컴퓨터가 자동으로 제품을 검사하는 솔루션을 적용하여 인적, 물적, 시간적 비용을 크게 줄이고 있다 [1-5]. 특히, 딥러닝 기반의 방법들을 사용하여 제품 검사의 정확도를 높이는 연구가 진행되고 있다 [6-8].

딥러닝 기반의 비전 검사는 주로 객체 검출 모델들을 사용하며, 단지 제품 이미지를 학습하는 것만으로도 편리하게 비전 검사를 수행할 수 있다. Faster R-CNN [9]은 입력 영상에서 객체가 있을 확률이 높은 영역을 찾아낸 후 각 영역에 대해 개별적으로 객체 인식을 수행하는 두 단계의 과정을 통해 객체 검출을 수행한다. YOLO [10]는 입력 영상에

서 객체의 위치와 인식을 동시에 수행하는 단일 과정으로 빠르게 객체 검출을 수행할 수 있다. 이런 객체 검출 모델들을 공정에 적용하기 위해서는 말단 기기에서부터 영상 데이터를 처리하고 자동화할 수 있는 엣지 컴퓨팅 시스템을 사용해 촬영 장치와 공장 설비들을 연결해야 한다. 하지만 일반적인 딥러닝 모델을 사용하기 위해서는 GPU와 같은 고성능의 전용 연산 장치들이 요구되며, 특히 엣지 컴퓨팅에는 GPU와 같은 전용 연산장치를 사용하지 않는 주로 낮은 사양의 임베디드 보드들을 사용하므로 많은 연산을 요구하는 딥러닝 모델을 사용하기에는 어려움이 있다. 낮은 성능의 엣지 컴퓨팅 장치에서 딥러닝 모델을 활용하기 위해 YOLOv3-tiny [11]와 같은 경량 딥러닝 모델이 연구되고 있지만 상대적으로 낮은 객체 검출 정확도를 보여준다. 따라서 엣지 컴퓨팅 환경에서 딥러닝 모델을 사용한 비전 검사를 적용하기 위해 경량화 모델을 사용하면서 동시에 정확도를 높일 수 있는 방법이 필요하다.

본 논문에서는 엣지 컴퓨팅 시스템에서 경량 딥러닝 모델을 사용하면서 높은 검출 정확도를 얻을 수 있는 Two-Stage 객체 검출 기반 라벨 검사 시스템을 제안한다. 제안하는 Two-Stage 방법은 경량화 모델의 검출 정확도를 높이기 위해 영상에서 제품의 라벨 부분을 검출하는 Label Area Detection Network와 검출된 라벨 부분만 사용해 라

\*Corresponding Author (kilyhan@etri.re.kr)

Received: Feb. 11, 2022, Revised: Mar 06, 2022, Accepted: Mar 24, 2022.  
J.W. Bae: Korea Maritime and Ocean University (Ph.D. Student)

B.G. Han: Electronics and Telecommunications Research Institute

\* 본 논문은 한국전자통신연구원 연구운영지원사업의 일환으로 수행되었음 (22ZD1100, 대경권 지역산업 기반 ICT 융합기술 고도화 지원사업).

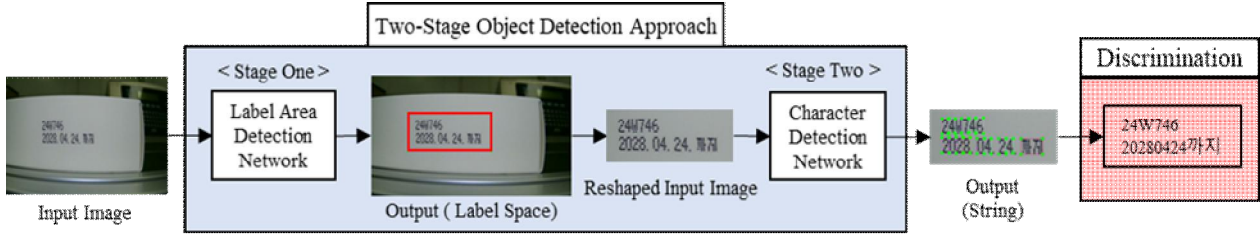


그림 1. 제안하는 Two-Stage 객체 검출 방법의 순서

Fig. 1. Sequence of proposed Two-Stage object detection approach

벨 내 문자를 검출하는 Character Detection Network를 사용하는 두 단계로 수행된다. 또한, 엣지 컴퓨팅 시스템에서 사용하기 위한 경량화 모델인 SF-YOLO [12]를 사용함으로써 적은 연산량으로 빠르고 정확한 문자 검출이 가능하다. 제안하는 시스템의 성능을 검증하기 위해 안드로이드 기반 임베디드 보드에서 검출 정확도, 연산량, 처리시간을 비교한다. 마지막으로, 본 논문의 구성은 다음과 같다. 2장에서 비전 검사와 딥러닝 기반 객체 검출에 대해 설명하고, 3장에서는 제안하는 Two-Stage 객체 검출 기반 라벨 검사 시스템의 구성 및 Two-Stage 객체 검출 방법, SF-YOLO 모델 그리고 학습 과정이 서술되며, 4장에서는 실험을 통해 제안하는 시스템의 성능을 검증한 후 5장에서 결론을 맺는다.

## II. 관련 이론

### 1. 머신 비전 기반 비전 검사

머신 비전 기반 비전 검사 시스템은 기계에 시각과 판단 기능을 부여해 제품을 검사할 수 있는 시스템이며, 모든 공정 과정을 자동화하는 스마트 팩토리의 제품 검사 솔루션으로써 사용되고 있다. 비전 검사는 규칙 기반의 방식 [4, 13, 14]과 머신 러닝 기반의 방식 [15-17]이 있다. 규칙 기반의 방식은 특정한 규칙에 따라 이미지에서 객체의 특징을 추출하고 추출된 특징을 재가공하여 제품의 결함을 검사한다. 규칙 기반의 방식은 규칙을 정하는 사람의 기준이나 제한된 환경, 검사 대상마다 규칙을 따로 설정해야하기 때문에 구축비용이 많이 소요된다는 단점이 있다. 하지만 머신 러닝 기반의 방식은 제품의 학습 데이터만 구축되면 쉽게 적용할 수 있어 규칙 기반의 방식보다 구축비용이 적고, 딥러닝 기술을 주로 사용하기 때문에 검사 대상에 따른 데이터셋만 구축하면 바로 적용 가능하기 때문에 최근 활발하게 연구되고 있다 [5, 18]. [5]는 비원형 유리병 생산 공정의 품질 향상을 위해 여러 CNN과 Bottleneck구조를 연결한 모델을 사용해 불량률 검출하는 연구를 수행했다. [18]는 VGG-19를 사용해 제조 영상 데이터를 학습하고 불량 영상은 원인에 따라 분류하여 구분할 수 있는 기법에 대해 연구했다. 이러한 딥러닝 기반 비전 검사 연구들은 딥러닝 모델을 학습하고 운용하기 위해 고성능의 GPU를 사용한 PC 환경을 요구하는데 엣지 컴퓨팅 시스템은 저사양으로 운용되기 때문에 딥러닝 모델의 경량화 등을 통한 최적화가 필요하다.

### 2. YOLO

딥러닝 기반 객체 검출 모델인 YOLO는 객체 검출에 필요한 객체 인식과 객체 영역 검출을 단일 회귀로 빠르고 정확하게 수행한다는 장점이 있다. YOLO는 입력 이미지를 특정 크기의 그리드로 분할한 후, 각 그리드에서 검출 객체와 객체가 이미지에서 차지하고 있는 공간인 Bounding box를 예측하는 형태로 학습한다. YOLO는 처리 과정이 단순하기 때문에 다른 객체 검출 모델 [9, 19, 20]보다 연산량을 크게 줄여 사용성을 높였고, 꾸준히 성능향상을 위한 알고리즘 개선에 관한 연구가 이루어지고 있다 [21-23]. 하지만 여전히 저사양의 임베디드 보드에서 사용하기에는 어려움이 있으며 이를 해결하기 위한 경량화 연구도 최근 다양하게 이루어지고 있다. YOLOv3-tiny [11]는 기존의 YOLOv3 [21]에서 연산량을 줄이기 위해 YOLOv3의 구조를 간소화했으며 정확도는 다소 떨어지지만 검출 속도를 크게 높였다. YOLOv3-Lite [24]는 경량화된 YOLOv3에 특징 추출의 Backbone Network에서 고차원의 특징과 저차원의 특징들을 결합해 풍부하게 특징 정보를 보존하는 모델을 제안했다. SF-YOLO [12]는 Residual Network를 기반으로 이미지 특징 정보를 보강하고 특징 정보 보존과 학습을 최적화하는 Scalable Convolutional Blocks를 사용하여 타 경량화 모델들보다 적은 연산량과 높은 정확도를 보였다.

## III. 제안하는 시스템

### 1. 제안한 시스템의 구성

본 논문에서는 엣지 컴퓨팅 시스템에서도 높은 정확도의 문자 검출이 가능한 Two-Stage 객체 검출 방법을 사용하는 머신 비전 기반 라벨 검사 시스템을 제안한다. 제안하는 시스템은 그림 1과 같은 순서로 진행되며, 입력받은 제품 이미지에 대해 문자열 영역을 검출하고 검출된 영역만큼 이미지를 재가공한 후 재가공된 이미지에서 문자를 검출하는 Two-Stage 객체 검출 방법을 통해 라벨 내용에 대한 문자열을 만들어낸다. Two-Stage 객체 검출 방법은 문자열 영역을 검출하는 Label Area Detection Network가 포함된 Stage One과 검출된 문자열 영역에서 숫자와 영문자를 검출하는 Character Detection Network가 포함된 Stage Two의 순서로 이루어진다. 그 다음, 실제로 출력될 정답 문자열과 일치하는지 비교하는 검사과정을 통해 라벨이 정확하게

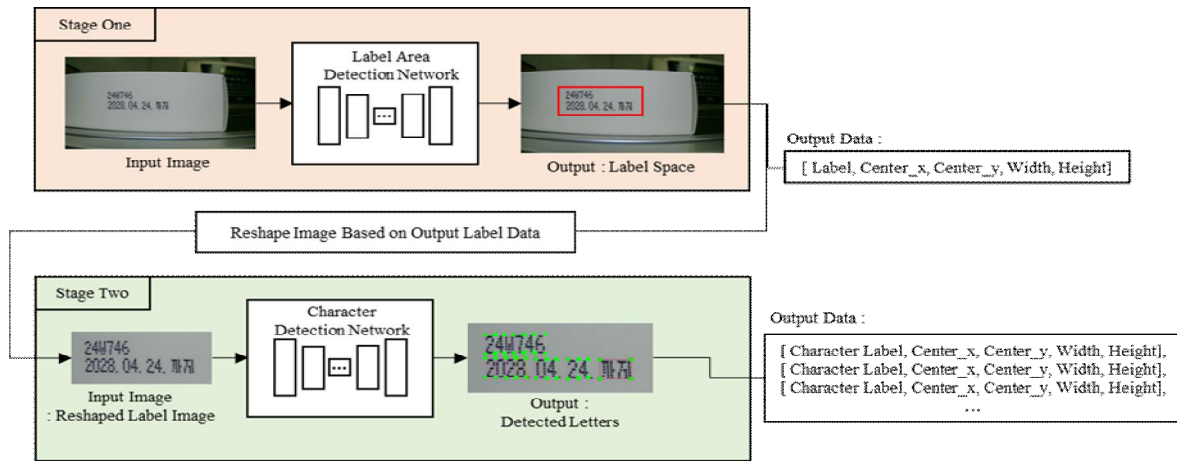


그림 2. 제안하는 Two-Stage 방법의 내부 데이터 입출력 과정  
 Fig. 2. The Internal Data Input · Output Procedure of the Proposed Two-Stage Approach

출력되었는지 아닌지를 검사한다. 이후 절에서는 제안하는 Two-Stage 객체 검출 방법과 제안하는 모델에서 사용하는 객체 검출 모델, 제안하는 시스템에서 사용하는 데이터셋에 대해 자세히 서술한다.

2. Two-Stage 객체 검출 방법

옛지 컴퓨팅 환경에서 사용하기 위한 경량 객체 검출 모델은 빠른 연산처리 속도를 가지고 있지만 그만큼 사용하는 CNN 레이어의 수도 작기 때문에 검출 정확도가 낮다. 또한, YOLO는 전체 이미지를 특정 크기의 그리드로 분할하여 그리드 내 객체들을 검출하는데 제품 이미지의 라벨은 보통 전체 이미지에 비해서 크기가 작다. 따라서 문자들의 Bounding box가 그리드보다 작게 형성되어 이미지에서 문자를 학습할 때 학습이 제대로 이루어지지 않는다. 본 논문에서는 이러한 문제를 해결하기 위해 먼저 객체의 라벨 영역을 학습한 모델을 사용해 이미지에서 라벨 영역을 검출하고, 라벨의 문자를 학습한 모델에 라벨 영역만 입력함으로써 영역 내부의 문자를 검출하는 Two-Stage 객체 검출 방법을 제안한다. 제안하는 Two-Stage 객체 검출 방법은 그림 2와 같은 순서로 수행된다.

첫 번째 과정인 Stage One은 Label Area Detection Network를 사용해 카메라로 수집된 이미지에서 제품의 라벨 영역을 검출한다. Label Area Detection Network는 제품의 이미지와 제품 내 라벨 영역을 표시한 라벨 위치 정보를 검출한다. 객체 검출에서 라벨 영역의 위치만을 찾는 역할을 한다. Stage One을 통해 출력되는 데이터를 시각화하면 그림 2의 첫 번째 Output인 붉은 색 영역의 Label Space로 표현되며, 데이터는 라벨의 중심에 대한 (x, y)의 좌표와 라벨 영역의 Width, Height로 구성된다.

그리고 첫 번째 과정에서 검출된 라벨 영역의 위치 정보를 기반으로 이미지를 재가공하여 그림 2의 Reshaped Label Image로 변환한다. 이렇게 변환한 라벨 영역 이미지에서 문자가 차지하는 영역의 비율은 원본 이미지의 문자가

차지하는 영역의 비율보다 높기 때문에 YOLO 기반의 객체 검출 모델이 더 정확하게 이미지를 학습할 수 있다.

두 번째 과정인 Stage Two는 Character Detection Network를 사용해 라벨 영역 이미지에서 문자들을 검출한다. Character Detection Network는 이미지 내 문자들의 종류와 검출된 문자가 이미지에서 차지하는 영역인 Bounding Box의 정보들을 출력한다. 기존 제품의 이미지에서 문자를 검출하는 것보다 라벨 영역만 포함된 이미지에서 문자를 검출함으로써 문자 이미지들의 특징 정보에만 집중해 정확한 결과를 얻을 수 있다. 이 과정을 통해 출력되는 데이터를 시각화하면 그림 2의 두 번째 출력인 초록색 점으로 연결된 영역들이며, 데이터는 문자의 종류, 문자 영역 중심에 대한 (x, y)의 좌표와 라벨 영역의 Width, Height로 구성된다.

이러한 Two-Stage 기반의 객체 검출 방법을 사용하면 제품에서 작은 영역을 차지하고 있는 라벨 정보에 대한 결과를 정확하게 얻을 수 있다. 그리고 이러한 두 과정을 거치기 위해서 빠르고 정확한 경량된 객체 검출 모델을 사용하면 더 높은 효과를 얻을 수 있으며 제안하는 시스템에서는 옛지 컴퓨팅에 효율적인 경량 객체 검출 모델인 SF-YOLO를 사용한다.

3. SF-YOLO

본 논문에서 제안하는 Two-Stage 객체 검출 시스템은 저사양의 옛지 컴퓨팅 시스템에서도 정확하고 빠른 객체 검출을 수행하기 위해 옛지 컴퓨팅을 위한 경량 객체 검출 모델인 SF-YOLO를 Label Area Detection Network와 Character Detection Network로 사용한다. SF-YOLO는 모델의 경량화를 위해 다층의 CNN에 Residual Network를 적용한 Residual Convolutional Block과 이미지 특징 정보를 보강하는 Dense Convolutional Blocks, 이미지의 특징 부분을 보강하기 위해 이전의 특징 추출 레이어와 차원 보강 레이어를 연결하는 Recursive Convolutional Blocks들을 사용해 낮은 연산량에도 높은 객체 검출 정확도를 보였다.

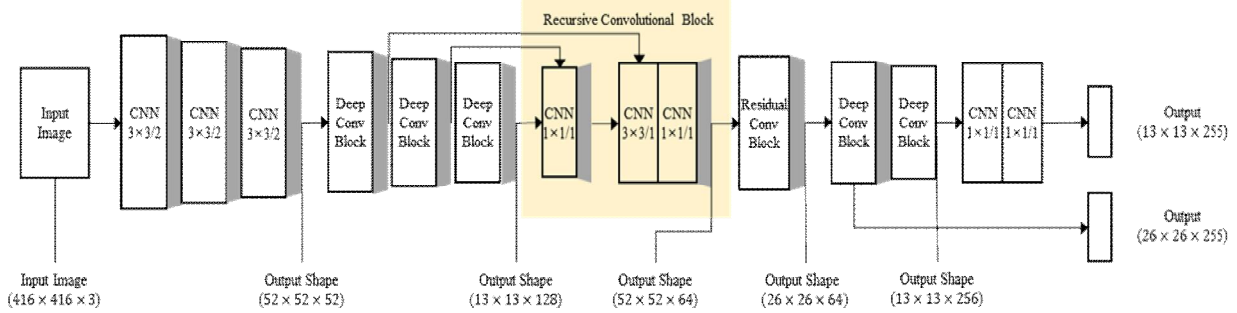


그림 3. SF-YOLO 모델의 구조  
Fig. 3. Structure of SF-YOLO Model

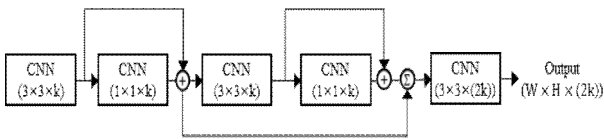


그림 4. Dense Convolutional Blocks의 구조  
Fig. 4. Structure of Dense Convolutional Blocks

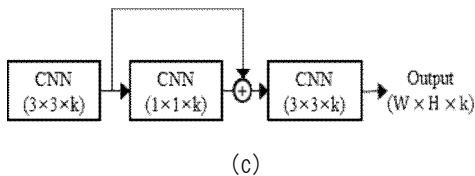


그림 5. Residual Convolutional Blocks의 구조  
Fig. 5. Structure of Residual Convolutional Blocks



그림 6. 라벨 데이터셋의 구조  
Fig. 6. Structure of label dataset

SF-YOLO의 구조는 그림 3과 같다. 그림 3의 SF-YOLO에서 입력된 이미지는 Convolutional Blocks를 거치면서 이미지의 특징 정보들을 추출한다. 그리고 Dense Convolutional Blocks를 거치면서 특징 정보의 채널수를 늘리고, Recursive Convolutional Blocks와 Residual Convolutional Blocks에서 이전 레이어의 특징 정보들을 추가하면서 특징 정보들을 보강한다. SF-YOLO는 Residual 구조를 사용함으로써 이전 특징 정보들을 계속 보강하여 이전의 경량화 모델들보다 정확한 검출 성능을 얻을 수 있었고 채널수를 최적화함으로써 연산량도 낮출 수 있었다.

그림 4은 Dense Convolutional Blocks의 구조이다. Dense Convolutional Blocks는 2개의 CNN만을 사용하는 방법에

비해 계산 복잡도를 낮추고 Concatenate를 통해 정보를 보강한다. 그림 5는 Residual Convolutional Blocks의 구조이다. Residual Convolutional Blocks의  $3 \times 3$  CNN은 공간적인 특징 정보들을 분석하고  $1 \times 1$  CNN을 통해 채널들 간의 관계를 분석하며 Residual 구조를 통해 정보들을 연결하여 계산 복잡도를 최소화한다. 이러한 Block들을 사용함으로써 더 최적화된 모델 크기와 높은 객체 검출 성능, 빠른 모델 연산처리 속도를 보이게 된다.

#### 4. 데이터셋 구축

제안하는 라벨 결합 검출 시스템의 객체 검출 모델에 학습하기 위해 본 논문에서는 그림 6과 같이 데이터셋을 구축하였다. 라벨은 그림 6에 포함된 세 종류의 형태로 되었으며 숫자와 영문자는 단어 하나 당, ‘제조’와 ‘까지’는 함께 지정하여 검출한다. 따라서 문자 데이터는 숫자 0~9의 10자, 영문자는 숫자 0, 1과 유사한 ‘O’, ‘I’를 제외한 24자, ‘제조’, ‘까지’를 포함해 36개로 구성되어있다. 데이터셋에는 라벨 영역의 Bounding box와 라벨에 포함된 각 문자의 종류, Bounding box의 정보가 포함되어 있는데 Bounding box의 정보는 Bounding box 중심의 X, Y 좌표, 라벨의 가로, 세로의 길이 순서로 저장되어있다. 이 때 좌표와 길이는 이미지의 X, Y 크기로 정규화한 형태로 저장된다. 3.3에서 서술한 Two-Stage 객체 검출 방법의 Label Area Detection Network는 라벨 영역의 정보만 학습해 이미지에서 라벨 영역을 찾고, Character Detection Network는 라벨 영역만 잘라낸 이미지와 문자 정보들을 학습해 문자열을 검출한다. 이를 위해 라벨의 Bounding box 정보를 사용하여 이미지를 재가공해 라벨 영역만 포함된 이미지를 얻어서 Character Detection Network의 학습에 사용한다.

### IV. 실험 결과 및 분석

#### 1. 실험 설계

제안하는 라벨 검사 시스템의 정확도와 연산속도에 대한 타당성을 검증하기 위해 고사양 데스크톱 환경이 아닌 실제 생산공정에서 사용될 수 있는 저사양의 엣지 컴퓨팅 연산

표 1. Two-Stage와 One-Stage 검출 방법의 성능 비교  
Table 1. Comparison of performance of Two-Stage and One-Stage detection approach

Model	One-Stage Approach		Two-Stage Approach					
	Model FLOPs [ $\times 10^{10}$ ]	F1-Score	Stage-One FLOPs [ $\times 10^{10}$ ]	Stage-Two FLOPs [ $\times 10^{10}$ ]	F1-Score	Number Accuracy (%)	Word Accuracy (%)	Processing Time (ms)
SF-YOLO	<b>2.552</b>	0.31	<b>2.523</b>	<b>2.551</b>	<b>0.97</b>	<b>99.5</b>	<b>66.7</b>	<b>478</b>
YOLOv3-tiny	5.505	0.40	5.448	5.504	0.96	98.5	61.5	806
YOLOv4-tiny	6.845	0.24	6.787	6.843	<b>0.97</b>	<b>99.5</b>	65.9	1,004
YOLO v3	65.573	<b>0.89</b>	65.304	65.560	0.95	<b>98.8</b>	65.9	8,768

장치 (Amlogic S912 - Octa-core ARM Cortex-A53)를 사용하여 실험을 수행하였다. 일반적으로 테스트트의 환경 (NVIDIA 사의 Geforce RTX 2070 SUPER, 9.6062 TFLOPS)은 TFLOPS (FLoating point OPerations per Second) 단위이고 옛 컴퓨터에서 사용되는 임베디드 연산장치 (Amlogic S912 - Octa-core ARM Cortex-A53, 5.1 GFLOPS)는 GFLOPS 단위이기 때문에 연산처리의 성능은 1000배 이상의 차이로 상당히 모델 연산 성능의 제약이 있다.

제안하는 시스템에서 사용하는 데이터셋은 74개의 다른 걸면 색, 문자열들을 포함한 제품 박스를 컨베이어벨트에서 다른 시간대에 촬영한 232장의 제품 이미지로 만들어졌다. 컨베이어벨트의 다른 시간대에 촬영한 이미지를 사용한 이유는 제품이 컨베이어벨트 위에서 움직이는 상황을 고려하여 여러 위치에서 촬영함으로써 다양한 위치의 제품 검출이 가능하게 하기 위함이다. 구축된 데이터셋에서 139장은 학습, 30장은 검증, 63장은 테스트에 사용하였다. 학습은 대표적인 YOLO 프레임워크인 Darknet [25]에서 수행하였으며, 모델의 학습률인 Learning Rate는 0.001, 학습의 한 회 반복될 때 사용하는 이미지 개수인 Batch Size는 64, 전체 학습이 이루어지는 반복수인 Max Batch는 200,200 으로 설정해 충분한 학습이 이루어지게 하였다.

2. 제안하는 시스템의 성능 비교

이 절에서는 제안하는 시스템의 설계 타당성을 검증하기 위해 제안하는 방법인 Two-Stage 객체 검출 방법과 전체 제품 이미지를 사용해 한 번의 과정으로 문자를 검출하는 One-Stage 방식을 비교한다. 각 방법의 자체 성능을 검증하기 위해 구축한 테스트 데이터셋을 사용했으며 검출 정확도의 비교는 F1-Score를 사용한다. F1-Score는 Precision과 Recall의 조화평균으로 계산되는데, 검출 결과의 Precision과 Recall을 모두 고려한 검출 정확도를 알 수 있는 지표이다. 그리고 각 방식의 모델 처리 속도에 대한 비교는 사용하는 연산 장치의 성능이나, 모델을 사용하는 소프트웨어에 따라 달라질 수 있다. 따라서 시스템의 연산처리 속도를 비교하기 딥러닝 모델 자체의 연산량을 비교하는 FLOPs (FLoating point OPerationS)를 사용했다. 그리고 제안하는 시스템에서 사용하는 SF-YOLO의 성능을 입증하기 위해서

다른 객체 검출 모델들인 YOLO v3, YOLOv3-tiny, YOLOv4-tiny [25]를 학습하여 결과를 비교한다.

표 1의 결과에서 One-Stage는 한 모델만 사용해서 검출을 진행하고 Two-Stage 방법은 두 모델을 사용하기 때문에 전체 모델의 FLOPs는 Two-Stage 방법의 0.5배이며, 이를 통해 연산처리 속도는 2배 차이가 난다는 것을 알 수 있다. 하지만 One-Stage에서 경량화 모델들은 F1-Score가 0.34, 0.41, 0.24로 라벨 검사에서 사용하기 어려운 성능을 보인다. Two-Stage 방법은 비록 연산처리 속도는 One-Stage에 비해 느리지만 경량화 모델의 F1-Score가 0.97, 0.96, 0.97로 크게 향상되었음을 확인할 수 있다. One-Stage에서 가장 높은 F1-Score를 보유한 YOLO v3은 모델의 연산량이  $65.573 \times 10^{10}$ FLOPs인데, Two-Stage의 SF-YOLO의 총 연산량인  $5.074 \times 10^{10}$ FLOPs보다 12배 이상 차이가 나기 때문에 처리 속도가 매우 느린 것을 확인할 수 있다. 따라서 각 모델들의 F1-Score도 낮고 FLOPs가 커서 연산처리 시간이 많이 소요되는 One-Stage 방법은 실시간 검출에 적절하지 않은 방법이라는 것을 알 수 있다.

표 1의 Number Accuracy와 Word Accuracy는 비교 모델들의 문자, 숫자 검출 정확도와 직접 개발한 안드로이드 기반 소프트웨어에서의 처리 시간을 비교하였다. 실제 문자, 숫자 검출 정확도 비교를 위해 테스트 데이터셋에 포함된 숫자 1221자와 영문자 69자에 대해 출력 결과의 정확도를 비교하였다. 표 1에서 알 수 있듯이 연산량이 가장 작은 SF-YOLO를 사용했을 때 478ms로 가장 빨랐고 Two-Stage에서 가장 높은 F1-Score를 보임으로써 초당 2회의 라벨 검사를 진행함으로써 실시간 검사에 적절하다는 것을 알 수 있다.

문자 검출 정확도에 대해서, 표 1에서 알 수 있듯 Two-Stage에서의 경량화 모델들의 F1-Score는 0.96 이상으로 높았다. 숫자 검출의 정확도는 모든 모델이 98% 이상으로 높았는데 SF-YOLO와 YOLOv4-tiny가 99.5%로 가장 높았다. 하지만 문자 검출의 정확도는 YOLOv4-tiny와 YOLOv3는 65.9%였고 SF-YOLO는 66.7%로 가장 높았다. 문자 검출은 SF-YOLO가 가장 많이 정확하게 검출하였지만 모든 경량화 모델의 문자 검출 정확도는 숫자 검출 정확도에 비해 낮았다. 현재 구축한 데이터셋에 사용되는 문자

의 종류는 ‘제조’, ‘까지’와 같은 문자는 모든 이미지에 포함되어 있지만 사용하는 영문자에 대해서는 제품의 고유번호에 들어가는 영문자 1개, 유통기한의 ‘EXP’ 부분에만 영문자를 사용하기 때문에 다양한 영문자를 학습하기에는 데이터셋의 양이 다소 부족한 것으로 보인다. 엣지 컴퓨팅 시스템을 사용하는 스마트 팩토리에서는 제품을 생산할 때마다 제품의 영상 데이터를 얻을 수 있기 때문에 제조 공정 중 얻을 수 있는 추가 제품 데이터를 데이터셋에 추가하고, 이전에 학습된 가중치에서 추가적으로 학습을 수행함으로써 이를 해결할 수 있을 것이다.

## V. 결론

본 논문에서는 엣지 컴퓨팅 장치에서 라벨 검사가 가능한 Two-Stage 객체 검출 기반의 라벨 검사 시스템을 제안하였다. 제안하는 시스템은 Label Area Detection Network에서 라벨 영역을 찾고 Character Detection Network에서 라벨 영역 문자를 검출하는 Two-Stage 객체 검출 방법을 사용함으로써 경량 딥러닝 모델로도 정확한 라벨 검사를 가능하게 하였다. 또한 검사를 위한 객체 검출 모델로 연산량을 줄이면서 검출 정확도까지 향상한 최적 경량화 모델인 SF-YOLO를 사용해 빠르게 라벨 검사를 가능하게 하였다.

제안하는 시스템은 저사양의 안드로이드 기반 임베디드 보드에서도 정확하게 라벨 검사를 수행할 수 있기 때문에 시스템 구축비용이 낮다는 장점이 있다. 따라서 엣지 컴퓨팅 시스템을 사용하는 스마트 팩토리에 저렴하게 적용할 수 있다. 또한, 제안하는 시스템은 라벨 검사뿐만 아니라 다른 비전 검사 시스템에도 데이터셋만 학습시키면 활용할 수 있다. 하지만 보다 정확한 검출을 위해서 제품에 대한 적정량의 데이터 확보가 필요한데, 엣지 컴퓨팅 시스템에서 지속적으로 데이터를 확보하고 보강할 수 있기 때문에 향후 지속적인 데이터 확보와 추가적인 현장 적용, 검증을 통해 해결할 수 있을 것이다. 제안하는 엣지 컴퓨팅 시스템에서의 딥러닝 기반 라벨 검사 시스템을 통해 많은 스마트 팩토리에서 쉽게 엣지 컴퓨팅 기반의 라벨 검사 시스템을 적용함으로써, 엣지 컴퓨팅 기반 스마트 팩토리 시스템의 보급과 발전에 기여할 수 있을 것으로 기대된다.

## References

- [1] B. D. Kang, O. H. Kwon, C. U. Seong, J. D. Jeon, J. S. Eom, J. H. Kim, J. W. Lee, S. K. Kim, “Effective Face Detection Using Principle Component Analysis and Support Vector Machine,” *Journal of Korea Multimedia Society*, Vol. 9, No. 11, pp. 1435-1444, 2006.
- [2] H. Y. Maeng, J. H. Kim, Y. H. Ko, “Image Alignment Method Based on CUDA SURF for Multi-Spectral Machine Vision Application,” *Journal of Korea Multimedia Society*, Vol. 17, No. 9, pp. 1041-1051, 2014 (in Korean).
- [3] S. B. Hong, S. W. Hong, K. H. Lee, “Recognition Direction Improvement of Target Object for Machine Vision based Automatic Inspection,” *Journal of Korea Institute of Information and Communication Engineering*, Vol. 23, No. 11, pp. 1384-1390, 2019 (in Korean).
- [4] H. Y. Kim, J. S. Cho, “Exterior Vision Inspection Method of Injection Molding Automotive Parts,” *Journal of Institute of Korean Electrical Electronics Engineers*, Vol. 23, No. 2, pp. 127-132, 2019 (in Korean).
- [5] S. M. Yoon, S. H. Lee, “Development of Defect Inspection System for Polygonal Containers,” *Journal of Institute of Korean Electrical Electronics Engineers*, Vol. 25, No. 3, pp. 485-492, 2021 (in Korean).
- [6] M. J. Kim, J. H. Shin, “Development of Defect Inspection System for Polygonal Containers,” *Korean Journal of Computational Design and Engineering*, Vol. 25, No. 3, pp. 246-255, 2020 (in Korean).
- [7] G. H. Kang, J. M. Shon, G. W. Sim, “Comparative Analysis of Anomaly Detection Models using AE and Suggestion of Criteria for Determining Outliers,” *Journal of the Korea Society of Computer and Information*, Vol. 26, No. 8, pp. 23-30, 2021 (in Korean).
- [8] H. J. Park, Y. W. Lee, B. G. Kim, “Efficient Tire Wear and Defect Detection Algorithm Based on Deep Learning,” *Journal of Korea Multimedia Society*, Vol. 24, No. 8, pp. 1026-1034, 2021 (in Korean).
- [9] S. Ren, K. He, R. Girshick, 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, 2016.
- [10] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779-788, 2016.
- [11] P. Adarsh, P. Rathi, M. Kumar, “YOLO v3-Tiny: Object Detection and Recognition using One Stage Improved Model,” *2020 6th International Conference on Advanced Computing and Communication Systems*, pp. 687-694, 2020.
- [12] B. G. Han, J. G. Lee, K. T. Lim, D. H. Choi, “Design of a Scalable and Fast YOLO for Edge-Computing Devices,” *Sensors*, Vol. 20, No. 23, 2020.
- [13] Y. S. Jung, K. H. Park, “O-ring Size Measurement Based on a Small Machine Vision Inspection Equipment,” *Journal of the Korea Industrial Information Systems Research*, Vol. 19, No. 4, pp. 41-52, 2014 (in Korean).
- [14] B. J. Park, K. S. Hahn, H. J. Lee, “Automated Visual Inspection System of Junction Box Using Color Inspection and Template Matching,” *Journal of Korea*

Multimedia Society, Vol. 13, No. 3, pp. 392-399, 2010 (in Korean).

[15] D. M. Ko, K. S. Choi, "Accurate PCB Outline Extraction and Corner Detection for High Precision Machine Vision," Journal of the Semiconductor & Display Technology, Vol. 16, No. 3, pp. 53-58, 2017 (in Korean).

[16] G. S. Kim, S. Lee, J. S. Cho, "A Learning-Based Visual Inspection System for Part Verification in a Panorama Sunroof Assembly Line Using the SVM Algorithm," Journal of Institute of Control, Robotics and Systems, Vol. 19, No. 12, pp. 1099-1104, 2013 (in Korean).

[17] D. H. Kim, S. B. Boo, H.C . Hong, W. G. Yeo, N. Y. Lee, "Machine Vision-Based Defect Detection Using Deep Learning Algorithm," Journal of the Korean Society For Nondestructive Testing, Vol. 40, No. 1, pp. 47-52, 2020 (in Korean).

[18] G. A. Ryu, K. H. Yoo, "Application of Manufacturing Process Data Classification Using Image Data based CNN," Journal of Information Technology and Architecture, Vol. 15, No. 3, pp. 337-343, 2018 (in Korean).

[19] R. Girshick, "Fast r-cnn," Proceedings of the IEEE International Conference on Computer Vision, pp.1440-1448, 2015.

[20] K. He, G. Gkioxari, P.Dollár, R. Girshick, "Mask R-Cnn," Proceedings of the IEEE International Conference on Computer Vision, pp. 2961-2969, 2017.

[21] J. Redmon, A. Farhadi, "Yolov3: An Incremental Improvement," Computer Vision and Pattern Recognition, pp. 1804-2767, 2018.

[22] A. Bochkovskiy, C. Y. Wang, H. Y. M. Liao, "Yolov4: Optimal Speed and Accuracy of Object Detection," arXiv preprint arXiv:2004.10934, 2020.

[23] C. Y. Wang, A. Bochkovskiy, H. Y. M. Liao, "Scaled-Yolov4: Scaling Cross Stage Partial Network," Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13029-13038, 2021.

[24] Y. Li, Z. Han, H. Xu, L. Liu, X. Li, K. Zhang, "YOLOv3-lite: A Lightweight Crack Detection Network for Aircraft Structure Based on Depthwise Separable Convolutions," Applied Sciences, Vol. 9, No. 18, 2019.

[25] <https://github.com/AlexeyAB/darknet/>

**Ju-Won Bae (배 주원)**



2019 Ocean Engineering from Hankuk Korea Maritime and Ocean University (B.S.)  
 2021 Electronics & Electrical Engineering from Korea Maritime and Ocean University (M.S.)

2022~Electronics & Electrical Engineering from Korea Maritime and Ocean University (Ph.D. Student)

Field of Interests: deep learning & computer vision  
 Email: wjb0406@g.kmou.ac.kr

**Byung-Gil Han (한병길)**



2005 Electronics and Electrical Engineering from Kyungpook National University(B.S.)  
 2007 Electronics Engineering from Kyungpook National University(M.S.)  
 2021 Electronics Engineering from Kyungpook National University(Ph.D.)

Career:  
 2010~Senior Researcher of Electronics and Telecommunications Research Institute(ETRI)  
 Field of Interests: computer vision, pattern recognition, and deep learning  
 Email: kilyhan@etri.re.kr