

웨이퍼 맵의 다중 결함 패턴 인지를 위한 벡터 표현 기반 캡슐 네트워크 구현

김미선*, 최지환*, 이하림^o

Multiple Defect Pattern Recognition in a Wafer Map Using Vector-Representation Based Capsule Network

Misun Kim*, Ji Hwan Choi*, Harim Lee^o

요약

반도체 공정 과정은 고품질, 고수율의 제품 생산을 보장해야만 한다. 하지만 반도체 공정은 여러 세부 공정들로 복잡하게 이루어져 있기 때문에, 경험이 풍부한 기술자가 정밀한 장비와 전문화된 환경에서 작업을 한다 하더라도 오류가 없는 다이들 (dies) 로만 구성된 웨이퍼를 만드는 것은 매우 어렵다. 따라서 결함이 발생했을 때 어떠한 공정 과정에 이상이 있는지를 신속하게 파악하고 문제를 대응하는 것이 필수적이다. 다행히 웨이퍼 맵에서 오류가 있는 다이들이 형성하는 결함 패턴 별로 특정 공정 과정의 이상 징후와 연관되어 있기 때문에 결함의 패턴을 정확하고 신속하게 검출하는 기술은 오류가 없는 웨이퍼 생산에 도움을 줄 수 있다. 본 연구에서는 웨이퍼 맵에 다양한 결함 패턴들이 혼재하고 있는 상황에서 결함 패턴 별로 높은 정확도와 재현율을 보장하는 딥러닝 기반의 다중 결함 검출 기술을 구현한다. 해당 기술을 위해 벡터 표현 기반의 캡슐 네트워크를 사용하여 모든 결함 패턴에 대한 mAP 가 피쳐 맵 표현 기반의 네트워크들에 비해 최대 65% 의 성능 이득이 있음을 보인다.

Key Words: Vector representation, CapsNet, Learning, Machine Learning, Deep Learning

ABSTRACT

To satisfy the demand of semiconductor market, the semiconductor manufacturing process should guarantee the production of high-yield and high-quality semiconductors. The fabrication process is complexly compound of several sub-processes, and thus even if an experienced engineer manages the process with precise equipment in a clean environment, it is difficult to make a wafer with no error-free dies. Therefore, the engineer should quickly discover which sub-process is mal-functioning for high yield. Fortunately, error dies make a specific defect pattern, which corresponds to specific abnormal operation of some fabrication sub-processes. Hence, a scheme that recognizes defect patterns in a wafer map can allow fabrication engineers to make the high-quality wafer with few error dies. In this paper, we implement a capsule network based multiple-defect recognition scheme with high precision and recall per each defect pattern. This work is the first to exploit a vector-representation based network for the recognition of defects in a wafer map, and verifies that the network using vector representation shows higher performance compared to the conventional feature-map based networks.

※ 이 연구는 금오공과대학교 대학 학술연구비로 지원되었음(2021)

• First Author : Kumoh National Institute of Technology, Department of Semiconductor System Engineering, ms9497@kumoh.ac.kr, 학생회원

^o Corresponding Author : Kumoh National Institute of Technology, the School of Electronic Engineering, hrlee@kumoh.ac.kr, 정회원

* Kumoh National Institute of Technology, Department of Semiconductor System Engineering, 20236121@kumoh.ac.kr, 학생회원
논문번호 : 202304-070-C-RN, Received April 4, 2023; Revised June 13, 2023; Accepted June 27, 2023

I. 서 론

최근 시장 전망을 예측하는 기관의 보고서에 따르면 반도체 시장의 크기가 2022년부터 2029년까지 연평균 성장률이 12.2% 정도 될 것으로 예측 되고 있다. 최근 챗 GPT와 같은 AI 기반 서비스들은 대량 연산이 가능한 고성능 고용량 메모리 반도체를 필수로 요구한다. 이렇듯 다양한 분야에서 반도체의 수요가 급증함에 따라 발생한 반도체 공급량 부족 현상으로 인해 반도체 공정 과정은 고품질, 고수율의 제품 생산을 보장해야만 한다.

반도체 공정의 경우 산화공정, 포토공정, 식각공정, 이온주입 공정, 금속배선공정 등 다양한 화학적, 기계적, 전기적 공정들로 복잡하게 구성되어 있다. 이는 공정 경험이 풍부한 기술자가 정밀한 장비와 깨끗하고 전문화된 환경에서 공정을 진행하더라도 오류가 없는 다 이들 (dies) 로만 구성된 웨이퍼 맵 (wafer map) 을 만드는 것은 매우 어렵다.

모든 공정을 거친 후 만들어진 웨이퍼에 있는 다이들의 동작 유무를 판단하기 위해 회로 프로브 측정 (circuit prob test) 을 통해 웨이퍼 맵을 생성한다. 웨이퍼 맵에서 오류가 있는 다이들은 공간적으로 특정한 패턴을 보이며 이러한 패턴은 반도체 공정 과정 중 특정이상 징후와 연관이 있을 수 있다.^{[1],[2]} 예를 들어, 중심 (Center) 패턴의 경우 화학적 기계적 연마 (chemical mechanical polishing) 공정에서, 가장자리 (edge local) 패턴은 박막 증착 (thin film deposition) 에서 그리고 가장자리원 (edge ring) 패턴은 식각 공정에서의 문제가 있음을 암시할 수 있다.^{[3],[4]}

따라서 웨이퍼 맵의 결함 패턴을 분석하는 것은 반도체 공정 과정에서 발생하고 있는 비정상적 처리를 실시간으로 인지할 수 있게 하고 즉각적으로 대응함으로써 웨이퍼 맵의 수율을 향상 시킬 수 있다.

웨이퍼 맵의 결함 패턴을 판단하는 좋은 대안은 딥러닝을 기반으로 한 기법이 될 수 있다.^{[5],[6],[7],[8]} 이를 바탕으로 본 연구에서는 웨이퍼 맵에 존재하는 다중 결함 패턴을 정확히 인지하기 위한 딥 러닝 기반의 기법을 연구한다. 특히 기존에 많이 사용하는 스칼라 출력 특징 검출기 (scalar output feature detector) 가 아닌 벡터 출력 검출기 (vector output detector) 를 사용하는 것이 웨이퍼 맵의 다양한 결함 패턴을 검출하는데 더 용이하다는 것을 보인다.

II. 데이터 셋

본 연구에서는 MixedWM38 데이터 셋을 이용했다.^[9] 웨이퍼 맵 내의 결함이 있는 다이들의 공간적 분포에 따라 결함 패턴이 결정된다. 단일 결함 패턴은 9개로 Center, Donut, Edge Local, Edge Ring, Local, Near Full, Random, Scratch, Normal이 있다. 다중 결함 패턴은 단일 결함 패턴 중 2개 이상의 패턴이 동시에 나타나는 것을 의미하며, 29개를 가지고 있다. MixedWM38은 총 38개의 단일 및 다중 결함 패턴을 가지고 있다. 그림 1은 MixedWM38에 있는 38가지의 결함 패턴 중 단일 결함 9개와 다중 결함들 중 일부 3개를 보여준다. 이미지의 한 픽셀은 웨이퍼 맵에서 하나의 다이를 의미하고, 짙은 파란색으로 표시된 것이 오류가 있는 다이를 의미한다. 그림 1(a)를 보면 오류가 있는 다이가 웨이퍼 맵의 가운데에서 원 형태로 모여있기에 Center 패턴이라 한다. 본 논문에서는 단일 결함 패턴의 이름을 줄여서 C, D, EL, ER, L, NF, R, S, N이라 칭한다.

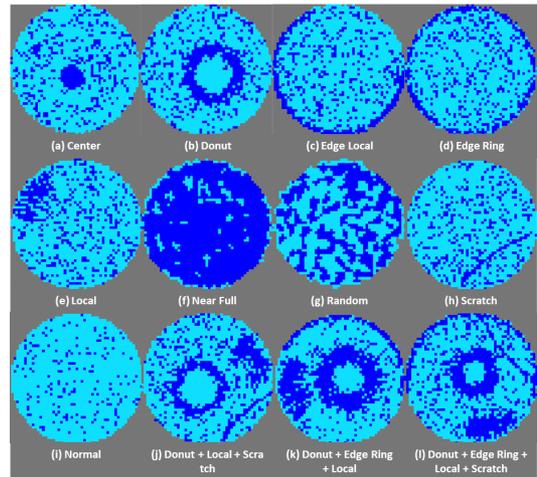


그림 1. MixedWM38 데이터 셋의 웨이퍼 맵 이미지.
Fig. 1. Wafer map image of the MixedWM38 dataset.

III. 기존 딥 러닝 네트워크

전 결함 층 (Fully-connected layer) 에 이미지를 넣어 학습시키기 위해서는 3차원 데이터를 1차원으로 펼쳐 줘야 한다. 이 과정에서 공간 정보가 소실되어 학습이 잘 이루어지지 않게 된다. 이와 같은 문제를 해결하기 위해 이미지의 공간 정보를 유지한 상태로 학습이 가능한 합성곱 신경망 (convolution neural network) 기법이 나오게 되었다.

LeNet은 합성곱 신경망을 처음으로 활용하여 신경망 네트워크를 구성하였다.^[10] AlexNet의 경우 신경망 네트워크의 성능을 증가시키기 위해 활성화 함수로 ReLU를 사용하였다.^[11] 또한 드롭아웃 (dropout) 을 사용해 과잉적합 (overfitting) 의 문제를 해결하고자 하였으며, 학습 속도 향상을 위해 GPU를 사용하였다. VGGNet은 기본적인 합성곱 신경망의 구조에 깊이를 증가시킨 것이다.^[12] 추가적으로 모든 레이어 (layer) 에 3x3 필터 (filter) 를 이용하여 네트워크의 파라미터 (parameter) 수를 감소시켰다. GoogleNet은 인셉션 모듈 (Inception module) 을 적용하여 네트워크의 깊이를 증가시켰으며 네트워크 깊이 증가에 따라 발생할 수 있는 기울기 소실 (gradient vanishing) 문제를 완화하기 위해 중간 레이어들에 보조 분류기 (auxiliary classifier) 를 추가하였다.^[13] ResNet은 기울기 소실 문제를 해결하고자 잔여 학습 프레임워크 (deep residual learning framework) 를 사용하였다.^[14] ResNet에서의 정보 흐름 지연 문제를 해결하기 위해 DenseNet의 경우 이전 레이어의 출력이 이후에 나오는 모든 레이어에 전달될 수 있는 신경망 구조를 사용하였다.^[15] 너비 (width), 깊이 (depth), 해상도 (resolution) 사이의 균형을 맞추는 것이 네트워크 성능 향상에 매우 중요하기에 EfficientNet에서는 혼합된 스케일링 기법 (compound scaling method) 을 사용하여 주어진 환경을 고려한 최적의 너비, 깊이, 해상도를 도출하여 네트워크를 구축하였다.^[16]

앞서 설명한 LeNet, AlexNet, VGGNet, GoogleNet, ResNet, DenseNet, EfficientNet은 스칼라 출력 특징 검출기이다. 스칼라의 경우 크기는 존재 하지만 방향 정보는 존재하지 않기 때문에 스칼라 출력 특징 검출기는 물체가 가진 상대적인 위치 및 방향에 대한 정보를 표현하기 어렵다. 이와 달리 캡슐 네트워크 (Capsule Network) 의 경우에는 출력 층을 벡터 출력 검출기로 구축함으로써 물체의 상대적 위치 정보에 강인하다.^[17]

IV. 피쳐 맵 기반의 표현 vs. 벡터 기반의 표현

본 연구에서는 벡터를 기반으로 특징을 표현하는 캡슐 네트워크를 바탕으로 웨이퍼 맵에서 다중 결합 패턴을 인지하는 방식을 연구한다. 또한 벡터 표현 기반 방식의 네트워크 성능을 비교하기 위해 피쳐 맵 표현 기반 방식으로 특징을 추출하는 네트워크들인 ResNet, DenseNet, EfficientNet을 이용하였다.

벡터 표현 기반과 피쳐 맵 표현 기반 네트워크들의 구분은 네트워크 내부에서 추출되는 특징 들을 다루는

형태에 따라 구분할 수 있다. 먼저 피쳐 맵 표현 기반의 네트워크들은 추출되는 특징 값 들을 3차원 블록 형태로 다루며 이러한 블록을 피쳐 맵이라 한다. 네트워크 내부의 최종 출력 층에서 사용되는 특징이 피쳐 맵 형태이면 이를 피쳐 맵 표현 기반 네트워크라 한다. 반면 특징 값 들을 벡터의 형태로 다룬다면 이러한 네트워크 들은 벡터 표현 기반 네트워크라 한다.

피쳐 맵 표현 기반 네트워크들의 최종 출력 층 결과는 각 클래스 별 예측이 하나의 스칼라 값으로 구성되어 있기 때문에 이를 스칼라 출력 특징 검출기라 한다. 벡터 표현 기반 네트워크의 출력 층에서는 각 클래스 별로 예측값이 하나의 벡터이므로 이를 벡터 출력 검출기라고 한다.

4.1 피쳐 맵 표현 기반 네트워크들

본 연구에서 비교 대상 피쳐 맵 표현 기반 네트워크 들로 ResNet, DenseNet, EfficientNet을 고려하며 해당 4.1 절에서는 해당 네트워크들의 특징에 대해 간략히 설명한다. 그림 2(a)는 ResNet의 기본 구조인 잔여 블록 (residual block) 을 보여준다. 해당 블록의 주요 특징

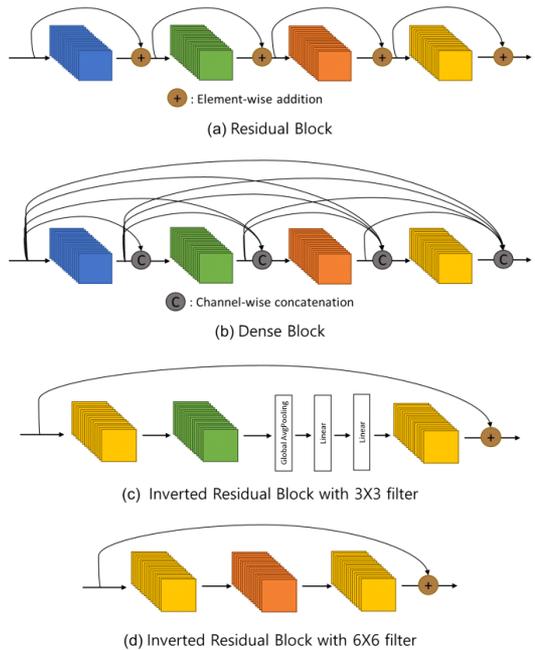


그림 2. 본 연구에서 피쳐 맵 표현 기반 네트워크들을 구성하는 기본 블록:(a) 잔여 블록, (b) 밀집 블록, (c) 3X3 필터를 이용한 반전된 잔여 블록, (d) 6X6 필터를 이용한 반전된 잔여 블록

Fig. 2. The basic block constituting feature map representation-based networks in this study:(a) Residual Block, (b) Dense Block, (c) Inverted Residual Block with a 3X3 filter, (d) Inverted Residual Block with a 6X6 filter

은 입력값을 직접적으로 출력값에 전달하는 우회 연결 (short connection) 이다. 우회 연결을 통해 입력값과 출력값이 더해지고 해당 결과가 다음 레이어의 입력값으로 들어가게 한다. 그림 3(a)는 본 연구에서 사용한 ResNet 구조를 보여준다. 해당 ResNet은 VGGNet 구조를 기반으로 잔여 블록들을 추가 연결하여 구성된다.

그림 2(b)는 DenseNet의 기본 구조인 밀집 블록 (dense block) 을 보여준다. 해당 블록은 순방향 전파 (forward propagation) 시 모든 층의 피쳐 맵들을 연결한다. 즉 이전 층의 피쳐 맵을 이후의 모든 층에 직접 연결하여 네트워크 내부의 정보 및 기울기의 흐름을 향상시킨다. 또한 밀집 블록은 정규화 효과를 가지고 있고 과잉 적합이 감소한다. 그림 3(b)는 본 연구에서 사용한 밀집 블록 기반 DenseNet 구조이다.

그림 2(c)와 (d)는 EfficientNet을 구성하는 기본 구조들을 보여준다.^[18] 너비, 깊이, 해상도 사이의 균형을 맞추으로써 네트워크의 성능을 향상시킬 수 있다. 각 층의 너비의 증감에 따라 정확도와 연산량 사이의 트레이드오프가 존재한다. 네트워크의 깊이에 따라 특징 값 표현의 복잡성과 기울기 소실 문제 간의 트레이드오프가 있으며 마지막으로 입력 이미지의 해상도에 따라서는 특징 값 들의 정확성과 연산량 사이에서 트레이드오프가 발생한다. 이에 EfficientNet에서는 위 3개의 트레이드오프를 고려하는 혼합된 스케일링 기법을 통하여 최적의 너비, 깊이, 해상도를 도출하였다. 그림 3(c)는 본 연구에서 사용한 EfficientNet의 구조를 보여준다.

4.2 피쳐 맵 기반 네트워크의 한계점

앞서 설명한 바와 같이 하나의 컨볼루션 층에서 사용하는 필터의 수가 많아질수록 네트워크의 정확도를 향상시킬 수 있다. 필터 수의 증가는 피쳐 맵 수의 증가로 이어지고 이는 표현 가능한 특징의 차원이 증가한다는 것으로 이로 인해 네트워크가 복잡한 특징들을 잘 표현할 수 있게 된다.

필터 수의 증가에 의한 파라미터 수의 증가는 필연적으로 연산량을 증가시킨다. 이를 방지하기 위해 합성곱 신경망에서는 풀링 층 (pooling layer) 을 이용한다. 특히, 최대값 풀링 층 (max pooling layer) 을 사용하여 분류에서 공간적 불변성 (spatial invariance) 을 얻을 수 있다.

하지만 최대값 풀링 층을 거치면 물체 혹은 특징의 상대적인 위치와 방향에 대한 정보를 잃어버린다. 풀링 층에 의해 상대적인 위치와 방향에 대한 정보를 포함하지 않는 특징이 추출되면 임의의 하나의 물체가 가지고 있는 다양한 부분들이 독립적으로 추출이 될 수는 있지만 독립적으로 추출된 부분들 간 복합적으로 이루어지는 관계 정보를 잃어버리게 된다.^[19] 이는 결과적으로 분류 성능을 저하시킬 수 있다. 얼굴로 예를 들면 기존의 피쳐 맵 표현 기반의 네트워크들은 얼굴의 각 요소인 눈, 코, 입, 귀 등으로부터 특징을 독립적으로 추출해낼 수 있지만, 각 얼굴 요소들의 상대적 위치 정보는 잃어버릴 수 있기 때문에 추출된 눈, 코, 입, 귀로부터 어떠한 사람인지를 분류하는 성능은 저하될 수 있다.

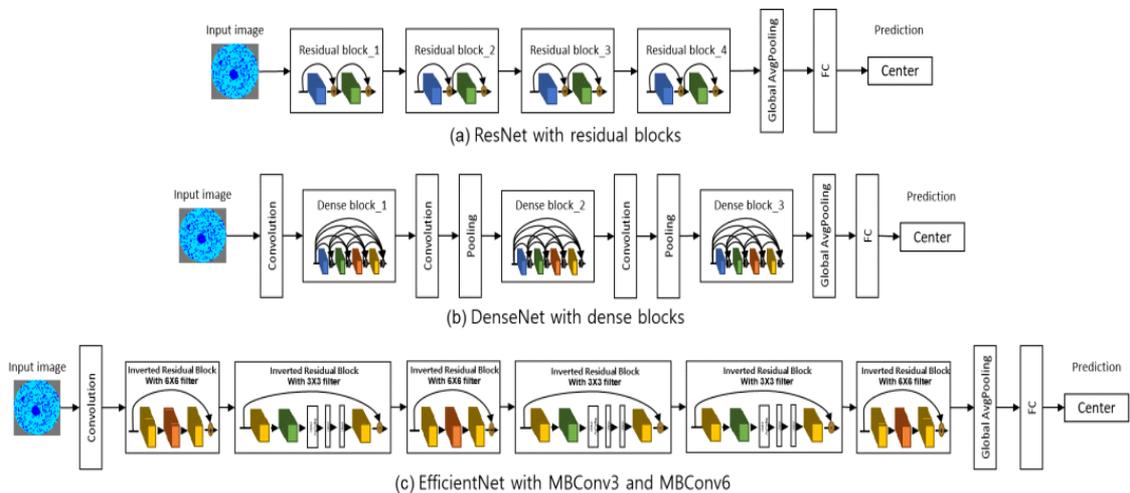


그림 3. 본 연구에서 비교 대상으로 사용된 피쳐 맵 표현 기반 네트워크들:(a) ResNet, (b) DenseNet, (c) EfficientNet
 Fig. 3. Feature map representation-based networks used for comparison:(a) ResNet, (b) DenseNet, (c) EfficientNet

4.3 벡터 표현 기반 네트워크: 캡슐 네트워크

피쳐 맵 표현 기반의 네트워크들이 가지는 문제점을 해결하기 위해 본 연구에서는 벡터 표현 기반의 네트워크인 캡슐 네트워크를 사용한다.

캡슐 네트워크는 피쳐 맵 표현 기반의 네트워크들의 한계점을 극복하기 위해 출력 층을 스칼라 출력 특징 검출기에서 벡터 출력 검출기로 변경하였다. 캡슐 네트워크에서 캡슐은 여러 개의 뉴런 들로 이루어진 1차원 벡터이다. 캡슐을 얻기 위해서는 컨볼루션 층의 연산 결과인 피쳐 맵을 여러 개의 캡슐 들로 분할한다. 하나의 캡슐에 있는 뉴런 들은 이미지 내에 있는 물체의 위치, 회전, 크기 등의 특징을 학습한다. 그림 4(a)에서 볼 수 있듯이 네트워크 초반부의 컨볼루션 층에 의해 피쳐 맵을 생성하고 해당 피쳐 맵을 캡슐이라는 벡터 단위로 나누고 다이나믹 라우팅 (dynamic routing) 으로 처리하여 최종 결과를 도출한다. 캡슐 네트워크의 최종 출력 층은 분류하고자 하는 클래스 수만큼 벡터 형태의 캡슐 들로 구성되어 있고 따라서 벡터 출력 검출기라 할 수 있다.

4.4 벡터 출력 검출기의 장점

출력 층의 캡슐 하나는 앞서 말하였듯 여러 개의 뉴런들로 구성되어 있고 각 뉴런 들은 이미지 내에 있는 임의의 물체의 크기, 형태, 위치 등의 정보를 가지고 있을 수 있다. 이러한 정보들을 포함하고 있는 출력 층의 캡슐들은 각 클래스 별 벡터값을 도출한다. 따라서 벡터 출력 검출기는 스칼라 출력 특징 검출기보다 표현력이 향상된다.

V. 웨이퍼 맵 내의 다중 결합 패턴 인지를 위한 캡슐 네트워크 적용

5.1 캡슐 네트워크 구조

그림 4(a)는 본 연구에서 사용하는 캡슐 네트워크의 구조를 보여준다. 캡슐 네트워크의 구조에서 다이나믹 라우팅 이전 층의 결과 피쳐 맵을 초기 캡슐 층 (primary capsule layer) 이라 하고 다이나믹 라우팅의 결과를 결합 캡슐 층 (defect capsule layer) 이라 칭한다. 본 연구에서는 초기 캡슐 층의 캡슐 하나를 초기 캡슐이라 하고 결합 캡슐 층의 캡슐 하나를 결합 캡슐이라 한다.

초기 캡슐 및 결합 캡슐을 구성하는 뉴런들의 수에 따라 벡터의 차원이 결정되므로 각 캡슐들의 벡터 차원은 하이퍼파라미터 (hyperparameter) 이다. 그림 4(a)에서 하나의 초기 캡슐은 8개의 뉴런들로 구성됨으로 8차원의 벡터이고 결합 캡슐은 16차원의 벡터이다.

5.2 다이나믹 라우팅

캡슐 네트워크는 피쳐 맵 표현 기반 네트워크의 단점 중 하나인 최대값 풀링 층을 거칠 때 상대적인 위치와 방향에 대한 정보를 잃어버린다는 문제점을 해결하고자 특징 값들을 캡슐이라는 벡터로 다루며 초기 캡슐 들로부터 더 복잡한 특징 값들을 예측해내기 위해 그림 4(b)에서와 같이 다이나믹 라우팅 방식을 사용한다.

다이나믹 라우팅 방식에서 사용되는 변수들은 다음과 같다. u_i 는 초기 캡슐 층의 i 번째 캡슐이고 W_{ij} 는 u_i 로부터 차원이 증가된 벡터 \hat{u}_{ji} 를 예측하기 위한

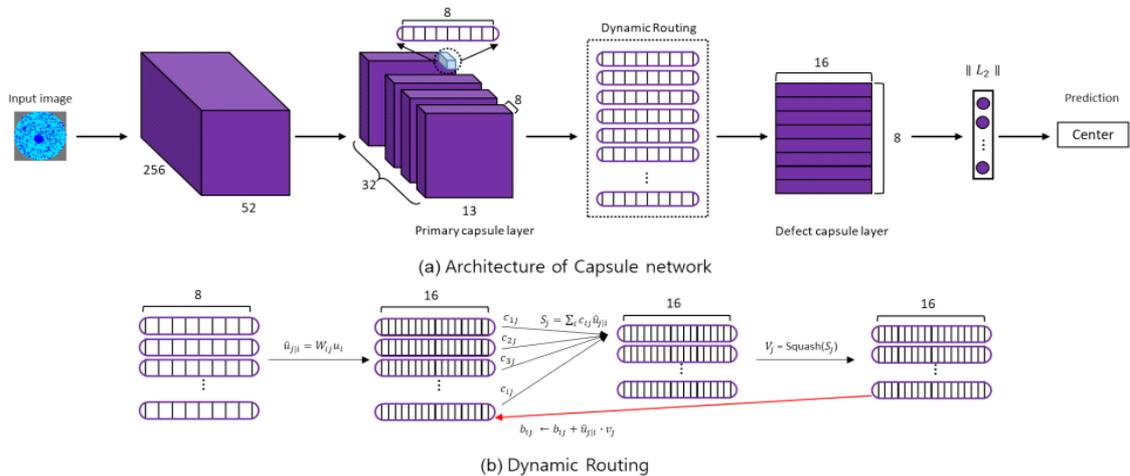


그림 4. 본 연구에서 사용된 캡슐 네트워크의 구조 및 다이나믹 라우팅
Fig. 4. Architecture of Capsule network and Dynamic Routing used in this study

가중치 값이다. 변수 \hat{u}_{ji} 는 수식 (1)과 같이 구해진다.

$$\hat{u}_{ji} = W_{ij}u_i. \quad (1)$$

즉, \hat{u}_{ji} 는 i 번째 초기 캡슐로부터 얻어진 j 번째의 차원 증가 캡슐이다. 결합 캡슐 층의 j 번째 캡슐을 s_j 라 하며 수식 (2)를 통해 구할 수 있다.

$$s_j = \sum_i c_{ij}\hat{u}_{ji}. \quad (2)$$

위 수식의 결합 캡슐 층 캡슐 s_j 를 구하기 위해 필요한 변수 c_{ij} 를 구하는 식은 다음과 같다.

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_h \exp(b_{ih})}. \quad (3)$$

수식 (2)와 (3)을 통해 b_{ij} 는 \hat{u}_{ji} 과 s_j 사이의 관계성을 의미하는 것을 알 수 있다. 마지막으로 예측 결합 캡슐 v_j 는 s_j 가 Squash 함수를 통과하면서 얻은 값으로 다음 수식 (4)를 통해 구한다.

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|}. \quad (4)$$

다이나믹 라우팅 방식은 먼저 u_i 와 W_{ij} 를 곱하여 차원을 변경한 \hat{u}_{ji} 를 구하고 c_{ij} 와 곱하여 결합 캡슐 층으로 몇 개의 캡슐들을 전달할지를 결정한다. 변수 c_{ij} 는 결합 계수 (coupling coefficient) 로 차원 증가 캡슐 \hat{u}_{ji} 가 결합 캡슐 층의 캡슐 s_j 에 미치는 영향도를 의미한다. 이후 Squash 함수를 통과하여 얻은 예측 결합 캡슐 v_j 를 구한다. 마지막으로 b_{ij} 는 아래 수식 (5)를 통해 지속적으로 업데이트되며 해당 업데이트가 다이나믹 라우팅의 핵심이다.

$$b_{ij} \leftarrow b_{ij} + \hat{u}_{ji} \circ v_j. \quad (5)$$

위 수식에서 차원 증가 캡슐 \hat{u}_{ji} 과 예측 결합 캡슐 v_j 의 내적 값으로 b_{ij} 를 업데이트한다. 이때 내적은 두 벡터의 유사도를 측정하는 것이기 때문에 이러한 갱신 알고리즘은 v_j 와 \hat{u}_{ji} 사이의 유사도가 크면 클수록 b_{ij} 가 더 커지도록 한다. 이러한 갱신 방식은 다이나믹 라우팅 알고리즘이 b_{ij} 의 두 벡터 관계성을 잘 이용할

수 있도록 해준다.

5.3 캡슐 네트워크 출력 층

캡슐 네트워크의 출력 층은 결합 캡슐들로 이루어져 있다. 결합 캡슐 하나는 단일 결합 패턴 하나에 대한 정보를 가진다. 캡슐 네트워크는 결합 캡슐 벡터의 크기로 입력 이미지에 해당 단일 결합 패턴의 존재 유무를 결정한다. 다이나믹 라우팅의 결과는 Squash 함수를 통하여 벡터의 크기가 0 과 1 사이 값으로 표준화된다. 따라서 결합 캡슐 벡터의 크기를 해당 단일 결합 패턴의 존재 확률값으로 생각할 수 있다.

5.4 결합 존재 유무 예측 방식

네트워크는 출력 층의 하나의 결합 캡슐의 크기가 하나의 주어진 문턱치 값보다 크면 해당 캡슐이 가리키는 단일 결합 패턴이 웨이퍼 맵 내에 있다고 판단한다. 이때 문턱치 값은 하이퍼파라미터로써 0에서 1 사이 값으로 주어질 수 있다. 따라서 실험 부분에서 다양한 문턱치 값에 대한 네트워크 성능을 도출한다.

5.5 데이터 셋 맞춤 네트워크 변형

본 연구에서 사용된 캡슐 네트워크의 경우 기존 캡슐 네트워크의 구조와 달리 입력 이미지를 4개의 컨볼루션 층들을 통과시켜 얻은 피쳐 맵을 초기 캡슐 층으로 사용하였다. 기존 캡슐 네트워크는 28x28 크기의 MNIST 데이터 셋을 대상으로 설계되었다.^[17] 하지만 MixedWM38 데이터 셋은 52x52 크기의 이미지이며 상대적으로 MNIST의 손글씨 데이터 셋보다 결합 패턴들이 더 다양하고 복잡하다. 따라서 기존 캡슐 네트워크 구조처럼 하나의 컨볼루션 층만 통과시켜 초기 캡슐 층을 얻게 되면 초기 캡슐들의 수가 필요 이상으로 많아지고 과잉 적합이 발생할 수 있다. 또한 층이 깊어질수록 더 복잡한 특징들을 추출할 수 있는 컨볼루션 층의 특성에 비추어 보았을 때 MNIST의 손글씨 데이터 셋에 비해 더 복잡한 특징들을 가지는 MixedWM38 데이터 셋을 잘 처리 하기 위해서는 하나 이상의 컨볼루션 층이 필요하다. 따라서 해당 연구에서는 컨볼루션 층을 네 번 사용하여 초기 캡슐 층의 크기를 줄여 초기 캡슐의 수를 감소시키고 동시에 초기 캡슐들이 더 복잡한 특징들을 포함할 수 있도록 하였다.

VI. 평 가

6.1 데이터 증대 (data augmentation)

본 연구에서 학습을 위해 사용한 MixedWM38 데이

터 셋의 경우에는 웨이퍼 맵 내의 단일 결합 패턴 별 그리고 다중 결합 패턴 별 데이터의 개수가 상이하다. 학습 시 패턴 별 데이터의 수가 다른 경우에는 데이터가 많이 있는 패턴 쪽으로 네트워크의 학습이 편향 될 수 있고, 네트워크의 일반화 능력에 문제가 발생할 수 있다. 이러한 데이터 편향 문제를 해결하기 위해 데이터 증대 (augmentation) 기법들을 활용하였다. 무작위 수평 반전 (random horizontal flip), 무작위 수직 반전 (random vertical flip), 회전 (rotation) 을 사용하여 단일 및 다중 결합 패턴 별 학습 데이터의 수를 5000개로 맞추어 주었다. 테스트 데이터 셋은 4098개를 사용한다. 본 연구에서는 입력 이미지에 크기 조정 (resize) 은 적용하지 않았다. 웨이퍼 맵에 크기 조정을 적용하여 이미지의 크기를 증가시키는 것은 실제로 존재하지 않는 다이들 이미지를 처리적으로 생성하는 것이기 때문에 물리적인 의미가 없는 증대 방식이라 생각하여 입력 이미지의 크기 조정은 배제하였다.

6.2 네트워크 학습 변수

네트워크의 입력 데이터인 웨이퍼 맵은 52x52 크기의 RGB 이미지를 사용하였다. 네트워크의 학습을 위한 변수들의 설정은 다음과 같다. 하나의 배치는 50개의 이미지로 구성되며 학습은 모든 네트워크에서 동일하게 20 epochs 동안 진행되었다. 네트워크의 최적화를 위해 최적화 기법은 Adam을 스케줄러는 cosineannealing을 사용하였다. 이때 cosineannealing은 학습률을 설정된 최소 및 최대 학습률 사이를 코사인 함수 형태로 반복적으로 변화시킨다. 따라서 본 연구에서는 최대 및 최소 학습률을 10^{-5} 과 10^{-8} 으로 설정하여 학습을 진행하였다. 또한 피쳐 맵 표현 기반의 네트워크들과 벡터 표현 기반의 네트워크는 손실 함수로 각각 교차 엔트로피 (cross entropy) 와 평균 제곱 오차 (mean squared error) 를 사용하였다.

6.3 성능 지표: AP 와 mAP

본 연구에서는 벡터 표현 기반 네트워크와 피쳐 맵 표현 기반 네트워크들의 성능을 비교하기 위해 분류 성

능 지표 중 정밀도 (precision) 와 재현율 (recall) 을 사용하였다. 정밀도는 네트워크가 참이라고 분류한 것 중 실제로 참인 것의 비율이며 재현율은 실제 참인 것 중 네트워크가 참이라고 예측한 것의 비율이다. 예를 들어 총 100개 중 20개의 데이터가 단일 Center 패턴일 때, Center가 5개 있다고 네트워크가 예측하면 Center에 대한 정밀도는 1이 된다. 반면 Center에 대한 재현율은 0.25이다. Center에 대한 정밀도는 높지만, 재현율은 낮기에 네트워크의 성능이 좋다고 판단하기는 어렵다. 일반적으로 정밀도와 재현율은 트레이드오프에 있지만 네트워크의 성능이 좋으면 모두 1에 가까워질 수 있다.

정밀도와 재현율의 값을 이용하면 네트워크 하나의 성능은 정확히 파악할 수 있지만 여러 네트워크의 성능을 비교하기에는 한계가 있다. 네트워크들을 동시에 비교해야 하는 경우에는 성능 지표 average precision (AP) 과 mean average precision (mAP) 을 이용하는 것이 일반적이다. 성능 지표 AP는 정밀도-재현율 그래프에서 그래프 아래쪽 면적을 의미하며 AP 값이 클수록 정밀도와 재현율 모두가 높은 네트워크라고 할 수 있다. 성능 지표 mAP는 AP를 클래스 수로 평균 취한 것을 의미한다.

6.4 캡슐 네트워크 변수 설정

캡슐 네트워크에서 초기 캡슐 층의 캡슐 크기 (N_{pri}), 예측 결합 캡슐 층의 캡슐 크기 (N_{pre}), 그리고 초기 캡슐 층의 채널 크기 (N_{ch}) 가 조절 가능한 하이퍼파라미터이고 표 1의 첫 번째 열은 (N_{pri} , N_{pre} , N_{ch}) 의미한다. 본 연구에서는 적절한 하이퍼파라미터 값들을 설정하기 위해 표 1에서와 같이 여러 경우 들에 대해서 실험을 진행하였다. 먼저 $N_{ch} = 256$ 으로 고정된 상태에서 나머지 두 개의 변수 ($N_{pri} = 4, N_{pre} = 12$), ($N_{pri} = 8, N_{pre} = 16$) 와 ($N_{pri} = 16, N_{pre} = 24$) 를 비교해보면 각 결합 패턴 별 AP는 유사한 결과가 나왔지만, mAP의 경우 ($N_{pri} = 8, N_{pre} = 16$) 와 ($N_{pri} = 16, N_{pre} = 24$) 가 높게 나왔다. 각 캡슐 층의 캡슐들은 이미지 내의 특징들을 학습한 것이므로 캡슐

표 1. 캡슐 네트워크의 초기 캡슐 층의 캡슐 크기와 결합 캡슐 층의 캡슐 크기 변화에 따른 AP 성능 비교
Table 1. Performance comparison with varying the size of the primary capsule and output capsule of capsule network

| CapsuleNet | AP | | | | | | | | mAP |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | C | D | EL | ER | L | NF | R | S | |
| (4, 12, 256) | 0.998 | 0.998 | 0.985 | 0.919 | 0.987 | 0.899 | 0.859 | 1.000 | 0.956 |
| (8, 16, 256) | 0.999 | 1.000 | 0.992 | 0.978 | 0.997 | 0.894 | 0.934 | 0.999 | 0.974 |
| (16, 24, 256) | 1.000 | 1.000 | 0.995 | 0.985 | 0.998 | 0.906 | 0.910 | 1.000 | 0.974 |

의 크기는 각 캡슐들이 추출하여 가지고 있는 특징 수라 할 수 있다. 일반적으로 특징 값을 많이 가지고 있을수록 네트워크는 주어진 이미지로부터 더 많은 특징들을 추출하게 되고 학습 성능이 더 좋아질 수 있다. 다만 ($N_{pri} = 8, N_{pre} = 16$) 보다 캡슐의 크기가 큰 ($N_{pri} = 16, N_{pre} = 24$) 과 동일한 mAP 성능을 보이는 데 이는 추출하는 특징의 수가 많아질수록 즉, 네트워크의 용량이 커질수록 과잉 적합에 빠질 가능성이 높아지게 되고 이러한 경우에는 오히려 네트워크의 성능을 저하시킬 수 있기 때문에 더이상 네트워크의 성능이 증가되지 않는 것이라 볼 수 있다.

네트워크의 크기와 네트워크의 처리 속도는 트레이드 오프이기 때문에 성능이 동일한 경우에는 처리 속도가 더 빠른 설정 값을 사용하는것이 바람직하다. 따라서 본 연구에서는 네트워크의 처리 속도가 더 빠른 ($N_{pri} = 8, N_{pre} = 16$) 을 캡슐 네트워크의 변수로 사용하였다.

초기 캡슐 층의 입력 데이터로 들어오는 채널의 수에 따라 초기 캡슐 층에서 만들어지는 초기 캡슐의 수가 달라진다. 표 2는 $N_{pri} = 8$ 와 $N_{pre} = 16$ 을 고정시킨 후 N_{ch} 를 변화시켜가며 캡슐 네트워크의 성능을 비교한 것이다. 합성곱 신경망에서 피쳐 맵의 채널의 수는 필터의 수와 동일한 의미를 가지고 있다. 즉, 채널의 수가 많을수록 네트워크의 용량이 증가하여 특징을 더 잘 추출할 수 있다. 앞서 설명한 바와 유사하게 채널의 수의 증가는 네트워크 용량의 증가를 의미하며 따라서 네트워크의 성능이 향상될 수 있다. 하지만 네트워크 용량의 증가는 입력 데이터로부터 편향된 학습으로 이어질 수 있으며 이는 과잉 적합을 발생시켜 네트워크의 성능을 오히려 악화시킬 수 있다. 표 2에서 채널이 증가할수록 AP, mAP가 감소하는 것은 위와 같은 사유에 의해 발생할 수 있다. 따라서, 본 연구에서는 캡슐 네트워크의 하이퍼파라미터들을 $N_{pri} = 8, N_{pre} = 16, N_{ch} = 256$ 으로 설정하였다.

6.5 피쳐 맵 표현 기반 네트워크들과 캡슐 네트워크의 성능 비교

그림 5는 본 연구에서 사용하는 피쳐 맵 표현 기반 네트워크들과 캡슐 네트워크의 성능을 비교하는 정밀도-재현율 그래프이다. 해당 그래프는 네트워크가 최종 예측 시 사용하는 문턱치 값을 0에서 1사이 값 중 100개를 선택하여 설정해주었다. 각 결합 패턴 별 x축은 재현율, y축은 정밀도를 의미한다.

그림 5에 따르면 대부분의 결합 패턴들에 대해서 캡슐 네트워크가 가장 좋은 정밀도를 보이는 것을 알 수 있다. 다만 재현율의 경우에는 피쳐 맵 표현 기반 네트워크들과 큰 차이를 보이지 않거나 일부 패턴에서 미세한 차이로 성능이 역전되는 현상을 보이기도 한다.

그림 5를 통해 개괄적인 성능 비교가 가능하지만 네트워크 별 명확한 성능 비교를 위해서는 정밀도와 재현율을 복합적으로 고려한 수치적 성능 비교가 필요하다. 본 연구에서는 표 3에서 보여주는 각 네트워크 별 결합 패턴에 대한 AP 및 결합 패턴들의 AP를 평균화 한 mAP의 값을 이용하여 수치적 성능을 비교 한다.

표 3을 보면 EL을 제외한 결합 패턴들에 대해서는 CapsuleNet이 가장 좋은 AP 성능을 보여주고 있다. EL에 대해 가장 좋은 성능을 보여주는 DenseNet과 CapsuleNet의 성능을 비교해보면 C, D의 경우에는 AP 성능의 차이가 거의 없거나 동일한 것을 확인할 수 있다. EL, L, NF, R, S의 경우에는 CapsuleNet이 DenseNet에 비해 최대 0.2이상의 성능 향상을 보여준다. 네트워크 별 결합 패턴들의 평균적인 AP 성능을 나타내는 mAP의 경우에도 피쳐 맵 표현 기반 네트워크들 보다 CapsuleNet이 최소 0.17에서 최대 0.36 차이로 더 좋은 성능을 가지고 있는 것을 확인할 수 있다.

표 4는 각 네트워크 별 4098장의 테스트 이미지를 처리할 때 걸리는 처리 속도를 보여준다. ResNet이 가장 빠르고 캡슐 네트워크는 2번째로 빠른 성능을 보여준다. 하지만 그림 5와 표 3을 통해서 캡슐 네트워크가 가장 좋은 예측 성능을 보인다는 것을 알 수 있다. 결과적으로 캡슐 네트워크가 빠른 처리 속도와 가장 좋은 성능을 보이고 있다.

표 2. 초기 캡슐 층의 캡슐의 크기와 결합 캡슐 층의 캡슐의 크기를 고정한 상태에서 채널 크기의 변화에 따른 AP 성능 비교
Table 2. Performance comparison with various channel sizes and the size of the primary capsule and the output capsule fixed

| CapsuleNet | AP | | | | | | | | mAP |
|---------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | C | D | EL | ER | L | NF | R | S | |
| (8, 16, 256) | 0.999 | 1.000 | 0.992 | 0.978 | 0.997 | 0.894 | 0.934 | 0.999 | 0.974 |
| (8, 16, 512) | 0.998 | 0.999 | 0.988 | 0.918 | 0.989 | 0.924 | 0.839 | 0.993 | 0.956 |
| (8, 16, 1024) | 0.444 | 0.401 | 0.338 | 0.520 | 0.713 | 0.140 | 0.508 | 0.168 | 0.404 |

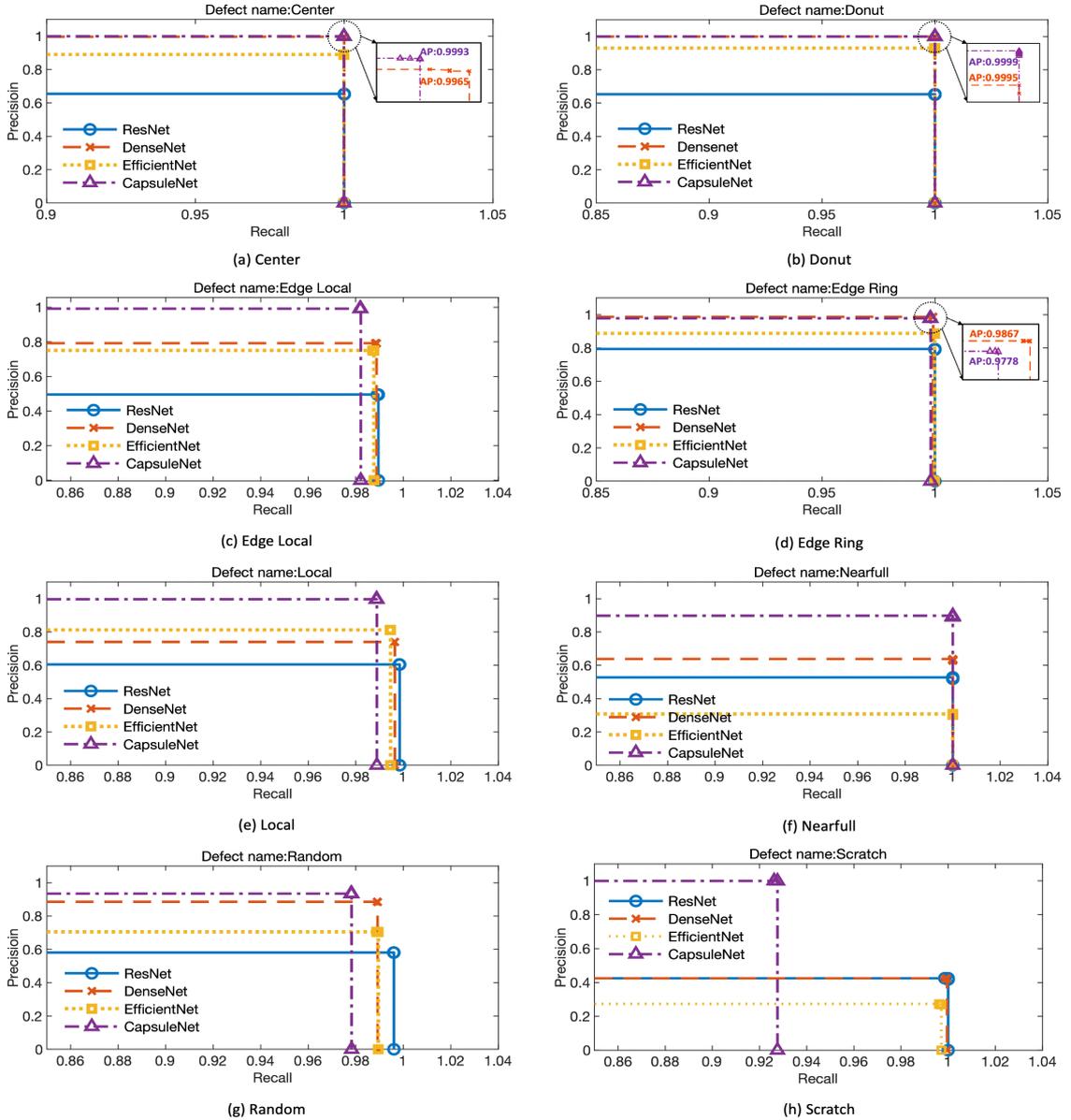


그림 5. 각 결함 패턴 별 예측 문턱치 값의 변화에 따른 각 네트워크의 정밀도-재현율 그래프
 Fig 5. Precision-recall curve of each network with various prediction threshold values for each defect pattern

표 3. 각 네트워크 별 각 결함 패턴에 대한 AP 성능 및 mAP 성능
 Table 3. AP performance and mAP performance for each defect pattern and each network

| Network | AP | | | | | | | | mAP |
|--------------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| | C | D | EL | ER | L | NF | R | S | |
| ResNet | 0.652 | 0.652 | 0.495 | 0.792 | 0.605 | 0.522 | 0.580 | 0.420 | 0.590 |
| DenseNet | 0.996 | 0.999 | 0.791 | 0.986 | 0.740 | 0.633 | 0.885 | 0.423 | 0.807 |
| EfficientNet | 0.889 | 0.929 | 0.750 | 0.886 | 0.812 | 0.305 | 0.704 | 0.271 | 0.693 |
| CapsuleNet | 0.999 | 0.999 | 0.991 | 0.977 | 0.996 | 0.894 | 0.934 | 0.999 | 0.974 |

표 4. 각 네트워크 별 처리 속도 비교
Table 4. Comparison of processing time for each network

| Network | ResNet | DenseNet | EfficientNet | CapsuleNet |
|-----------------------|--------|----------|--------------|------------|
| Processing time [sec] | 31.4 | 67.1 | 56.3 | 54.8 |

VII. 결 론

본 연구에서는 웨이퍼 맵에서 다중 결함 패턴 인지를 위해 벡터 표현 기반 네트워크인 캡슐 네트워크를 적용하였고 다른 피쳐 맵 표현 기반 네트워크들에 비해 거의 모든 경우에 대해서 성능이 좋은 것을 확인 할 수 있었다. 특히 스크래치 패턴의 경우에는 피쳐 맵 표현 기반 네트워크들보다 벡터 표현 기반 네트워크에서 훨씬 더 높은 성능을 보여주고 있다. 스칼라 출력 특징 검출기를 통과하면 각 결함 패턴 별로 가지고 있는 특징의 상대적인 위치나 방향에 대한 정보가 사라지게 되는데, 벡터 출력 검출기의 경우 정보가 사라지지 않기 때문에 더 좋은 성능이 나온 것이라 할 수 있다. 이를 통해 웨이퍼 맵에서 결함 패턴 별로 가지고 있는 공간 정보가 중요하는 것을 알 수 있었고, 피쳐 맵 표현 기반 네트워크보다 벡터 표현 기반 네트워크를 사용하는 것이 웨이퍼 맵 내의 결함 패턴을 검출하는데 용이하다.

본 연구에서 특징의 상대적인 위치나 방향에 대한 정보가 네트워크 성능에 중요하다는 것을 확인하였으며, 이는 공간 정보가 중요한 데이터 셋의 경우에는 벡터 표현 기반 네트워크 사용이 필요하다는 것을 보여준다.

References

[1] M. Saqlain, Q. Abbas, and J. Y. Lee, "A deep convolutional neural network for wafer defect identification on an imbalanced dataset in semiconductor manufacturing processes," *IEEE Trans. Semiconductor Manufacturing*, vol. 33, no. 3, pp. 436-444, Aug. 2020. (<https://doi.org/10.1109/TSM.2020.2994357>)

[2] K. Kyeong and H. Kim, "Classification of mixed-type defect patterns in wafer bin maps using convolutional neural networks," *IEEE Trans. Semiconductor Manufacturing*, vol. 31, no. 3, pp. 395-402, Aug. 2018. (<https://doi.org/10.1109/TSM.2018.2841416>)

[3] M. Fan, Q. Wang, and B. van der Waal, "Wafer defect patterns recognition based on OPTICS and multi-label classification," in

Proc. 2016 IEEE Advanced Inf. Manag., Commun., Electr. and Automat. Control Conf. (IMCEC), pp. 912-915, Xi'an, China, 2016. (<https://doi.org/10.1109/IMCEC.2016.7867343>)

[4] M.-J. Wu, J.-S. R. Jang, and J.-L. Chen, "Wafer map failure pattern recognition and similarity ranking for large-scale data sets," *IEEE Trans. Semiconductor Manufacturing*, vol. 28, no. 1, pp. 1-12, Feb. 2015. (<https://doi.org/10.1109/TSM.2014.2364237>)

[5] J. Choi, M. Kim, and H. Lee, "A study on machine learning based wafer defect detection using multi-defect recognition loss," in *Proc. Symp. KICS*, pp. 1496-1497, Jeju Island, Korea, Jun. 2022.

[6] Y. Hyun and H. Kim, "Memory-augmented convolutional neural networks with triplet loss for imbalanced wafer defect pattern classification," *IEEE Trans. Semiconductor Manufacturing*, vol. 33, no. 4, pp. 622-634, Nov. 2020. (<https://doi.org/10.1109/TSM.2020.3010984>)

[7] H. S. Shon, E. Batbaatar, W.-S. Cho, and S. G. Choi, "Unsupervised pre-training of imbalanced data for identification of wafer map defect patterns," *IEEE Access*, vol. 9, pp. 52352-52363, 2021. (<https://doi.org/10.1109/ACCESS.2021.3068378>)

[8] D. Kim and P. Kang, "Dynamic clustering for wafer map patterns using self-supervised learning on convolutional autoencoders," *IEEE Trans. Semiconductor Manufacturing*, vol. 34, no. 4, pp. 444-454, Nov. 2021. (<https://doi.org/10.1109/TSM.2021.3107720>)

[9] J. Wang, C. Xu, Z. Yang, J. Zhang, and X. Li, "Deformable convolutional networks for efficient mixed-type wafer defect pattern recognition," *IEEE Trans. Semiconductor Manufacturing*, vol. 33, no. 4, pp. 587-596,

Nov. 2020.

(<https://doi.org/10.1109/TSM.2020.3020985>)

- [10] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proc. IEEE*, vol. 86, no. 11, pp. 2278-2324, Nov. 1998.
(<https://doi.org/10.1109/5.726791>)
- [11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 26th Annu. Conf. NIPS*, pp. 1106-1114, Lake Tahoe, Nevada, United States, Dec. 2012.
- [12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd ICLR*, San Diego, CA, USA, May 2015.
- [13] C. Szegedy, L. Wei, J. Yangqing, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. 2015 IEEE Conf. CVPR*, pp. 1-9, Boston, MA, USA, 2015.
(<https://doi.org/10.1109/CVPR.2015.7298594>)
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. 2016 IEEE Conf. CVPR*, pp. 770-778, Las Vegas, NV, USA, 2016.
(<https://doi.org/10.1109/CVPR.2016.90>)
- [15] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. CVPR*, 2017.
- [16] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, PMLR, pp. 6105-6114, May 2019.
- [17] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Advances in NIPS*, 2017.
- [18] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Conf. CVPR*, 2018.
- [19] G. E. Hinton, A. Krizhevsky, and S. D. Wang, "Transforming auto-encoders," in *Proc. Artificial Neural Netw. and Mach. Learn.* -

ICANN 2011: 21st Int. Conf. Artificial Neural Networks, Espoo, Finland, Jun. 2011.

김 미 선 (Misun Kim)



2023년 2월 : 금오공과대학교 전자공학부 정보전자전공 학사
2023년 3월~현재 : 금오공과대학교 반도체 시스템공학 석사
<관심분야> 딥러닝 기반 지능형 시스템, 타국어간 영어 발음 교환 생성 모델

[ORCID:0009-0003-3146-2099]

최 지 환 (Ji Hwan Choi)



2023년 2월 : 금오공과대학교 전자공학부 정보전자전공 학사
2023년 3월~현재 : 금오공과대학교 반도체 시스템공학 석사
<관심분야> 딥러닝 기반 지능형 시스템, 지능형 신호등 상태 감지 시스템, 2D 기반 3D 이

미지 생성 연구

[ORCID:0009-0008-3520-347X]

이 하 립 (Harim Lee)



2013년 2월 : 경북대학교 전자공학과 학사
2015년 2월 : POSTECH IT 융복합전공 석사
2020년 2월 : UNIST 박사
2021년 8월 : POSTECH 박사후연구원

2021년 9월~현재 : 금오공과대학교 전자공학부 조교수
<관심분야> 딥러닝 기반 지능형 시스템, 목소리 감정 생성 모델, 목소리-얼굴 잠재 벡터 정렬 연구, 차세대 통신

[ORCID:0000-0001-8758-3250]