creative commons
COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

공학석사학위논문

반도체 결함 검출을 위한 딥러닝 모델과 합성 데이터 생성

A Semantic Segmentation Network and Synthetic Data Generation for Defects Detection

2020년 2월

서울대학교 대학원
컴퓨터공학부
손 장 현

반도체 결함 검출을 위한 딥러닝 모델과 합성 데이터 생성

A Semantic Segmentation Network and Synthetic Data Generation for Defects Detection

지도교수 이 재 진

이 논문을 공학석사 학위논문으로 제출함

2019년 12월

서울대학교 대학원

컴퓨터공학부

손 장 현

손장현의 석사학위논문을 인준함

2020년 1월

위 원 장	하 순 회	(인)
부위원장	이 재 진	(인)
위 원	이 창 건	(인)

요약

DRAM 칩의 수율은 DRAM 제조 회사의 수익성과 직접 관련이 있다. 다양한 원인에 의해 메모리 셀(cell)에 결함이 발생한다. 발생 원인에 따라 결함 있는 셀들은 다양한 패턴을 형성한다. 결함이 있는 셀들의 패턴을 감지하고 분석함으로써 DRAM 생산 공정 과정과 칩 설계상의 문제를 식별할 수 있다. 결과적으로 DRAM 제조 회사는 파악된 문제를 해결함으로써 더 나은 수율(Yield)을 달성할 수 있다. 그러나, 기존의 검출 방법을 사용하여 모든 결함의 모양과 위치를 정확하게 검출하는 것은 매우 어렵다.

본 논문은 클러스터 결함을 정확하게 검출하기 위한 Semantic Segmentation Network (SSN)를 제안한다. 제안된 SSN은 결함 있는 셀이 클러스터 결함에 속하는지 판별한다. 따라서 클러스터 결함의 정확한 형태와 위치를 검출할 수 있다.

DRAM 생산 공정은 먼지 발생 및 유입을 줄이기 위해 청정실(Clean room) 환경에서 수행된다. SSN을 학습시키기 위한 충분한 실제 클러스터 결함 데이터를 얻는 데 오랜 시간이 걸린다. 따라서 DCGAN을 기반으로 한 심층 신경망을 제안하여 실제 클러스터 결함을 흉내 내는 합성 학습 데이터를 생성할 것이다. 합성 학습 데이터에 포함된 결함은 실제 클러스터 결함과 매우 유사하다.

실험은 제안된 SSN 네트워크를 통해 클러스터 결함을 정확히 검출할 수 있음을 보여준다. 실험 결과를 통해 또한 DCGAN 기반 DNN에 의해 생성된 합성 학습 데이터가 SSN의 성능을 크게 향상시킨다는 것을 알 수 있다.

주요어 : Semantic segmentation network, 반도체 결함 검출, 합성 데이터 생성
학번 : 2018-20534

목차

제 1 장 서론 1

제 2 장 관련 연구 6
2.1 반도체 결함 검출 6
2.2 Semantic Segmentation Network..... 7
2.3 Synthetic Data Generation..... 8

제 3 장 클러스터 결함 학습 데이터 생성 10
3.1 DCGAN 기반 DNN 구조..... 10
3.2 실제 결함 비트맵 모방..... 13
3.3 DCGAN 기반 DNN 학습..... 13

제 4 장 클러스터 결함 검출 15
4.1 SSN 구조..... 15
4.2 SSN 학습..... 17

제 5 장 실험 및 결과 분석 18
5.1 데이터셋 및 학습..... 18
5.2 실험 결과 및 분석..... 19

제 6 장 결론 25

참고문헌 26
Abstract 30

그림 목차

그림 1 Failure bitmap과 여러 결함의 예 3

그림 2 세그멘테이션 결과 4

그림 3 합성 데이터 예시 14

그림 4 SSN 구조 15

그림 5 SSN의 loss 변화 20

그림 6 세그멘테이션 결과 예시 23

표 목차

표 1 결함들의 일반적 패턴 2
표 2 DCGAN 기반 생성 네트워크 11
표 3 DCGAN 기반 구별 네트워크 11
표 4 데이터셋 구성 19
표 5 실험 결과 21
표 6 기존 검출 방식과 제안한 방식 비교 24

제 1장 서론

DRAM 생산과 같은 반도체 생산 공정에서 수율(Yield)은 모든 테스트를 통과하고 올바르게 작동하는 웨이퍼의 다이(Die)의 비율을 의미한다. 그것의 정의는 다음과 같다.

$$yield = \frac{N_{working}}{N}$$

여기서 N은 웨이퍼의 총 다이 수이고, $N_{working}$은 그 중 온전한 다이의 수이다. 수율은 반도체 회사의 수익성과 직접 관련이 있다. 수율이 증가하면 한 웨이퍼에서 생산될 수 있는 정상 칩의 개수가 증가한다. 따라서 수율을 높이는 것은 반도체 회사들의 중요한 목표이다[2].

DRAM 생산 공정 과정 중, 각 메모리 셀은 EDS (Electrical Die Sorting) 테스트 공정을 통해 결함이 검사된다. DRAM 제조 공정에서, 결함이 있는 메모리 셀은 다양한 이유로 발생한다. 그것들은 다양한 발생 원인[9, 10]에 따라 여러 가지 패턴을 형성한다. 결함이 있는 셀의 패턴을 분석함으로써 DRAM 제조 공정과 칩 설계상의 문제를 식별할 수 있다. 결과적으로, 반도체 회사는 결함 있는 셀들의 패턴을 분석하여 더 나은 수율을 얻을 수 있다[5, 9].

DRAM 블록(Block)은 DRAM 칩에서 가장 큰 반복 메모리 Instance로 정의된다. DRAM 블록의 비트 라인(Bit-line)은 열로 함께 연결되어 배열된 복수의 메모리 셀들을 의미하며, 마찬가지로, 워드 라인(Word-line)은 함께 연결되어 행으로 배열된 복수의 셀들을 의미한다. Failure bitmap은 DRAM 블록의 결함 정보를 저장하는 데 사용된다. Failure bitmap은 결함이 있는 셀 위치에 1을, 결함이 없는 셀 위치에 0을 저장한다.

메모리 결함 패턴은 단일 원인에 의해 발생하며 하나의 결함을 의

미한다[10]. 결함은 DRAM 블록에서 결함이 있는 메모리 셀의 분포로 정의된다. 결함 패턴은 일반적으로 6가지 대표적인 패턴 유형으로 분류할 수 있다[1, 9, 10]. 그것들은 표 1에 설명되어 있다. 그림 1은 failure bitmap의 몇 가지 예를 보여준다. 여기에는 클러스터 결함, 워드 라인 결함, 비트 라인 결함, 단일 비트 결함 및 트윈 비트 결함이 포함된다. 클러스터 결함은 빨간색 상자로 둘러싸여 표시된다.

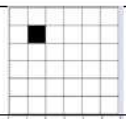
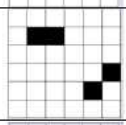
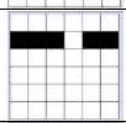
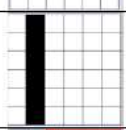
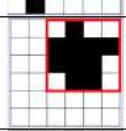
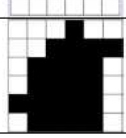
Pattern	Definition	Image
Single-bit failure	A single faulty cell without a neighboring faulty cell.	
Twin-bit failure	Two adjacent faulty cells.	
Word-line failure	A word line where the fail-cell density is higher or equal to a predefined value (e.g. $\geq$ 50%).	
Bit-line failure	A bit line where the fail-cell density is higher or equal to a predefined value (e.g. $\geq$ 50%).	
Cluster failure	Densely distributed faulty cells in a rectangle. The density is higher or equal to a predefined value(e.g. $\geq$ 70%).	
Block failure	Most of the cells in the memory block are faulty.	

표 1 결함들의 일반적인 패턴

기존의 결함 패턴 탐지 기술[9, 12, 23]은 결함 패턴에 미리 정의된 규칙을 사용하며, 규칙과 일치하는 결함이 있는 셀들의 집합을 찾는다. 기존의 검출 기술을 사용하여 단일 비트, 트윈 비트, 워드

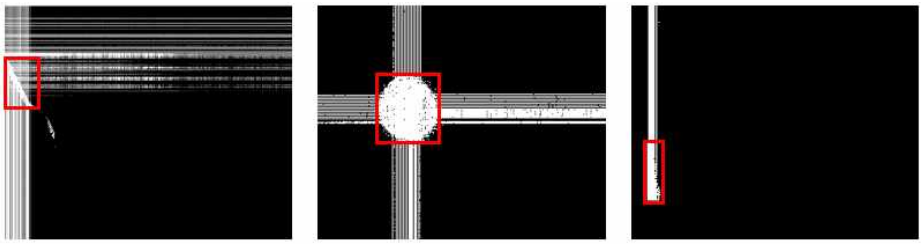


그림 1 Failure bitmap과 여러 결함의 예

라인 및 비트 라인 결함을 쉽게 감지할 수 있다. 더욱이, 다른 5가지 결함 패턴이 잘 검출되는 한, 블록 결함도 비교적 쉽게 검출할 수 있다. 그러나, 기존의 검출 기술을 사용하여 클러스터 결함을 검출하는 것은 매우 어렵다. 클러스터 결함은 반도체 제조 공정 중에 미세 먼지와 같은 이물질이 웨이퍼에 떨어지면서 발생한다. 이물질의 모양은 예측할 수 없으므로 클러스터 결함 패턴에 대한 규칙을 올바르게 정의하기가 매우 어렵다. 제대로 된 규칙을 만들 수 없으므로 기존의 규칙 기반 검출은 클러스터 결함을 제대로 검출할 수 없다.

또한, 그림 1의 예와 같이 클러스터 결함과 선형 결함(워드 라인과 비트 라인 결함)은 겹쳐져 있을 수가 있다. 겹쳐진 선형 패턴으로 인해 클러스터 결함은 쉽게 잘못 감지 될 수 있다. 나아가, 여러 선형 결함이 밀접하게 위치하면 기존의 검출 기술은 그것을 클러스터 결함으로 쉽게 오검출할 수 있다. 따라서, 기존의 검출 기술은 클러스터 결함 패턴을 정확하게 검출하지 못한다. 기존 검출 방식[9, 21]은 선형 결함을 먼저 검출한 후 남은 결함 셀들에 대하여 클러스터 결함 검출을 진행한다. 따라서 그림 1의 클러스터 결함(빨간 사각형 안)들은 선형 결함으로 검출된다.

이는 결함의 원인을 추론하는 데 방해가 된다. 하나의 클러스터 결함을 여러개의 다른 결함으로 판단하거나, 여러 개의 선형 결함을 하나의 클러스터 결함으로 판단할 수 있다. 따라서, 반도체 생산 공정에서 수율을 향상시키기 위해서는 클러스터 결함 패턴의 정확한

검출이 필수적이다.

본 논문은 클러스터 결함을 정확하게 검출하기 위한 Semantic Segmentation Network (SSN)를 제안한다. Semantic segmentation 문제는 이미지 픽셀 당 분류 문제라고 생각할 수 있다[17]. 제안된 SSN은 주어진 failure bitmap의 각 비트(메모리의 셀)가 클러스터 결함 패턴에 속하는지를 결정한다. 그림 2는 semantic segmentation을 사용하여 클러스터 결함을 검출한 결과를 보여준다. 우리가 아는 한, 이 논문은 심층 신경망을 사용하여 칩 수준에서 DRAM 셀 결함을 검출하는 첫 번째 논문이다.

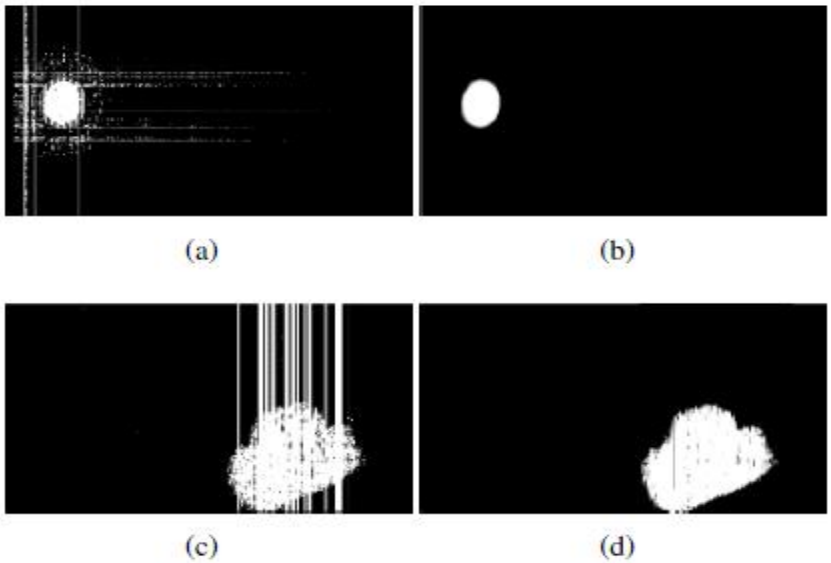


그림 2 세그멘테이션 결과

반도체 생산 공정은 먼지 발생 및 유입을 줄이기 위해 청정실 환경에서 수행되므로 일반적으로 클러스터 결함에 대한 많은 양의 학습 데이터를 얻는 데 시간이 오래 걸린다. 또한, 학습 데이터에는 전문가가 직접 셀 하나하나에 대하여 레이블을 지정해야 한다. 이 문제를 해결하기 위해 실제 클러스터 결함을 닮은 합성 학습 데이터 생성 방법을 제안한다. 합성 데이터는 실제 클러스터 결함 패턴

을 흉내 낸다. 실험에서 가짜 데이터를 통해 SSN이 성공적으로 학습될 수 있음을 보인다. 우리는 DCGAN (Deep Convolutional Generative Adversarial Network)[18] 기반의 DNN 네트워크를 구성하여 클러스터 결함의 특성을 배우고 클러스터 결함 패턴을 재현한다.

이 논문의 기여는 다음과 같이 요약된다. 1) 클러스터 결함에 속하는 메모리 셀을 올바르게 분류하는 SSN을 제안한다. 2) DCGAN 기반 DNN을 이용한 합성 클러스터 결함 패턴 생성 기법을 제안한다. 3) DRAM 제조 공정에서 생성된 실제 결함 데이터를 사용하여 제안된 방식이 실용적이고 효과적임을 보여준다.

240개의 실제 학습 데이터를 통해 학습한 SSN의 평균 IOU는 79.5 %이지만 240개의 실제 학습 데이터에 7820개의 합성 학습 데이터를 추가하면 평균 IOU가 84.1%로 향상된다. 실제 학습 데이터 수를 710으로 늘리면 평균 IOU가 82.1%가 된다. 710개의 실제 학습 데이터에 7820 개의합성 학습 데이터를 추가하면 평균 IOU가 88.4%로 크게 향상된다.

제 2장 관련 연구

2.1 반도체 결함 검출

기존의 결함 패턴 검출 방법은 미리 정의된 규칙에 따라 결함이 있는 셀의 패턴을 분류한다[12, 23, 21]. 이러한 검출 방식은 규칙 기반이므로 규칙이 정의되지 않은 결함을 감지할 수 없다. 클러스터 결함의 경우에는 그것의 규칙을 정의하기 어렵다. 일반적으로 직사각형 범위 내에 일정 개수 이상의 셀이 불량일 경우 클러스터 결함으로 간주한다[1, 9, 10]. 그렇기에 전통적인 규칙 기반 검출 방식을 사용하면 여러 결함을 하나의 클러스터 결함으로 판단하거나, 클러스터 결함을 여러 클러스터 결함으로 나누어 검출할 수 있다. 전통적인 검출 방식으로는 클러스터 결함을 올바르게 검출할 수 없다.

Li et al. [9]는 규칙 기반 방법과 인공 신경망 기반 방법을 결합한 결함 검출 기술을 제안한다. 먼저 규칙 기반 방법으로 워드 라인 및 비트 라인 결함과 같은 크기가 큰 결함을 감지한다. 그런 다음 나머지 결함이 있는 셀에 대하여, 일정 거리 이내의 인접한 결함 셀들을 하나의 클러스터로 정의한다. 그 클러스터는 정의되지 않은 결함으로 간주한다. Self-organizing map[7]은 특성의 유사성 (e.g.크기 및 종횡비)에 따라 정의되지 않은 결함을 학습하고 비슷한 특징의 결함들을 클러스터링한다. 이렇게 학습된 SOM을 규칙으로 사용하여 규칙 기반 방법으로 검출되지 않은 나머지 결함들을 분류한다. 이 방법은 선형 결함을 먼저 감지하므로 클러스터 결함을 여러 워드 라인 또는 비트 라인 결함으로 오해할 수 있다.

좌표 거리 기반 기술[10]은 정의되지 않은 결함에 대한 규칙을 자동으로 정의한다. Li et al. [9]에 의해 제안된 기술과 같이, 사전 정의된 거리 내의 인접 결함 셀 클러스터를 정의되지 않은 결함으로 정의한다. 정의되지 않은 대부분 결함에 속하는 공통 특징 (e.g. 종

횡비 및 인접한 결함 있는 셀 수)을 찾은 후 공통 특징을 사용하여 새 규칙을 정의한다. 새로운 규칙을 통해 규칙 기반 검출을 실행한다. 이 접근법은 여전히 클러스터 결함을 감지하기 위한 규칙(공통 특징의 예)을 정의해야 한다.

Nakazawa et al. [14]은 심층 컨볼루션 신경망을 사용한 웨이퍼 테스트 기법을 제안한다. 칩 수준 결함(i.e. 단일 결함으로 인한 결함이 있는 셀의 분포)이 아니라 웨이퍼 수준 결함 (i.e. 웨이퍼의 결함 분포)을 분류한다. 또한, 결함의 유형에만 중점을 두고 위치 정보는 제외한다.

이전 방법과 달리, 이 방법은 SSN을 사용하여 클러스터 결함의 정확한 모양 및 그것의 위치를 감지한다. 따라서 클러스터 결함에 대한 규칙이 필요하지 않으며, 여러 선형 결함을 클러스터 결함으로 잘못 검출하지 않는다.

2.2 Semantic Segmentation Network

Semantic segmentation network (SSN)[11, 17, 19, 24]는 주어진 입력 이미지에 대해 픽셀 당 분류를 수행한다. Fully convolutional network(FCN)[11]는 전통적인 SSN보다 우수한 성능을 제공한다. FCN은 SSN의 모든 연결 계층(e.g. fully connected layer)을 컨볼루션 계층(Convolution layer)으로 대체한다. 또한, 입력 이미지의 세부 정보를 보존하기 위해 Skip layer를 도입하여 풀링 계층(Pooling layer)으로 제거되는 입력 이미지의 세부 정보를 보존하였다.

U-Net[19]은 이미지에서 컨텍스트(Context)를 추출하는 추출 경로와 추출 경로에서 추출된 컨텍스트를 기반으로 세그먼트 맵을 생성하는 확장 경로로 구성된다. 확장 경로는 기존의 SSN보다 더 많은 레이어를 포함한다. 이를 통해 확장 경로에서보다 정확한 지역화가 가능하다. 또한, U-Net은 학습 데이터가 충분하지 않을 경우, 데이터를 Augment 하여 SSN을 성공적으로 학습하는 방법을 제안한다.

Ros et al.[20]는 실제 도시 환경과 유사한 가상 데이터셋을 생성하였다. 나아가, 생성한 합성 데이터를 사용하여 자율 주행을 위한 SSN을 학습할 수 있음을 보여준다. 가상 데이터셋은 실제 데이터에 대한 노동 집약적인 데이터 레이블링이 필요하지 않다고 설명한다. 따라서 가상 데이터셋은 실제 데이터를 수집하는 것에 비해 실용적이라고 주장한다.

Peng et al.[17]은 분류 대상이 이미지의 넓은 영역에 분포되어 있을 때 커널 크기가 클수록 분류 정확도가 더 우수하다고 설명한다. 그들은 수용 필드(Receptive field)의 크기를 증가시킴으로써 객체 분류가 개선됨을 실험적으로 보여준다.

Yu et al.[24]는 SSN의 지역화 정확도를 낮추는 풀링 단계를 사용하는 대신 확장된 컨볼루션 레이어를 사용하여 이미지의 세부 정보를 보존하는 동시에 큰 객체를 학습하는 기술을 제안한다.

우리의 SSN은 FCN을 기반으로 한다. U-Net과 마찬가지로 지역화를 강화하기 위하여 확장 경로를 강화한다. 또한, 결함 bitmap의 넓은 영역을 차지하는 선형 결함과 클러스터 결함을 구분하기 위해 기존의 SSN보다 큰 크기의 커널을 사용한다.

2.3 Synthetic Data Generation

GAN (Generative Adversarial Network) [3]은 Goodfellow et al.에 의해 제안되었다. 이들은 생성 및 구별 네트워크에 대한 적대 학습 기반 학습 방법을 제안한다. 성공적으로 학습된 생성 네트워크는 실제 데이터와 유사한 데이터 인스턴스를 생성한다. 예를 들어, GAN은 사람 얼굴 생성 및 이미지-이미지 변환 등 다양한 분야에서 실제 데이터와 유사한 가짜 데이터를 생성할 수 있음을 보여준다[3, 18, 25, 8].

Radford et al. [18]은 DCGAN이라는 컨볼루션 GAN 아키텍처를 제안한다. 원래 GAN에서 모든 fully connected 계층을 컨볼루션 단

계로 바꾼다. 이것을 통해 기존 GAN의 학습 단계에서 발생하는 불안정성 문제를 해결한다. 결과적으로 DCGAN은 기존의 GAN이 처리할 수 있는 것보다 더 복잡한 이미지를 처리한다.

우리는 DCGAN을 이용하여 가짜 클러스터 결함을 포함하는 합성 비트맵을 생성한다. 클러스터 결함에 대한 실제 학습 데이터가 부족한 상태에서도 이렇게 생성한 가짜 데이터를 통해 SSN을 성공적으로 학습시킬 수 있음을 보여준다.

제 3장 클러스터 결함 학습 데이터 생성

반도체 생산 공정 중 웨이퍼에 외부 입자가 떨어져 생기는 클러스터 결함의 모양은 매우 다양하다. 클러스터 결함의 모양은 예측할 수 없지만, 기본적으로 그림 1의 첫 번째 이미지에 나타난 것처럼 대체로 둥근 모양[13]을 띠고 있다. 하지만 여전히 다양한 형태를 띨 수 있다. 클러스터 결함은 DRAM 블록의 어느 곳에서나 발생할 수 있으며 크기 또한 다양하다. DRAM 칩 상에서 다른 결함 (e.g. 선형 결함)과 겹쳐진 위치에 존재할 수 있다.

반도체 생산 공정은 일반적으로 먼지의 발생 및 유입을 억제하는 청정실 환경에서 수행되어 외부물질에 의해 발생하는 결함을 방지하고자 한다. 결과적으로 실제 반도체 생산 공정에서 클러스터 결함에 대한 충분한 학습 데이터를 얻는 데 시간이 오래 걸린다. 또한, 학습 데이터에는 전문가가 직접 클러스터 결함에 속하는 셀들을 일일히 표시(Labeling)해야 한다. DNN을 성공적으로 학습하려면 충분한 학습 데이터를 제공해야 하고, 학습 데이터가 많을수록 DNN의 성능이 향상될 수 있다[19]. 또한, 학습 데이터의 수가 적으면 DNN이 과적합 문제를 일으킬 수 있다[22]. 이 경우, DNN을 실제 테스트 공정에 도입할 시 새로운 형태의 클러스터 결함을 검출하는 정확도가 떨어질 수 있다.

3.1 DCGAN 기반 DNN 구조

본 논문에서는 SSN을 학습하기 위한 합성 클러스터 결함을 생성하는 기술을 제안한다. 생성된 가짜 클러스터 결함은 실제 클러스터 결함과 유사하다. 이 장에서는 가짜 클러스터 결함을 생성하는 DCGAN 기반 DNN을 설계하고 구현하는 방법에 대하여 설명한다.

DNN 아키텍처의 생성기 네트워크(Generator)는 가짜 클러스터 결

Operation Layer	No. of Filters	Size of Each Filter	Stride	Size of Output (C*W*H)	Batch Norm.	Activation
Input				8*33*24		
Transposed Conv.	128	4*4	2	128*66*48	✓	ReLU
Conv.	128	3*3	1	128*66*48	✓	ReLU
Transposed Conv.	64	4*4	2	64*132*96	✓	ReLU
Conv.	64	3*3	1	64*132*96	✓	ReLU
Transposed Conv.	32	4*4	2	32*264*196	✓	ReLU
Conv.	32	3*3	1	32*264*196	✓	ReLU
Transposed Conv.	16	4*4	2	16*528*384	✓	ReLU
Conv.	16	3*3	1	16*528*384	✓	ReLU
Transposed Conv.	1	4*4	2	1*1056*768	✓	Tanh

표 2 DCGAN 기반 생성 네트워크

Operation Layer	No. of Filters	Size of Each Filter	Stride	Size of Output (C*W*H)	Batch Norm.	Activation
Input				1*1056*768		
Conv.	128	4*4	2	128*528*384	✓	Leaky ReLU
Conv.	64	4*4	2	64*264*196	✓	Leaky ReLU
Conv.	64	3*3	1	64*264*196	✓	Leaky ReLU
Conv.	32	4*4	2	32*132*96	✓	Leaky ReLU
Conv.	32	3*3	1	32*132*96	✓	Leaky ReLU
Conv.	16	4*4	2	16*66*48	✓	Leaky ReLU
Conv.	16	3*3	1	16*66*48	✓	Leaky ReLU
Conv.	4	4*4	2	16*33*24	✓	Leaky ReLU
Conv.	4	3*3	1	16*33*24	✓	Leaky ReLU
Conv.	1	33*24	1	1*1*1		Sigmoid

표 3 DCGAN 기반 구별 네트워크

함을 생성한다. 표 2는 생성기 네트워크의 구조를 나타낸다. 표 2를 참고하면, 생성기 네트워크는 전치된 컨볼루션 레이어와 일반 컨볼루션 레이어의 반복으로 구성된다. 전치된 컨볼루션은 feature map을 2씩 업-샘플링(Up-sampling)한다. 전치된 컨볼루션 레이어 및 컨볼루션 레이어 뒤에는 배치 정규화 레이어[4] 및 ReLU 레이어가 위치한다.

DNN의 판별자 네트워크(Discriminator)는 생성기가 생성한 데이터가 실제 데이터인지 가짜인지 확인한다. 구성은 표 3에 나와 있다. 컨볼루션 레이어로 구성된다. 각 컨볼루션 레이어 다음에는 배치 정규화 레이어와 새는 ReLU 레이어가 있다.

합성된 학습 데이터를 위한 DNN 아키텍처는 기존 DCGAN[18] 기반으로 하지만 생성기에 대한 입력의 형식이 다르다. 기존 DCGAN의 생성기는 입력으로 100차원 랜덤 벡터를 받는다. 제안한 DNN의 생성기는 입력으로 8 X 33 X 24 크기의 3차원 배열을 받는다 (i.e. 채널 X 너비 X 높이). 제안한 DNN의 출력은 가짜 클러스터 결함이 포함된 failure bitmap이다.

대부분의 클러스터 결함을 포함하는 failure bitmap들은 하나의 클러스터 결함을 포함한다. 생성자 네트워크에 대한 입력으로 100차원 랜덤 벡터를 사용하면 그림 3 (a)와 그림 3 (b)와 같이 출력 비트맵에 너무 많은 클러스터 결함이 생성된다. 100 차원 랜덤 벡터에는 클러스터 결함 수 및 위치에 대한 정보를 명시적으로 담고 있지 않기 때문이다.

이 문제를 해결하기 위해 생성기 네트워크에 대한 입력으로 8 X 33 X 24 크기의 3차원 배열을 사용한다. 이 배열에는 클러스터 결함이 생길 위치가 원형으로 명시적으로 표시된다. 배열의 각 채널에는 동일한 위치에 동일한 크기의 원(i.e. 위치 표시)이 포함된다. 채널은 33 X 24 2차원 하위 배열을 의미하며 각 2차원 하위 배열에는 하나의 원이 포함된다(i.e. 위치 및 개수 표시). 원의 지름은 1에서 9까지의 임의의 홀수이다. 원의 위치는 2차원 하위 배열에서 무작위로 선택된다. 0보다 크며 1보다 작거나 같은 임의의 실숫값이 원에 포함된 하위 배열의 요소에 할당된다. 원에 포함되지 않는 배열의 각 요소에는 0이 할당된다. 생성기 네트워크는 폭과 높이 방향으로 입력 배열을 업-샘플링하기 때문에 (e.g. 33 X 24 에서 1056 X 768) 출력 비트맵에는 입력된 원이 변형된 가짜 클러스터 결함이 포함된다. 이 방식은 말을 얼룩말로 변환하는 이미지-이미지 번역 기술[25]을 참고하였다.

생성기 네트워크의 입력 배열의 다양성을 높이기 위해 입력 배열의 채널 수를 증가시켰다. 2차원 하위 배열에서 지름이 1인 원은 원 안에 하나의 임의의 값을 갖다. 마찬가지로 지름이 3인 원은 5개의

임의의 값을 가진다. 따라서 2D 하위 배열은 평균적으로 원 안에 19.4 개의 임의의 값을 가질 수 있다. 8 개의 채널이 사용되는 경우 입력 배열은 평균적으로 모든 채널의 원 안에 155.2개 (i.e. 19.4 X 8)의 임의의 값을 가질 수 있다. 이러한 입력 다양성의 증가는 출력에서 더욱 다양한 형태의 클러스터 결함을 생성한다.

DNN 아키텍처의 구별 네트워크는 클러스터 결함이 포함된 failure bitmap을 입력받으며, 입력받은 failure bitmap이 가짜 비트맵인지 실제 비트맵인지를 판단한다.

3.2 실제 결함 비트맵 모방

실제 클러스터 결함을 포함하는 failure bitmap들의 40% 이상에서, 클러스터 결함과 겹치는 위치에 선형 결함이 포함되어 있다. 따라서 실제 클러스터 결함을 포함하는 failure bitmap 환경을 모방하기 위해, 합성 failure bitmap의 클러스터 결함이 발생한 위치에 선형 결함을 무작위로 포함 시킨다. 이때, 선형 결함은 비트 라인 또는 워드 라인에서 50% 이상의 셀에 결함이 발생한 경우로 정의된다. 예를 들어, 그림 3 (e)와 그림 3 (f)는 클러스터 결함과 겹치는 위치에 추가된 선형 결함의 예시를 보여준다. 선형 결함의 개수는 임의로 선택된다.

3.3 DCGAN 기반 DNN 학습

PyTorch[16]의 Adam optimizer[6] 구현을 사용하여 DNN을 학습한다. DCGAN 기반 DNN 출력의 정확도 기준으로 이진 교차 엔트로피 손실 함수(Binary cross entropy loss function)를 사용한다. 배치(Batch) 크기는 40이다. 판별기 및 생성기 네트워크 모두에 대한 learning rate는 0.005로 고정하였다. 학습 전에, 표준 편차 0.02,

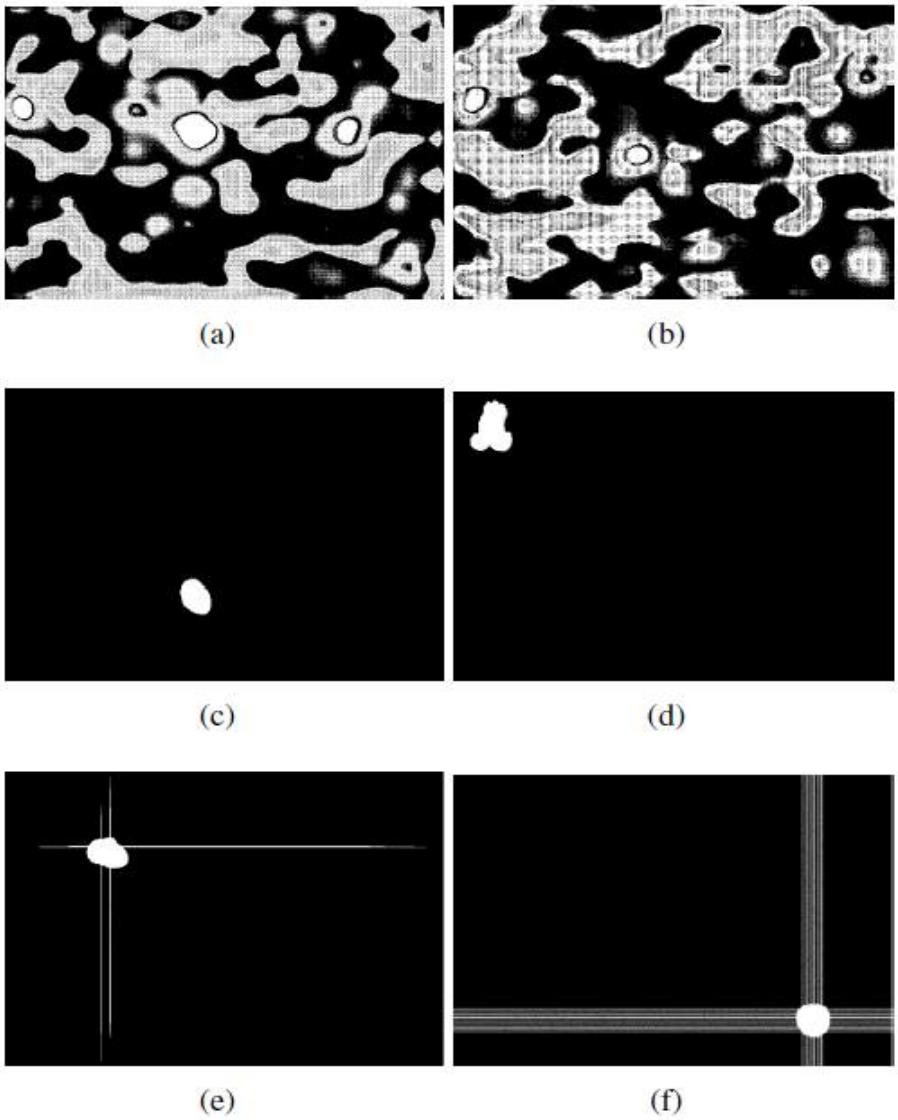


그림 3 합성 데이터 예시. (a)와 (b)는 100차원 랜덤 벡터를 DNN의 입력으로 사용한 경우. (c)와 (d), (e), (f)는 3차원 배열을 DNN의 입력으로 사용한 경우. (e)와 (f)는 DNN의 출력 결과에 임의로 선형 결함을 추가하였다.

평균 0인 가우시안(Gaussian) 분포를 사용하여 컨볼루션 레이어 및 배치 정규화 레이어의 모든 가중치를 임의로 초기화한다. 기존의 DCGAN[18]과 같은 분포를 사용하여 계층을 초기화한다.

제 4장 클러스터 결함 검출

클러스터 결함에 속하는 메모리 셀을 올바르게 분류하는 Semantic Segmentation Network (SSN)를 제안한다. SSN은 클러스터 결함의 모양과 위치를 정확하게 검출할 수 있다.

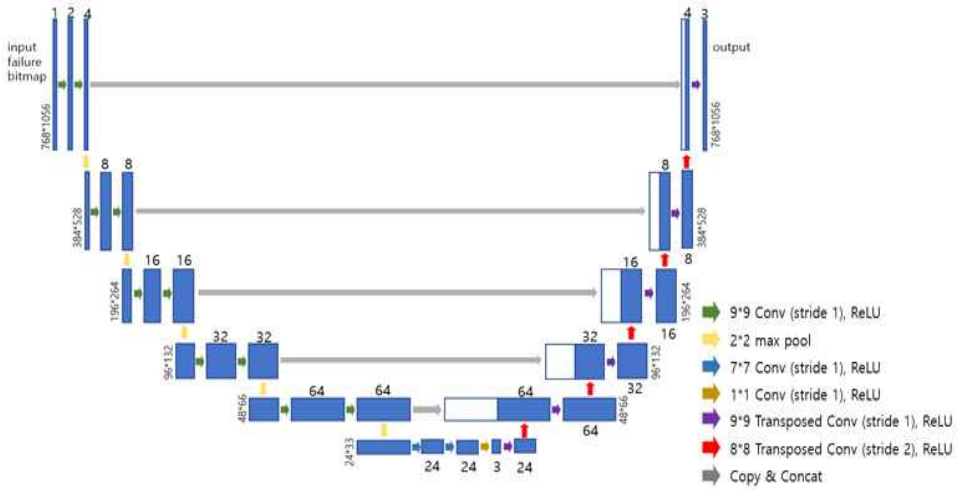


그림 4 SSN 구조

4.1 SSN 구조

그림 4는 제안된 SSN 아키텍처를 나타낸다. 입력으로 받는 failure bitmap에서 클러스터 결함 정보를 추출하는 추출 경로(그림 4의 왼쪽)와 추출된 정보를 사용하여 클러스터 결함만을 포함하는 failure bitmap을 생성하는 확장 경로로 구성된다(그림 4의 오른쪽). 그림 4의 파란색 상자는 컨볼루션 레이어의 출력 feature map을 의미한다. 흰색 상자는 추출 경로에서 복사된 출력 feature map을 의미한다. Skip layer 기법을 도입하여, 복사된 출력 feature map은 확

장 경로의 feature map과 연결되어 입력 이미지의 세부 정보를 제공한다.

추출 경로는 복수 개의 2D 컨볼루션 레이어로 구성된다. 각 컨볼루션 레이어 다음에는 ReLU 레이어가 위치한다. 모든 2개의 컨볼루션 레이어마다 feature map을 다운-샘플링하기 위해 2 X 2 Max-pooling 레이어가 있다. 컨볼루션 레이어의 커널 크기는 9 X 9이고 stride는 1이며 크기가 4인 Zero-padding이 적용된다.

확장 경로는 복수 개의 2D 전치된 컨볼루션 계층(Transposed convolution layer)으로 구성된다. 모든 전치된 컨볼루션 레이어 뒤에는 ReLU 레이어가 위치한다. 추출 경로와 유사하게, 전치된 컨볼루션 레이어의 커널 크기는 9 X 9 또는 8 X 8이다. 크기가 3인 Zero-padding은 stride 2를 가진 모든 전치된 컨볼루션 레이어에 사용된다.

stride 2인 전치된 컨볼루션 레이어는 feature map을 2씩 업-샘플링한다. 업-샘플링된 feature map은 입력 이미지의 세부 사항을 입력받기 위해 추출 경로의 대응하는 feature map과 연결된다(i.e. Skip layer).

Failure bitmap은 2차원 bitmap으로 1과 0의 정보만을 포함한다. 즉 해당 셀에 결함이 있고 없는지에 대한 정보만을 포함한다. 따라서 클러스터 결함과 다른 결함의 구분할 수 있는 정보는 결함 셀들이 이루는 분포의 모양과 밀도로 제한된다. 특히 하나의 클러스터 결함과 여러 개의 뭉친 선형 결함을 구별할 수 있는 기준은 failure bitmap 상에 길게 뻗은 모양인지 아닌지로 제한된다. 길게 뻗은 모양이라는 특징을 판단할 수 없다면 뭉쳐있는 선형 결함들을 여러 개의 클러스터 결함으로 분리하여 생각할 수 있다. 따라서 Peng et al.[17]이 제안한 방식을 활용하여 커널의 크기를 기존 U-net보다 큰 9 X 9를 사용한다. 커널의 크기를 증가시키면 receptive field가 증가하여 failure bitmap 상의 넓은 영역에 분포하는 선형 결함과 좁은 영역에 뭉쳐져 분포하는 클러스터 결함을 잘 구분할 것으로 예

상한다. 실제 실험 결과에서도 작은 크기의 커널을 사용하는 경우보다 큰 크기의 커널을 사용하면 분류 정확도가 상승하였다.

4.2 SSN 학습

제안된 SSN을 학습시키기 위해 한 쌍의 결함 비트맵과 그것의 모든 셀에 대하여 클래스(Class)를 표시한 세그멘테이션 맵(Segmentation map)이 사용된다. 결함 비트맵의 크기는 1056 X 768 이다. 세그멘테이션 맵의 크기는 결함 비트맵의 크기와 같다. 세그먼테이션 맵에서 클러스터 결함에 속하는 결함이 있는 셀은 1(클래스 1)로 레이블이 지정되고 클러스터 결함이 아닌 다른 결함에 속하는 결함이 있는 셀은 2(클래스 2)로 레이블이 지정된다. 결함이 없는 셀은 0(클래스 0)으로 레이블이 지정된다.

출력은 크기가 3 X 1056 X 768 (i.e. channel X width X height)인 3D 배열이다. 출력의 세 채널은 클래스 0, 클래스 1 및 클래스 2의 세 클래스에 해당한다. 입력 결함 비트맵의 각 셀이 특정 클래스에 속할 확률은 해당 채널의 같은 위치에 있는 셀에 표시된다. 출력에 softmax를 채널 단위로 적용하면 각 셀이 특정 클래스에 속할 확률이 계산된다. 셀은 확률이 가장 높은 클래스에 속하는 것으로 간주한다.

PyTorch의 확률적 경사 하강 최적화 (stochastic gradient descent optimizer) 구현을 사용하여 제안된 SSN을 학습한다. 네트워크 학습의 손실 판단 기준으로 교차 엔트로피 손실 함수(cross entropy loss function)가 사용된다. 배치 크기는 60이고 learning rate는 0.05로 고정된다.

제 5장 실험 및 결과 분석

이 장에서는 제안된 SSN 및 DCGAN을 기반으로 합성 학습 데이터 생성 기술을 평가한다. 평가에서 우리는 세 가지 방법으로 SSN을 학습시킨다. 첫 번째(Real)는 DRAM 생산 공정 프로세스에서 얻은 실제 클러스터 결함 데이터를 사용하여 학습하는 것이다. DRAM 제조 회사에서 클러스터 결함 비트맵과 해당 비트맵에 대한 세그멘테이션 맵을 얻었다. 두 번째(Fake)는 DCGAN 기반 DNN에서 생성된 합성된 학습 데이터만 사용하여 SSN을 학습시키는 것이다. DCGAN 기반 DNN은 DRAM 생산 프로세스에서 얻은 실제 클러스터 결함 데이터로 학습된다. 마지막으로 세 번째(Real&Fake)는 실제 학습 데이터와 합성 학습 데이터를 모두 사용하여 SSN을 학습하는 것이다.

SSN의 성능을 측정하기 위한 성능 평가 척도는 IOU (Intersect Over Union)이며 다음과 같이 정의된다.

$$IOU = \frac{T \cap P}{T \cup P},$$

여기서 T는 실제 클러스터 결함만을 포함하는 ground truth 영역이고, P는 SSN이 예측한 클러스터 결함이 있는 영역이다. IOU가 높을수록 예측한 영역과 실제 영역이 비슷함을 의미하며, 예측 성능이 좋다고 할 수 있다.

5.1 데이터셋 및 학습

SSN 학습에 미치는 실제 학습 데이터 개수의 영향력을 보여주기 위해 두 개의 데이터셋 (i.e. Small과 Large)를 사용하여 실험을 진행한다. 두 데이터셋은 실제 학습 데이터의 양이 다르며 표 4에 실

험 데이터셋의 구성을 명시하였다.

Dataset	Small	Large
Number of real training data	240	710
Number of synthesized training data	7820	7820
Number of validation data (real)	60	60
Number of test data (real)	60	60

표 4 데이터셋 구성

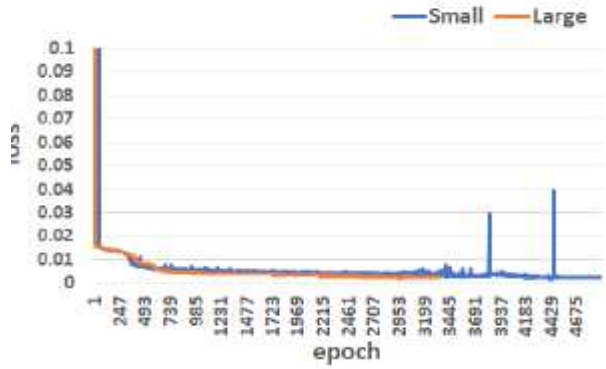
테스트 데이터셋은 60개의 failure bitmap으로 구성되며 모두 하나 이상의 클러스터 결함을 포함한다. 60개 중 21개의 failure bitmap에 선형 결함과 클러스터 결함이 겹쳐진 상태로 존재한다. 합성 학습 데이터를 생성하기 위해 DCGAN 기반 DNN은 DRAM 제조 회사에서 얻은 실제 클러스터 결함 데이터로 학습된다. Small의 합성 데이터는 Small의 real training data(240개)만을 사용하여 학습되었으며, Large도 마찬가지이다.

그림 5는 SSN이 수렴될 때까지 각각 Real, Fake 및 Real&Fake에서 SSN 검증 데이터(validation data)에 대한 loss 값의 변화를 보여준다. SSN 학습의 loss 값이 미리 결정된 기간 동안 감소하지 않으면 (e.g. SSN이 1000 epoch 미만 동안 학습된 경우 100 epoch) 중지된다. 검증 데이터셋에 대한 가장 낮은 loss 값을 보이는 모델이 선택된다.

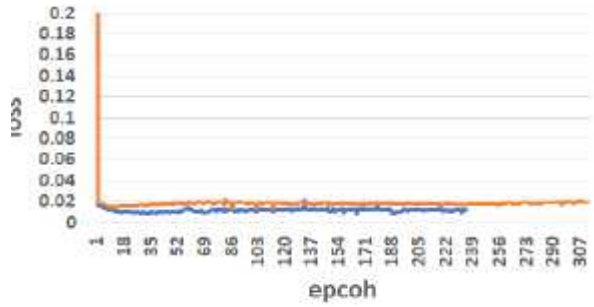
5.2 실험 결과 및 분석

표 5는 실험 결과를 요약한다. 전반적으로 Real&Fake가 가장 우수한 성능을 보여주며, Fake가 Small 및 Large 모두에 대해 최저 성능을 기록한다.

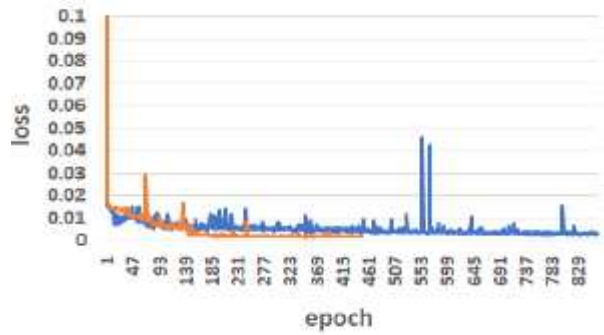
그림 6은 4개의 서로 다른 클러스터 결함을 포함하는 failure



(a)



(b)



(c)

그림 5 SSN의 loss 변화

bitmap에 대한 클러스터 결함 검출 결과를 보여준다. 첫 번째 열은

SSN에 입력되는 failure bitmap을, 두 번째 열은 그것의 세그먼테이션 맵을 의미한다. 빨간 점은 클러스터 결함에 속하는 셀을, 하얀 점은 클러스터 결함 이외의 다른 결함에 속하는 셀을 의미한다. 검은 점은 결함 없는 셀이다. 세 번째, 네 번째, 다섯 번째, 여섯 번째 열은 각각 REAL-Small, REAL&FAKE-Small, REAL-Large, REAL&FAKE-Large를 사용하여 학습했을 때의 SSN 출력 결과를 보여준다.

Smal			
	Real	Fake	Real&Fake
Mean IOU (Test)	79.5 %	49.4 %	84.1 %

Large			
	Real	Fake	Real&Fake
Mean IOU (Test)	82.1 %	64.3 %	88.4 %

표 5 실험 결과

그림 6 (a)에 볼 수 있는 것처럼, 그림 6 (a)의 failure bitmap은 4가지 학습 방법(REAL-Small, REAL&FAKE-Small, REAL-Large, REAL&FAKE-Large)에 대하여 클러스터 결함 패턴을 잘 감지한다.

그림 6 (b)에서 클러스터 결함을 구성하는 결함 있는 셀의 밀도는 클러스터 결함 내부에서 일정하지 않다. Real의 경우 결함 셀들의 밀도가 감소하는 클러스터 결함의 경계가 제대로 감지되지 않다. 반대로 Real&Fake는 밀도의 변화와 관계없이 경계를 잘 감지한다.

그림 6 (c)에서 데이터셋 Small (세 번째 및 네 번째 열)은 선형 결함과 클러스터 결함을 잘 구별하지 못한다. 클러스터 결함과 밀도가 비슷한 선형 결함들의 뭉치를 클러스터 결함으로 감지한다.

그림 6 (d)는 테스트 데이터셋(Test dataset)에서 가장 낮은 IOU를 기록한 failure bitmap이다. 선형 결함들의 그룹과 클러스터 결함을 구분하기 어렵다. 4개 학습 방식(REAL-Small, REAL&FAKE-Small, REAL-Large 및 REAL&FAKE-Large)은 모두 클러스터 결함을 완벽히 찾아내지 못하였다. 하지만 구체적인 위치는 찾

을 수 있다.

REAL-Large 및 REAL&FAKE-Small은 REAL-Small보다 향상된 성능을 보여준다. 그러나 REAL-Large와 REAL&FAKE-Small의 학습을 위한 비용은 다르다. REAL-Large는 REAL-Small보다 반도체 생산 공정 프로세스의 실제 학습 데이터가 더 필요하다. 또한, 전문가가 직접 failure bitmap의 모든 셀에 대하여 클래스를 지정해야 한다. 제안한 DNN 기반 합성 데이터 생성 방식은 학습 데이터 라벨링(labeling)이 데이터 생성 단계에서 자동으로 가능하여 시간이 덜 걸린다. 따라서 REAL&FAKE-Small의 경우, 전문가의 직접적인 도움 없이 더 적은 비용으로 학습 데이터를 추가할 수 있으며, SSN이 성공적으로 클러스터 결함의 모양을 학습하는 데에 도움을 준다.

Fake는 Real보다 나쁜 성능을 보여준다. 클러스터 결함의 고유한 특성(i.e. 미세 먼지 등에 의해 발생하여 그 모양을 예측할 수 없다.)으로 인해 클러스터 결함은 기본적으로 약간의 불규칙성을 내포하고 있다. 이에 따라 DCGAN 기반 DNN은 모든 유형의 클러스터 결함을 완전히 모방할 수는 없다. 하지만 DCGAN 기반 DNN은 클러스터 결함의 대표적인 특징(e.g. 높은 결함 셀들의 밀도, 넓게 퍼지지 않은 모양)에 중점을 두어 가짜 결함을 만든다. 이처럼 DCGAN이 재현하는 클러스터 결함은 클러스터 결함들의 공통점에 초점을 맞추어 생산되기에, Fake의 경우 학습의 한계를 보인다. 따라서 DCGAN이 재현하는 규칙성과 재현하지 못하는 불규칙성의 균형을 맞추기 위해 실제 클러스터 결함을 함께 사용하여 학습해야 한다. 또한, DCGAN 기반 DNN에서 합성된 클러스터 결함은 임의로 생성된 선형 결함과 겹치게 하였다. 따라서 클러스터 결함과 겹친 실제 선형 결함의 특성을 반영하지 못한다. 그림 6 (c) 또는 그림 6 (d)과 같은 선형 결함들의 특성을 완벽히 재현하지 못하였다.

Real&Fake는 최고의 성능을 보여준다. Real에는 학습 데이터가 매우 적으므로 과적합은 불가피하다. 결과적으로 일부 픽셀에 대하여 학습 데이터의 분포와 멀리 떨어진 형태를 보여주면, 잘못 감지

된다 (i.e. true negatives). 합성된 학습 데이터를 추가하면 학습 데이터의 범위가 증가하고 true negatives가 줄어든다. 또한, Real은 그림 6 (c)에 표시된 것처럼 클러스터 결함과 선형 결함이 겹치는 경우를 올바르게 처리하지 못한다. 이 경우 선형 결함의 뭉치를 클러스터 결함(i.e. false positives)으로 인식한다. 합성된 학습 데이터를 추가하면 SSN은 클러스터 결함의 공통된 특성(e.g. 둥근 형태, 높은 밀도)을 학습하여 검출 성능이 향상된다.

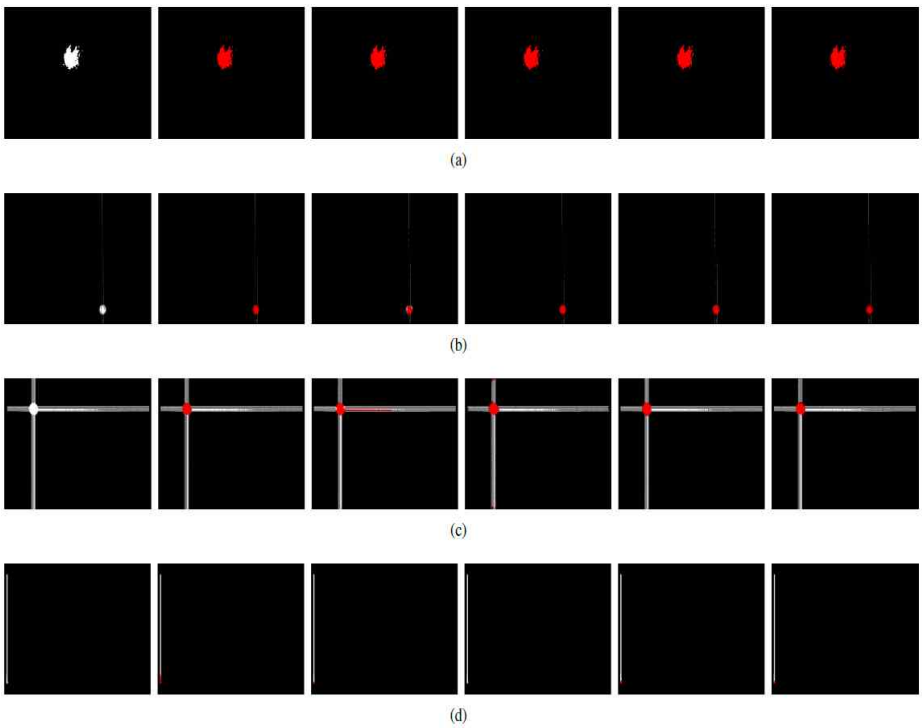


그림 6 세그멘테이션 결과

표 6은 규칙 기반 검출 방식[21]을 사용한 클러스터 결함 검출 결과와 REAL&FAKE-Large의 검출 결과를 비교한 자료이다. 규칙 기반 검출 방식을 위해, 비트 라인을 이루는 셀 중 516개 이상의 셀들에 결함 있을 때 비트 라인 결함으로, 워드 라인을 이루는 셀 중 90개 이상의 셀들에 결함 있을 때 워드 라인 결함으로 정의한다. 선형

결함으로 판단된 결함 셀들을 제외한 나머지 셀들에 대하여 클러스터 결함 검출을 시작한다. 5 X 5 크기의 정사각형 영역에 4개 이상의 셀이 결함 있을 때 클러스터 결함으로 판단한다. 표 6은 두 방식에 대하여, 테스트 데이터셋의 모든 bitmap에 대한 클러스터 결함 검출 IOU의 평균값(Test dataset)과 테스트 데이터셋의 bitmap 중 클러스터 결함과 선형 결함이 겹쳐져 있는 bitmap들에 대한 클러스터 결함 검출 IOU 평균값(Overlap)을 보여준다. 테스트 데이터셋 60개 중 20개의 bitmap에 선형 결함과 클러스터 결함이 겹쳐져 존재한다.

Mean IOU	규칙 기반 검출 방식	제안한 SSN (Real&Fake-Large)
Test dataset	76.5%	88.4%
Overlap	47.8%	82.3%

표 6 규칙 기반 검출 방식과 제안한 방식 비교

Test dataset에 대하여 규칙 기반 검출 방식은 76.5%의 IOU 정확도를 보여준다. 하지만 클러스터 결함이 선형 결함과 겹쳐 존재하는 bitmap(Overlap)에는 매우 낮은 IOU 정확도를 보여준다. 규칙 기반 방식은 결함 검출의 순서(e.g. 선형 결함 검출 이후 클러스터 결함 검출), 클러스터 결함의 밀도 기준(e.g. 5 X 5 크기의 정사각형 영역에 4개 이상의 결함 셀 존재) 등 다양한 검출 규칙을 직접 설정하여야 한다. 클러스터 결함은 그 발생 원인이 매우 다양하기에 클러스터 결함의 모든 특성을 포함하는 규칙을 정의하기 매우 어렵다. 따라서 규칙 기반 검출 방식은 클러스터 결함을 검출하기에 한계가 있다.

제안한 방식의 경우에는 클러스터 결함이 선형 결함과 겹쳐져 있더라도, 그림 6 (c)와 같이 정확한 위치를 찾을 수 있으며 클러스터 결함의 모양까지도 거의 정확히 검출해 낼 수 있다. 이는 딥러닝 네트워크가 다른 결함들과 구별되는 클러스터 결함의 여러 특성을 잘 학습하였다는 것을 의미한다.

제 6장 결론

DRAM 칩 레벨에서 클러스터 결함 패턴을 정확하게 탐지하기 위한 semantic segmentation network (SSN)를 제안한다. 클러스터 결함 패턴은 기존의 검출 기술을 사용하면 다른 결함 패턴과 구별하기가 어렵다. DRAM 생산 공정은 청정실(Clean room) 환경에서 실행되므로 SSN을 학습시키기에 충분한 양의 클러스터 결함 데이터를 수집하고 가공하는 데 오랜 시간과 큰 비용이 소요된다. 이 문제를 해결하기 위해 DCGAN을 기반으로 한 합성 학습 데이터 생성 기술을 제안한다. 우리의 실험 결과는 실제 및 합성 클러스터 결함 데이터를 함께 사용하여 SSN을 학습시킬 때 SSN이 클러스터 결함을 더욱 잘 검출하는 것을 보여준다. 240개의 실제 학습 데이터만 학습한 SSN의 평균 IOU는 79.5%이지만 240개의 실제 학습 데이터에 7820개의 합성 학습 데이터를 추가하여 학습한 SSN의 평균 IOU는 84.1%로 향상된다. 실제 학습 데이터 수를 710으로 늘리면 평균 IOU가 82.1%가 된다. 710개의 실제 학습 데이터에 7820개의 합성 학습 데이터를 추가하면 평균 IOU가 88.4%로 크게 향상된다.

참고문헌

[1] Bing-Yang Lin, M. Lee, and C. Wu. A memory failure pattern analyzer for memory diagnosis and repair. 2012 IEEE 30th VLSI Test Symposium (VTS), pages 234-239, 2012.

[2] Roger E. Bohn and Christian Terwiesch. Yield-driven processes. Journal of Operations Management, 18(1):41-59, 1999.

[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. NIPS, pages 2672-2680, 2014.

[4] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167, 2015.

[5] D. Kim, I. Pomeranz, M. E. Amyeen, and S. Venkataraman. Defect diagnosis based on dfm guidelines. 2010 28th VLSI Test Symposium (VTS) , pages 206-211, 2010.

[6] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. ICLR, 2015.

[7] T. Kohonen. The self-organizing map. Proc. of the IEEE, 78(9):1464-1480, 1990.

[8] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. CVPR, pages 105-114, 2017.

[9] J. Li, Y. Huang, W. Cheng, C. Schuermyer, D. Xiang, E. Faehn, and R. Farrugia. A hybrid flow for memory failure bitmap classification. 2012 IEEE 21st Asian Test Symposium, pages 314-319, 2012.
[10] B. Lin, C. Wu, M. Lee, H. Lin, C. Peng, and M. Wang. A local parallel search approach for memory failure pattern identification. IEEE Transactions on Computers, 65(3):770-780, 2015.
[11] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. Computer Vision and Pattern Recognition (CVPR), page 3431-3440, 2015.
[12] P. Mazumder and J. H. Patel. Parallel testing for pattern-sensitive faults in semiconductor random-access memories. IEEE Transactions on Computers, 38(3):394-407, 1989.
[13] L. S. Milor. Yield modeling based on in-line scanner defect sizing and a circuit's critical area. IEEE Transactions on Semiconductor Manufacturing, 12(1):26-35, 1999.
[14] T. Nakazawa and D. V. Kulkarni. Wafer map defect pattern classification and image. IEEE TRANSACTIONS ON SEMICONDUCTOR MANUFACTURING, 31(2):309-314, 2018.
[15] Seung Park, Sehoon Kim, and Jun-Geol Baek. Kernel-density-based particle defect management for semiconductor manufacturing facilities. Applied Sciences, 8:224, 2018.
[16] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Al-

ban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. NIPS Autodiff Workshop, 2017.
[17] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters – improve semantic segmentation by global convolutional network. arXiv:1703.02719, 2017.
[18] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434, 2015.
[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. MICCAI 2015, pages 234-241, 2015.
[20] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes. Computer Vision and Pattern Recognition (CVPR), pages 3234-3243, 2016.
[21] Sin-Yu Wei, B. Lin, and C. Wu. A fast sweep-line-based failure pattern extractor for memory diagnosis. 2016 21th IEEE European Test Symposium (ETS), pages 1-6, 2016.
[22] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1):1929—-1958, 2014.

[23] T. Viacroze and M. Lequeux. Analysis of failures on memories using expert system techniques. Proceedings.

International Test Conference 1990, pages 823-932, 1990.
[24] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. arXiv:1511.07122 , 2015.
[25] J. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. 2017 IEEE International Conference on Computer Vision (ICCV)), pages 2242-2251, 2017

Abstract

A Semantic Segmentation Network and Synthetic Data Generation for Defects Detection

Son Janghyun
Department of Computer Science & Engineering
The Graduate School
Seoul National University

The yield of DRAM chips is directly related to the profitability of DRAM fabrication companies. Faults in memory cells are caused by a variety of reasons. A different pattern of faulty cells comes from a different reason. By detecting and analyzing the patterns of faulty cells, it is possible to identify problems in the DRAM fabrication process or chip design. In turn, DRAM fabrication companies can achieve better yields by fixing the problems. However, among several categories of faulty cell patterns, it is very difficult to correctly detect cluster failures using conventional detection techniques. This paper proposes a semantic segmentation network (SSN) to correctly detect cluster failures. The proposed SSN determines if a group of faulty cells belongs to a cluster failure. Since the DRAM fabrication process

is performed in a clean room environment to reduce the generation and ingress of dust, it takes a long time to acquire real training data that are enough to train the SSN for detecting cluster failures. Thus we also propose a deep neural network based on a DCGAN to generate synthetic training data for cluster failures. The failures contained in the synthetic training data closely resemble real cluster failures. Experimental results show that our SSN detects cluster failures well even though the amount of real training data is not enough to fully train the SSN. The results also show that synthetic training data generated by the DCGAN-based DNN significantly increases the performance of the SSN.

Keywords : semantic segmentation network, defects detection, synthetic data
Student Number : 2018-20543

감사문

본 논문은 SK 하이닉스(주)의 위탁에 따른 ‘연구용역’의 결과입니다.