

Open DMQA Seminar

Introduction to Steel Surface Defects Detection

Korea University
Data Mining & Quality Analytics Lab.

안인범

2021. 5. 7

발표자 소개



- **안인범 (Inbum Ahn)**

- 고려대학교 산업경영공학과 Data Mining & Quality Analysis Lab.
- M.S. Student (2020.9 ~ Present)
- 지도 교수 : 김성범 교수님

- **Research Interest**

- Deep learning for steel surface defect recognition
- Prediction model for steel making process

- **Contact**

- E-mail : ahninb20@korea.ac.kr

Contents

- I. Steel making process
- II. Steel surface defects
- III. Data imbalance problem
- IV. Deep learning for steel surface defects detection
- V. Conclusion

I . Steel making process

❖ 철강이란 – Iron? Steel?

Iron Man



Man of **Steel** ^(Superman)



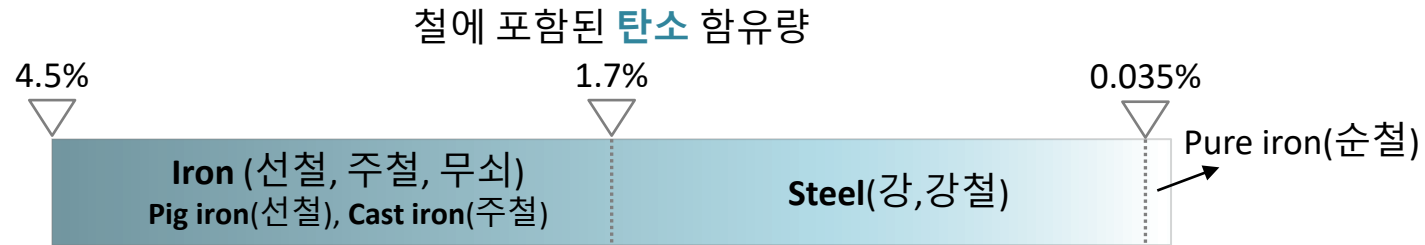
Iron Man 3 V.s Man Of Steel ... Who wins?

<http://jdsblog88.blogspot.com/2013/06/super-hero-movies-2013-iron-man-3-or.html>

I . Steel making process

❖ 철강이란 – Iron? Steel?

- 철에 포함된 탄소 함유량을 기준으로 Iron과 Steel을 구분하며, 철강은 선철(주철)과 강철을 합한 용어(Iron & "Steel")



단단함



충격에 약함
(깨짐, 부서짐)



질기고 늘어나며
충분한 강도

충격에 강함

가공성 열위

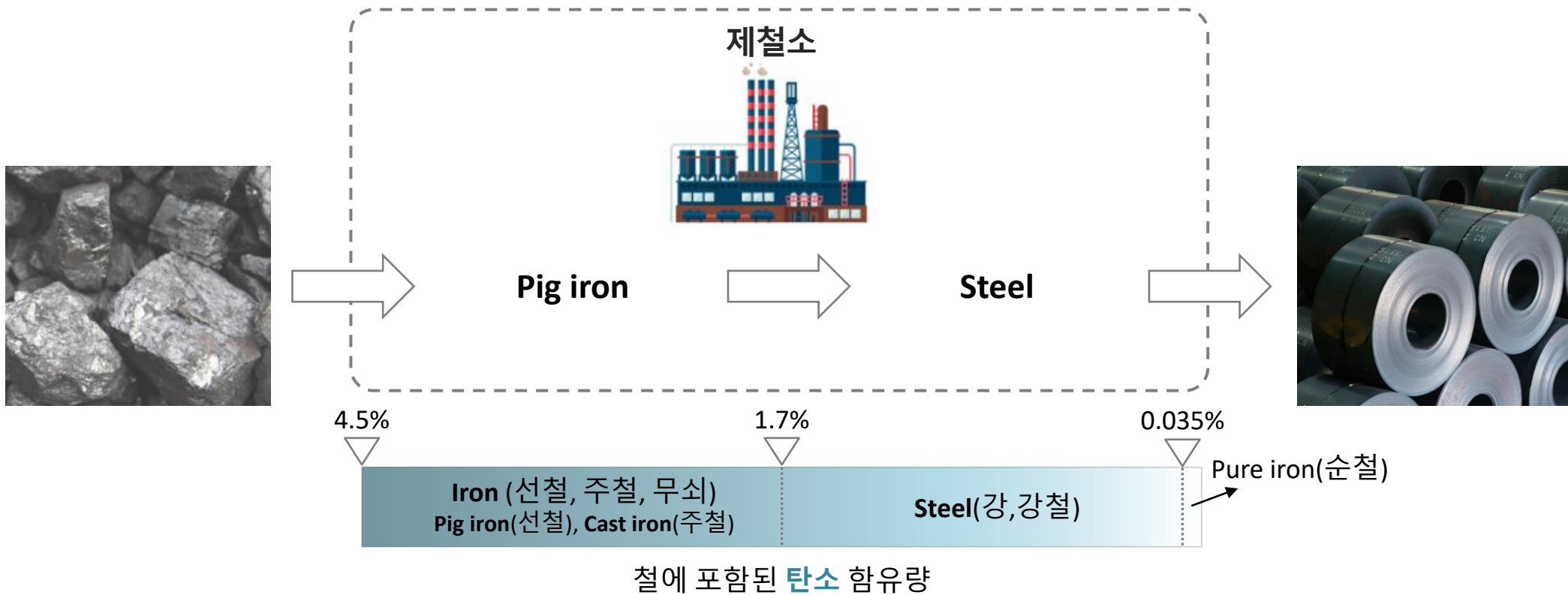


가공성 우수

I . Steel making process

❖ 철강 제조

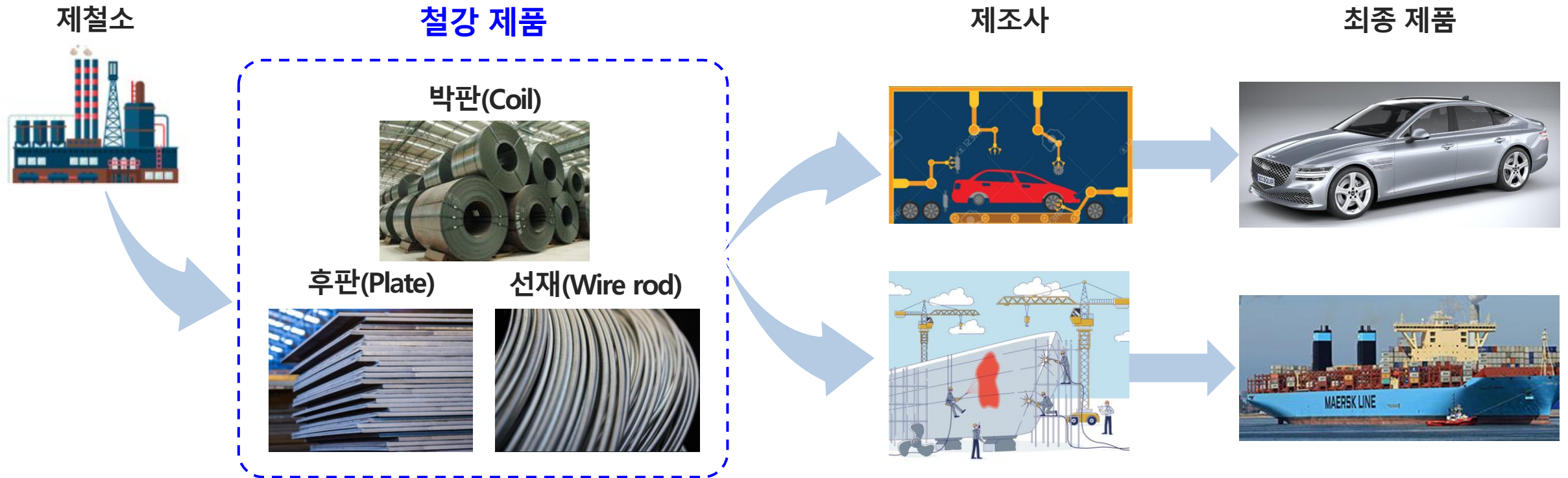
- 원재료인 철광석을 사용하여 선철(Pig iron)을 만들고, 탄소를 포함한 다양한 성분을 적정범위로 정제하여 강철(Steel)을 만드는 과정



I . Steel making process

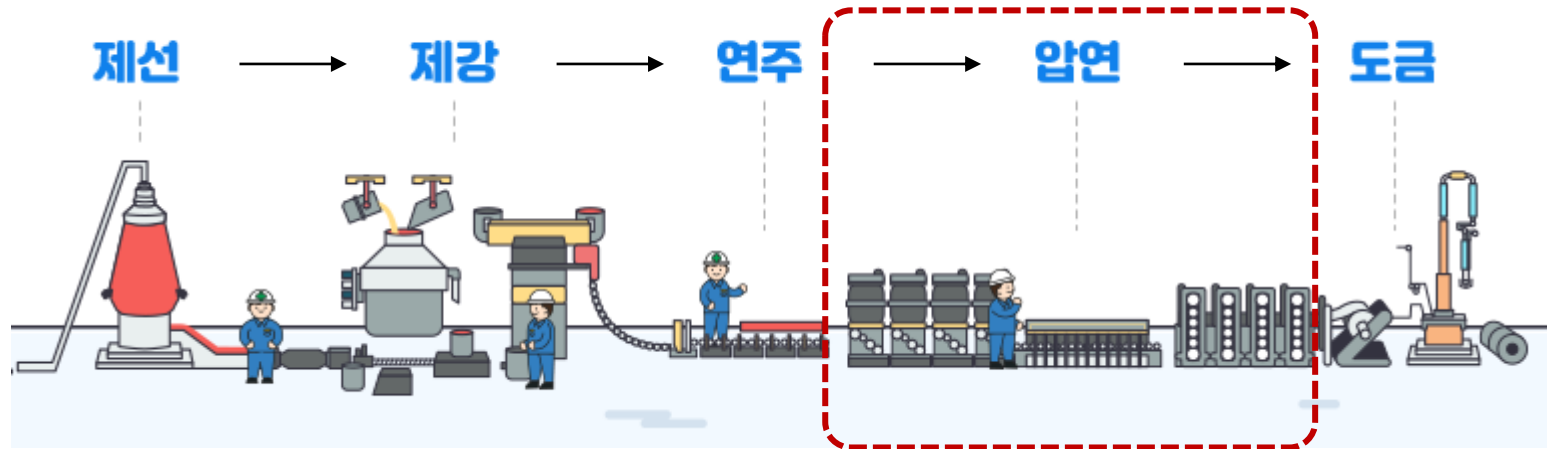
❖ 철강 제품

- 최종 산업 제품에 쓰이기 위한 소재로 철강 제품이 만들어지며, 형태에 따라 판재, 봉형강, 강관 등의 제품이 있음
- 판재는 두께에 따라 박판(얇은 강판)과 후판(두꺼운 강판)으로 크게 나눌 수 있고 박판은 Coil 형태로, 후판은 Plate 형태로 생산



I. Steel making process

❖ 철강 제조 공정



용선(선철)
Pig Iron



용강(강철)
Steel

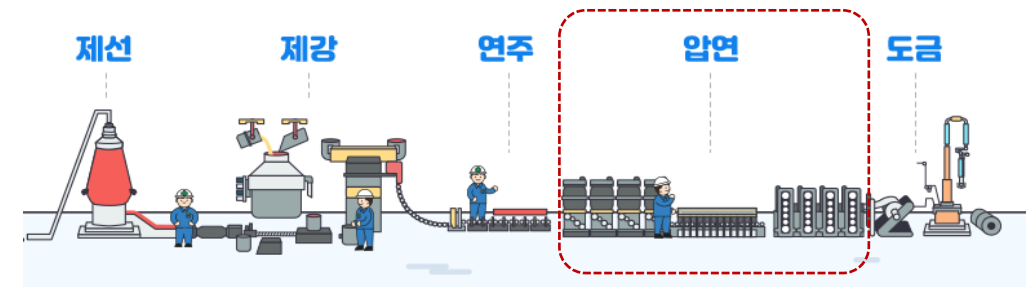


슬라브
Slab



코일
Coil

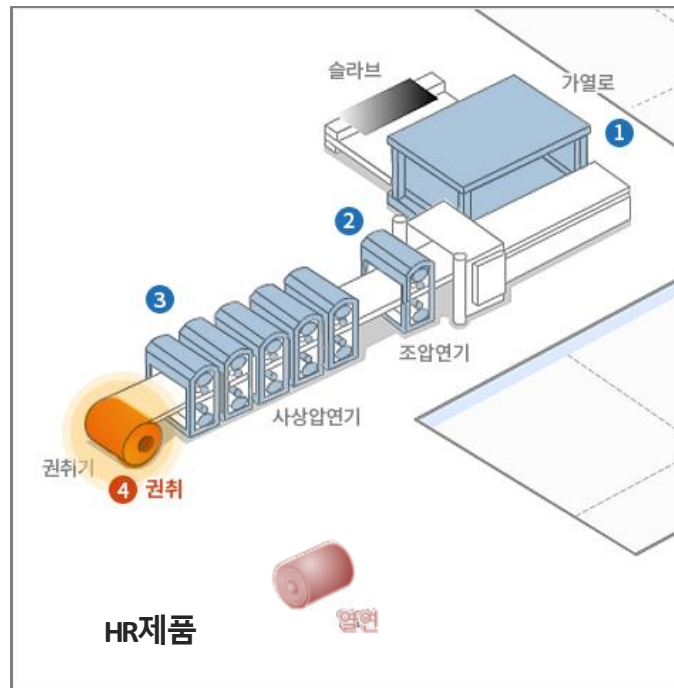
I. Steel making process



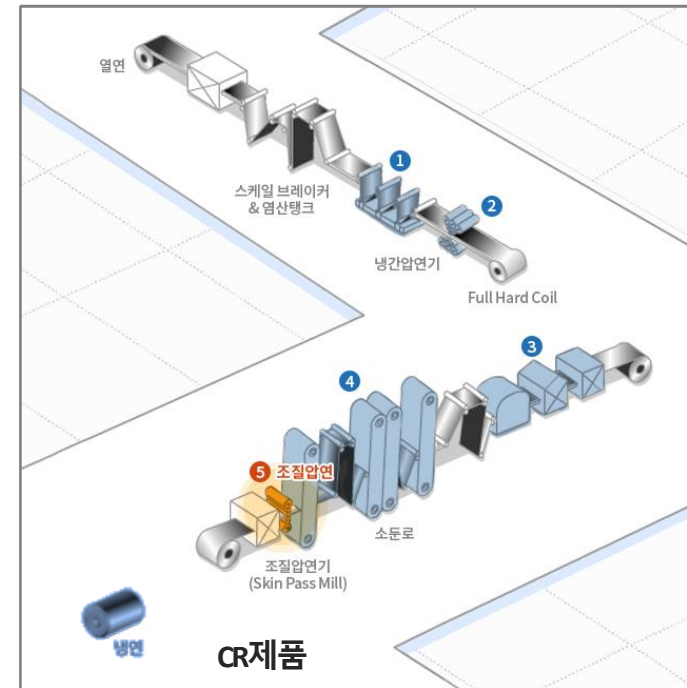
❖ 열연-냉연 공정

- 압연공정(Rolling) : 200~400mm 두께의 슬라브 소재를 두 개의 롤러 사이를 통과시키며 고객 요구 두께로 가공하는 공정
- 열간 상태에서 압연하는 열연 공정(1차 압연), 상온에서 압연을 실시하는 냉연 공정(2차 압연)이 있음
- 선공정에서의 품질 결함을 검출하지 못하고 후공정으로 보내지는 소재는 후공정에서 품질 악화, 설비 사고로 이어질 수 있음

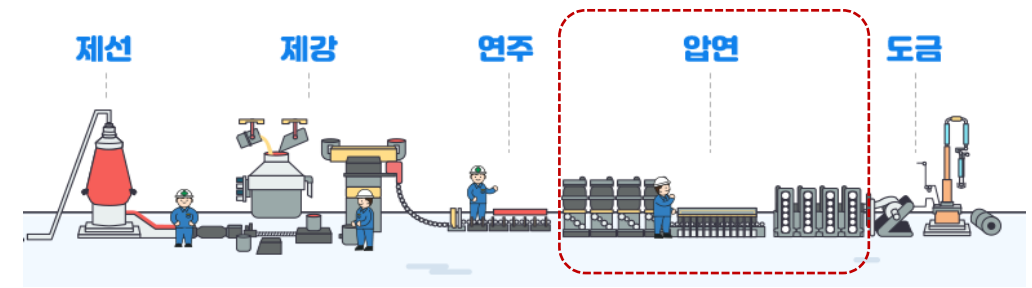
열간 압연(1차 압연)



냉간 압연(2차 압연)



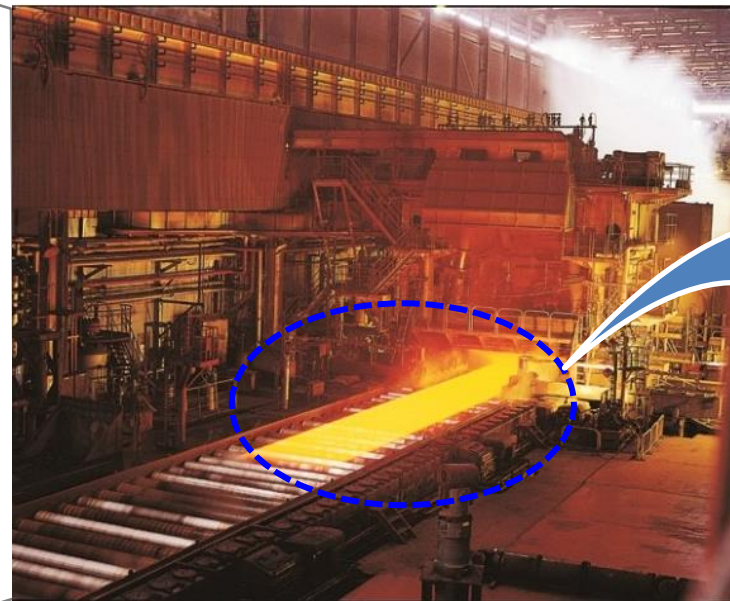
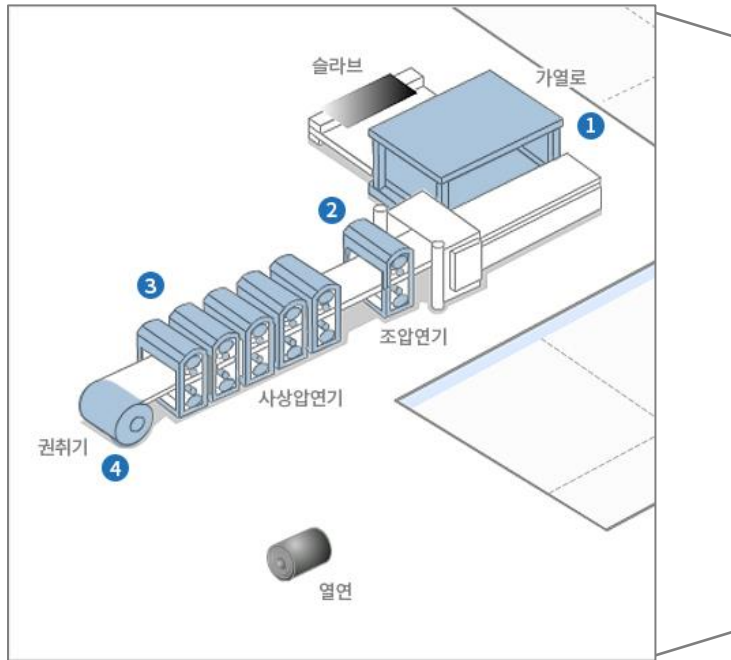
I. Steel making process



❖ Surface Defect Detector (SDD)

- 압연 공정 중 표면을 검사하는 장치로, 빠른 속도로 진행 중인 표면에서의 결함을 빠르고 정확하게 검출하는 것이 중요함
- 현재도 SDD(Surface Defect Detector)로 표면 결함 검출 및 분류를 하고 있으나, 지속적인 성능 개선이 필요함

열간 압연(1차 압연)



Surface Defect Detector

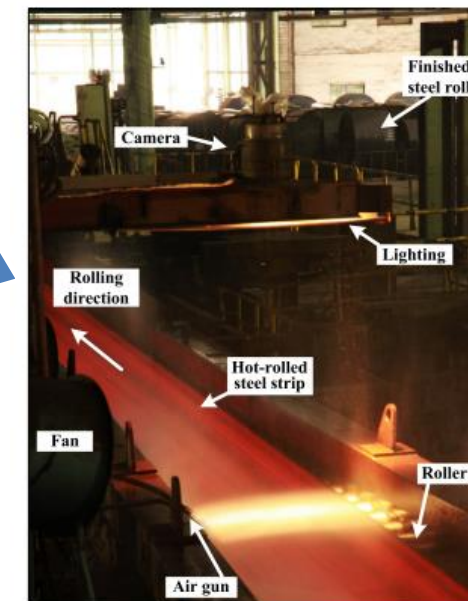
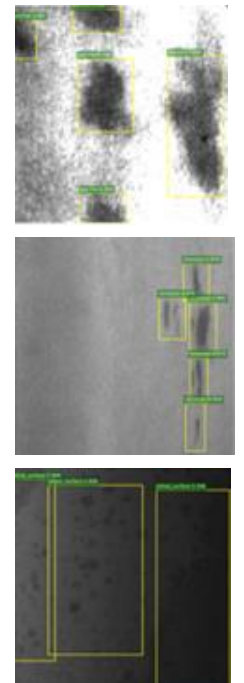


Fig. 3. AOI system installed at actual hot-rolling line.

Defect



I . Steel making process

❖ SDD 성능 개선 사례 (결함 분류율)

- 육안에 의한 결함검사시 분류 정확도 70~80%, Rule-base의 결함분류 Table 사용시 50~85% 정확도를 나타내었으나, Neural network, SVM 등의 알고리즘 개선으로 지속적인 성능 향상 진행 중임

육안 검사



70~80%

Rule-base(If-Then)

```

For J = 1 To m
  Wj = 0
  IF ( α1 ≤ X1 ≤ β1 ) THEN Wj = Wj + 1
  IF ( α2 ≤ X2 ≤ β2 ) THEN Wj = Wj + 1
  .
  .
  IF ( αn ≤ Xn ≤ βn ) THEN Wj = Wj + 1
Next

If Wk is the Maximum of Wj ( j = 1,m ) Then
Classify X to Class K
    
```

50~85%

Probabilistic Neural Network (Song, S. J., et al., 1997)

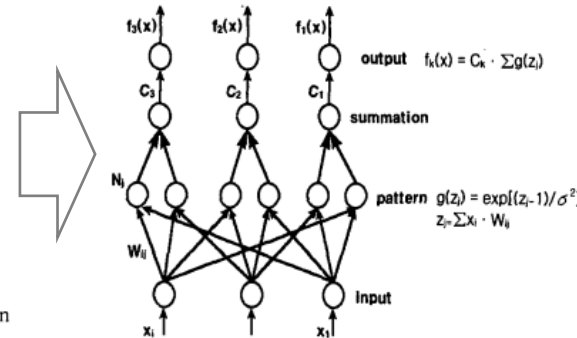


Fig. 3. Probabilistic neural network architecture

77%

SVM (Choi, Keesug, et al., 2006)

$$f(\mathbf{x}) = \text{sgn} \left(\sum_{i=1}^N \alpha_i y_i \langle \phi(x_i), \phi(\mathbf{x}) \rangle + b \right) \quad 87\sim 94\%$$

Neural Network (Moon, Chang-In, et al., 2007)

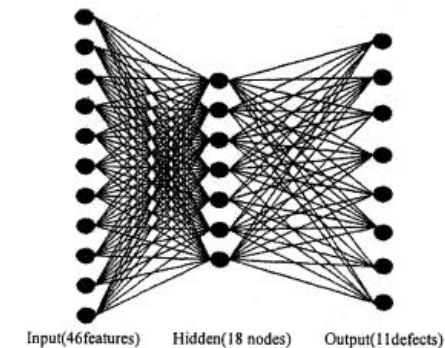


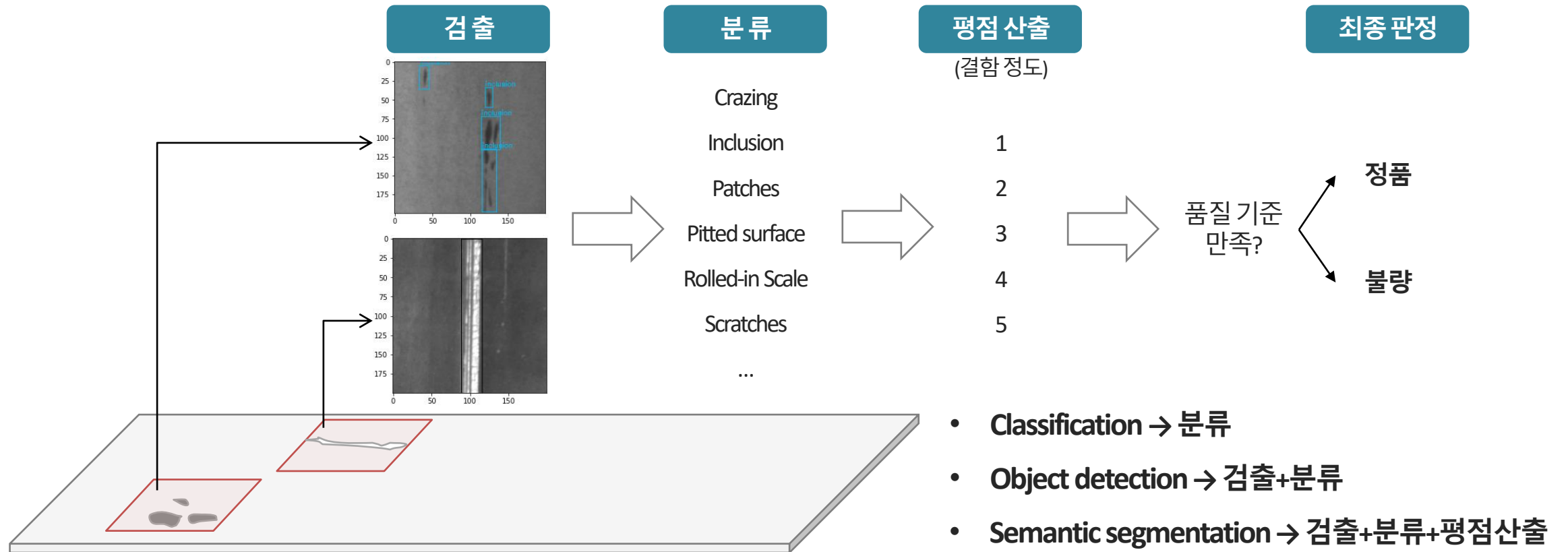
Fig. 8 Structure of neural networks

85%

II. Steel surface defects

❖ 표면 결함의 탐지 및 판정

- 제조 공정 중 발생하는 표면 결함은 **검출-분류-평점산출**을 거쳐 품질 기준 만족 여부에 따라 정품 또는 불량으로 최종 판정됨
- 결함의 종류, 개수, 크기, 위치 등에 따라 평점이 산출되고 최종 판정되기 때문에 **결함 검출만이 아닌 종류와 크기 탐지도 중요**



II. Steel surface defects

❖ 표면 결함 데이터

- 오픈 데이터셋인 Severstal, NEU에는 열연공정에서의 대표적인 표면 결함을 포함하고 있음

Dataset	Severstal	NEU
Data수	12,568 (정상/결함)	1,800 (결함)
결함 Class수	4	6
Image size	1600x256	200x200
결함정보	Pixel	Bounding box
대상공정	열연(1차 압연)	열연(1차 압연)
Task	Segmentation	Classification, Object detection
공개년도	2019	2013

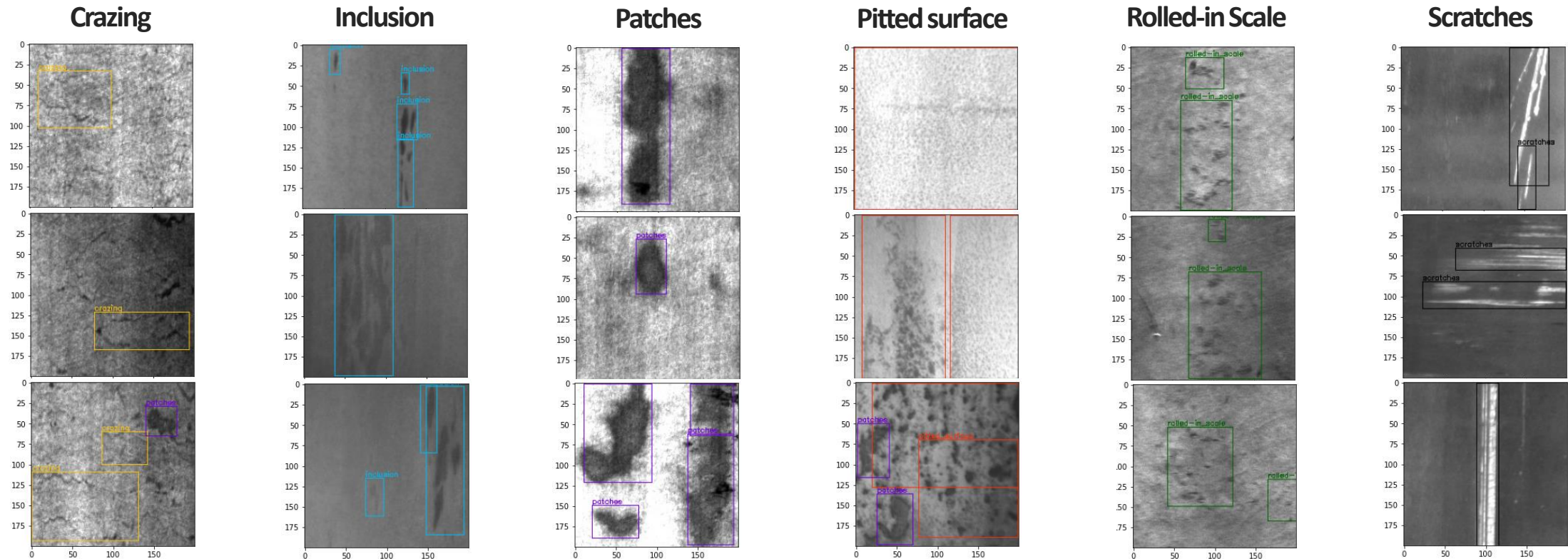
- Severstal dataset : <https://www.kaggle.com/c/severstal-steel-defect-detection/>
- NEU dataset : http://faculty.neu.edu.cn/yunhyan/NEU_surface_defect_database.html

II. Steel surface defects

❖ NEU dataset

- 열연 공정에서의 대표적인 결함 6개에 대해 Image 단위, Bounding box 단위로 Label이 제공되며, 결함 Class당 Image수는 300개

■ crazing, ■ inclusion, ■ patches, ■ pitted surface, ■ rolled-in scale, ■ scratches



II. Steel surface defects

❖ Severstal dataset

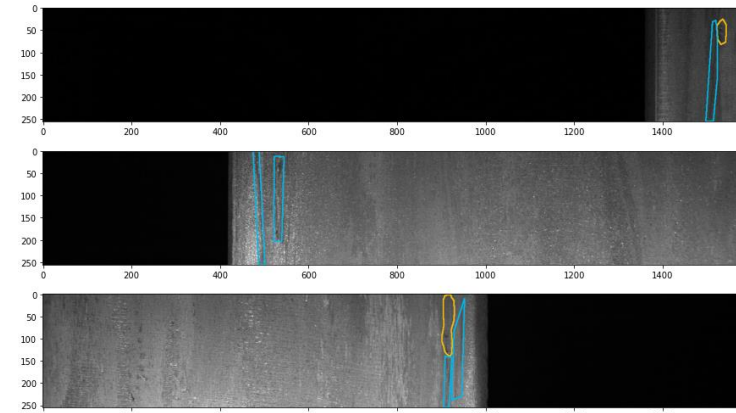
- 열연 공정에서의 결함 4개에 대해 Image 단위, Pixel 단위로 Label이 제공되며, 정상 5,902/결함 6,666개로 총 12,568개의 Image

■ pitted surface, ■ crazing, ■ scratches, ■ patches

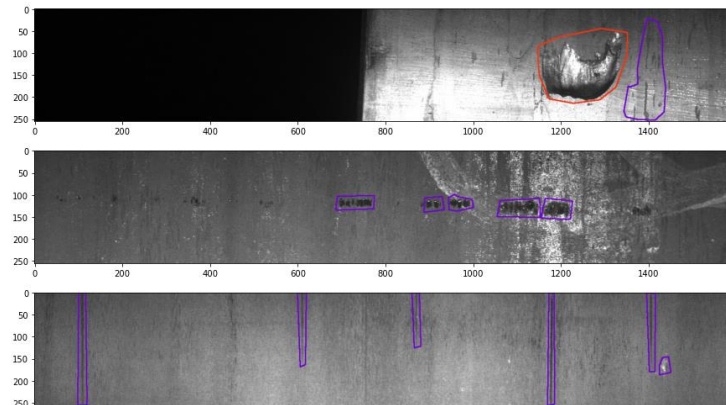
Pitted surface
(1)



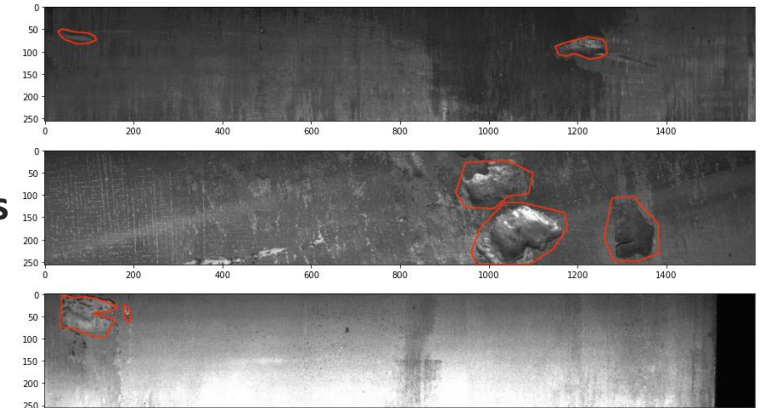
Crazing
(2)



Scratches
(3)



Patches
(4)



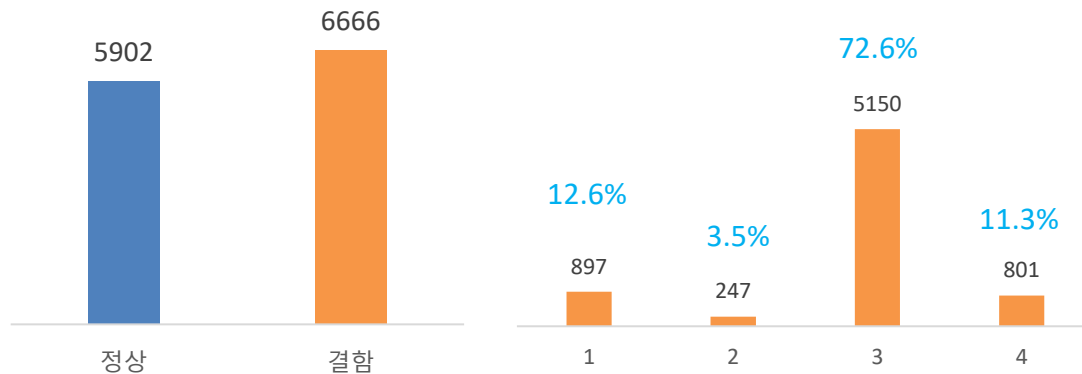
II. Steel surface defects

❖ 표면 결함 데이터의 특징

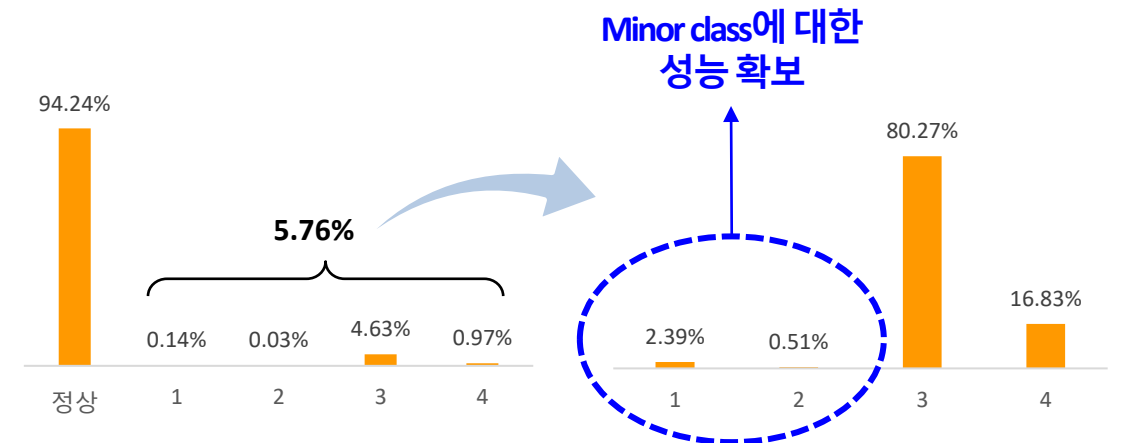
- 1) 표면 결함은 정상 영역 대비 매우 작은 부위를 차지하여 정상(Background)-결함(Defect)간 불균형이 심함
 - 2) 표면 결함은 제조 공정 중 불특정하게 나타나기 때문에 결함 Class간의 불균형이 존재함
- Severstal은 표면 결함 데이터의 특징을 잘 나타내고 있는 데이터셋

[Severstal]

< 이미지 단위 분포 >



< Pixel 단위 분포 >



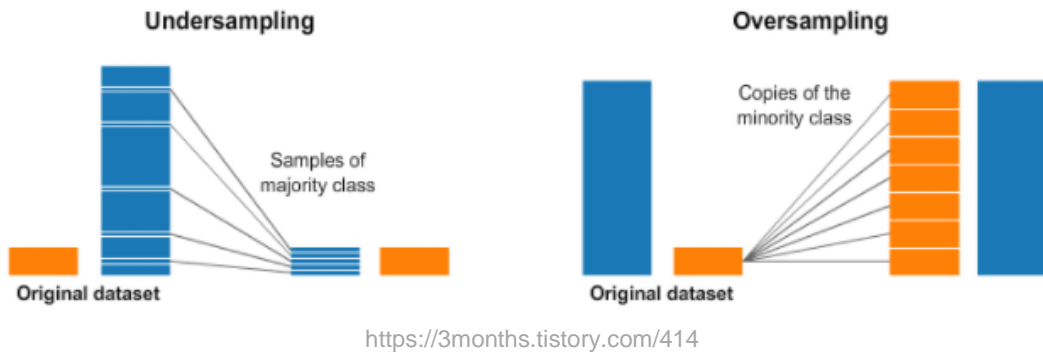
- 결함 이미지내 Pixel단위 결함은 5.76%를 차지함
- Pixel단위 결함은 Class별로 극심한 Imbalance 분포를 보임

III. Data imbalance problem

❖ 데이터 불균형을 위한 방법론

- 데이터 불균형 문제를 위한 방법론으로 Sampling 기반, Objective function 기반 외에도 다양한 방법들이 존재함
- 그 중 Objective function 기반의 방법론으로 Weighted cross entropy loss와 Focal loss에 대해 소개

Sampling based methods



Objective function based methods

Weighted cross entropy
Focal loss
...

Methods for imbalance problems

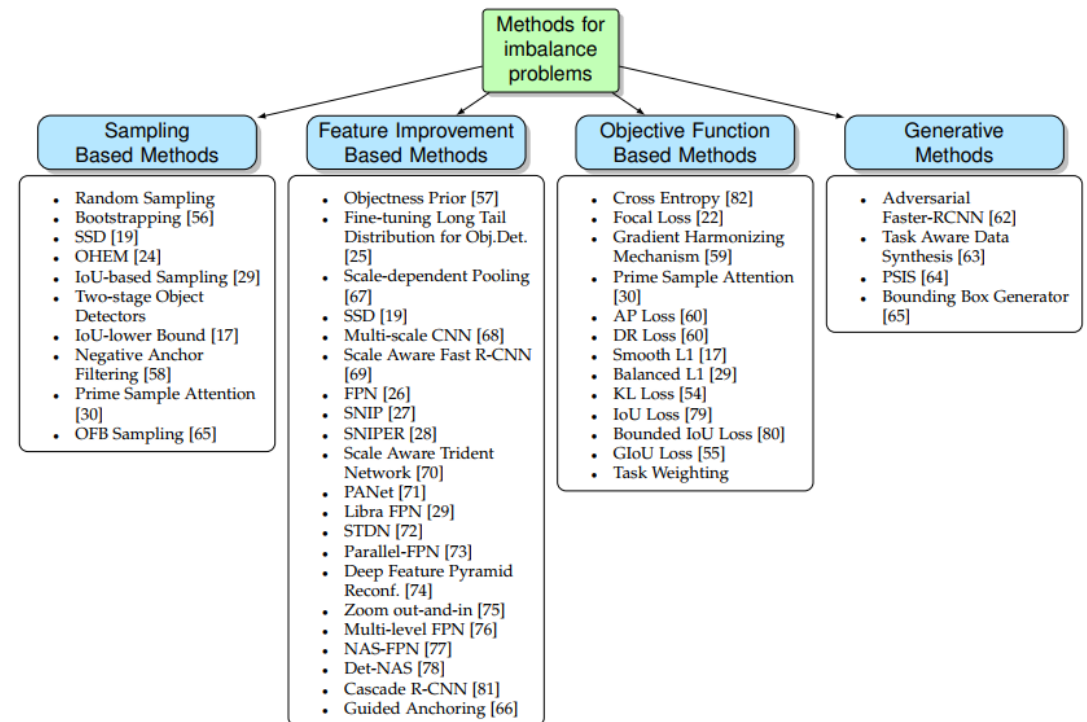
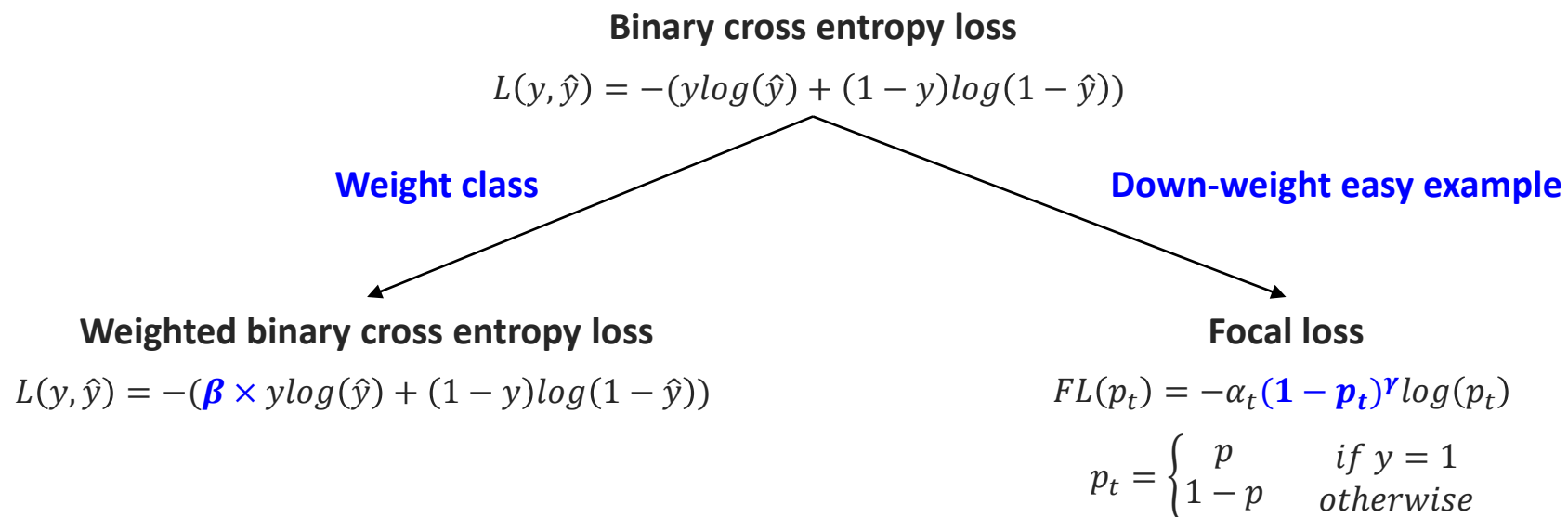


Fig. 4: Solution-approach based categorization of the methods used to address imbalance problems.

III. Data imbalance problem

❖ Objective function based methods

- Weighted BCE loss : BCE loss에 Positive weight을 추가하여 특정 Class에 가중치를 부여하는 Loss
- Focal loss : BCE loss에 Modulating factor를 추가하여 Easy example에 대한 가중치를 줄이고 Hard example에 집중할 수 있는 Loss



III. Data imbalance problem

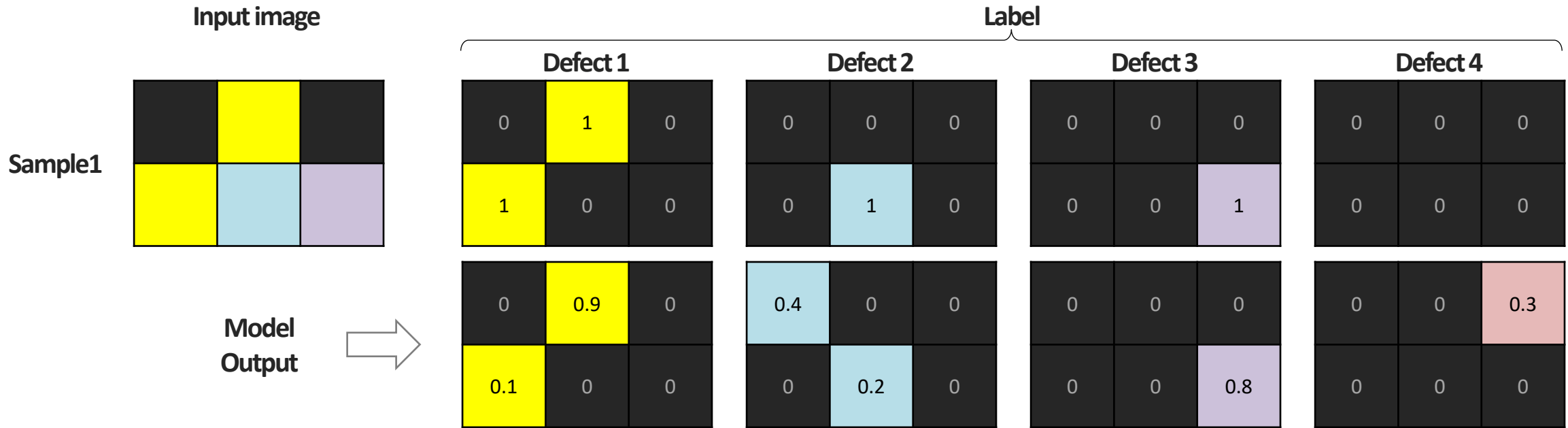
❖ Objective function based methods – Example

- Semantic segmentation을 통한 표면 결함 탐지에 대한 예시로, 데이터 불균형이 있는 4개의 결함 Class를 가정(Minor class : 2)
- 6개 Pixel을 가진 Image로 단순화하여 BCE loss, Weighted BCE loss, Focal loss가 계산되는 과정을 살펴보고자 함



III. Data imbalance problem

❖ Example : Binary cross entropy loss



Binary cross entropy loss

$$L(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

$$L(1, 0.9) = -(1 \log(0.9) + (1 - 1) \log(1 - 0.9)) = 0.1054$$

$$L(1, 0.1) = -(1 \log(0.1) + (1 - 1) \log(1 - 0.1)) = 2.3026$$

$$L(0, 0.4) = -(0 \log(0.4) + (1 - 0) \log(1 - 0.4)) = 0.5108$$

$$L(1, 0.2) = -(1 \log(0.2) + (1 - 1) \log(1 - 0.2)) = 1.6094$$

$$L(1, 0.8) = -(1 \log(0.8) + (1 - 1) \log(1 - 0.8)) = 0.2231$$

$$L(0, 0.3) = -(0 \log(0.3) + (1 - 0) \log(1 - 0.3)) = 0.3567$$

$$L_{BCE} = (0.1054 + 2.3026 + 0.5108 + 1.6094 + 0.2231 + 0.3567) / 24 = \mathbf{0.2128}$$

Defect 1

Defect 2

Defect 3

Defect 4

III. Data imbalance problem

❖ Example : Weighted binary cross entropy loss (Positive weight = [1,3,1,1])



Weighted binary cross entropy loss

$$L(y, \hat{y}) = -(\beta \times y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

Note: β value can be used to tune false negatives and false positives.
 E.g; If you want to reduce the number of false negatives then set $\beta > 1$,
 similarly to decrease the number of false positives, set $\beta < 1$. [1]

$$L(1,0.9) = -(1 \log(0.9) + (1 - 1) \log(1 - 0.9)) = 0.1054$$

$$L(1,0.1) = -(1 \log(0.1) + (1 - 1) \log(1 - 0.1)) = 2.3026$$

$$L(0,0.4) = -(3 \times 0 \log(0.4) + (1 - 0) \log(1 - 0.4)) = 0.5108$$

$$L(1,0.2) = -(3 \times 1 \log(0.2) + (1 - 1) \log(1 - 0.2)) = 4.8283$$

$$L(1,0.8) = -(1 \log(0.8) + (1 - 1) \log(1 - 0.8)) = 0.2231$$

$$L(0,0.3) = -(0 \log(0.3) + (1 - 0) \log(1 - 0.3)) = 0.3567$$

$$L_{WBCE} = (0.1054 + 2.3026 + 0.5108 + 4.8283 + 0.2231 + 0.3567) / 24 = 0.3470$$

Defect 1

Defect 2

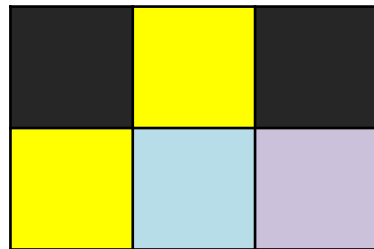
Defect 3

Defect 4

III. Data imbalance problem

❖ BCE vs. Weighted BCE

Input image



Sample1

Label

	Defect 1	Defect 2	Defect 3	Defect 4
Label	0 1 0	0 0 0	0 0 0	0 0 0
Model Output	0 0.9 0	0.4 0 0	0 0 0	0 0 0.3

Labels: Defect 1 (yellow), Defect 2 (light blue), Defect 3 (purple), Defect 4 (pink)

Model Output



BCE loss
0.2128

$w=[1,3,1,1]$

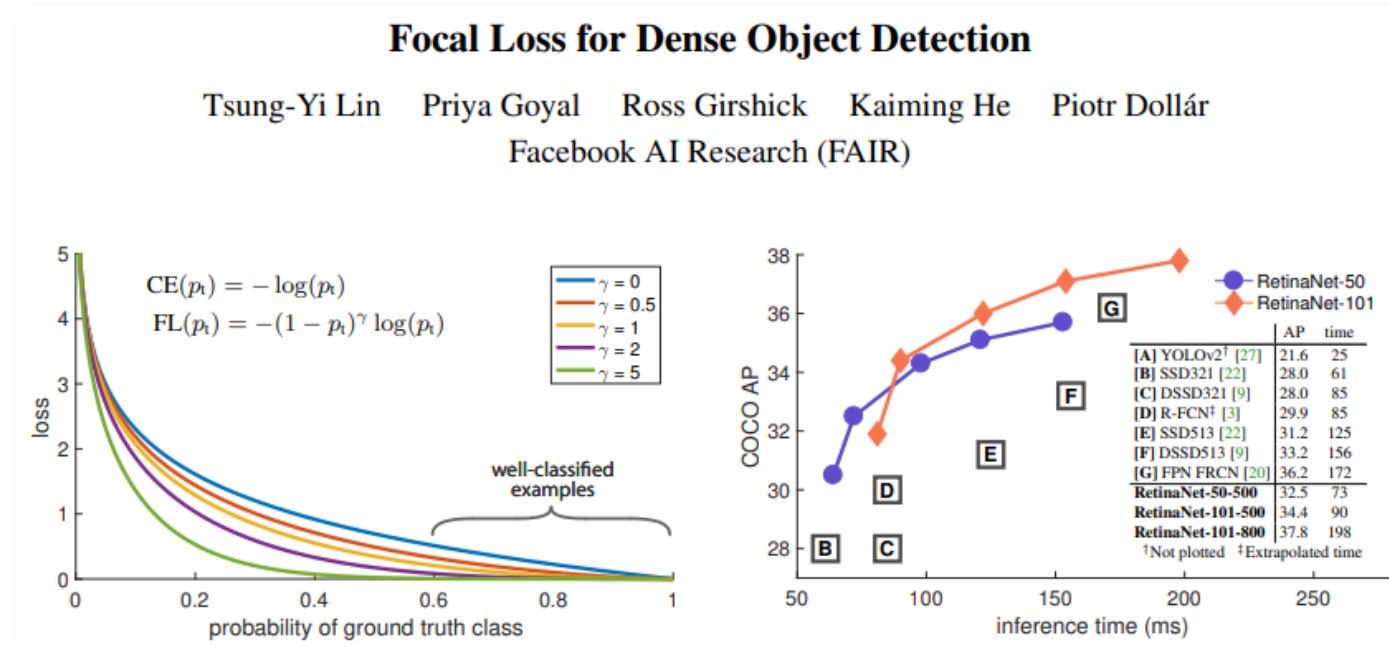


Weighted BCE loss
0.3470

III. Data imbalance problem

❖ “Focal loss for dense object detection”

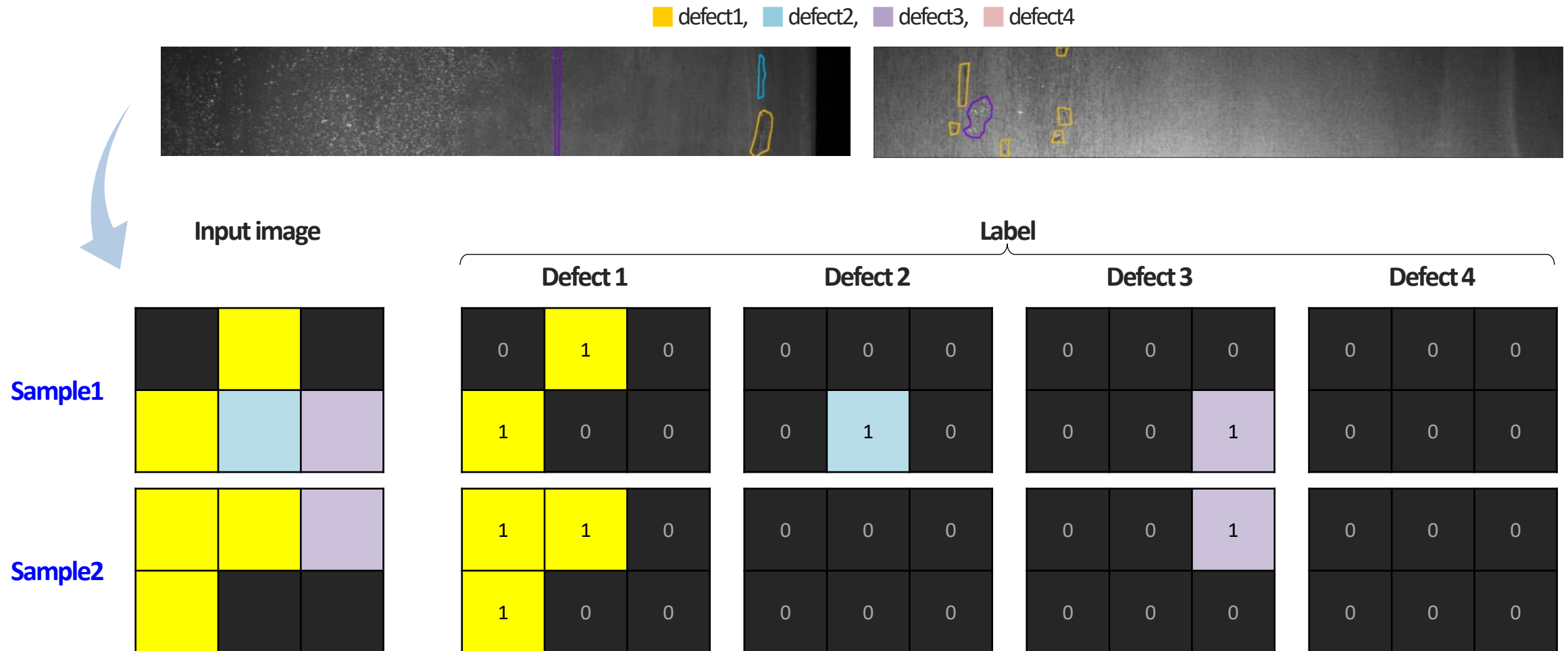
- 1-stage object detector의 성능 향상을 위해 Foreground와 Background간의 Class imbalance를 해결하는 Focal loss를 제안
- 예측이 잘 되는 Easy example의 가중치를 줄이고 예측이 잘 안되는 Hard example에 집중할 수 있도록 기존 Cross entropy loss에 Modulating factor $(1 - p_t)^\gamma$ 를 적용



- Lin, Tsung-Yi, et al. (Facebook AI Research)
- 2017 International conference on computer vision (ICCV)

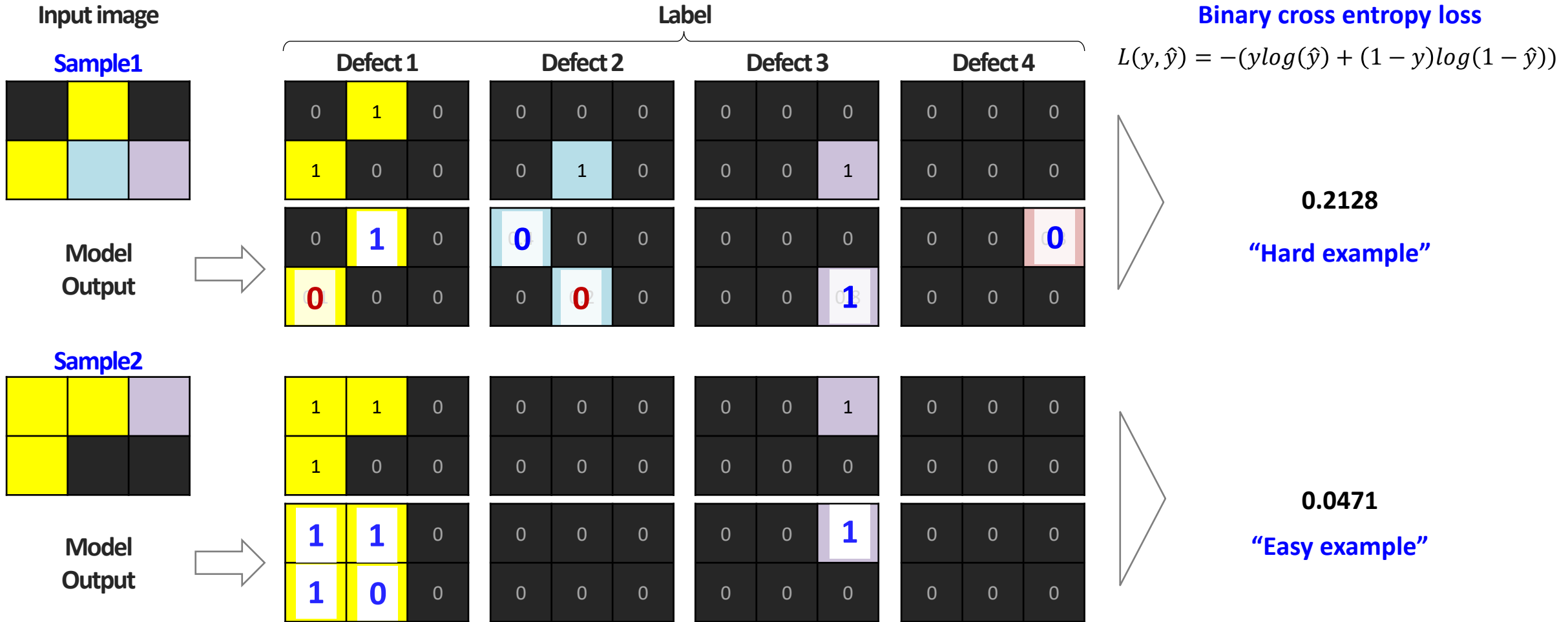
III. Data imbalance problem

❖ Focal loss



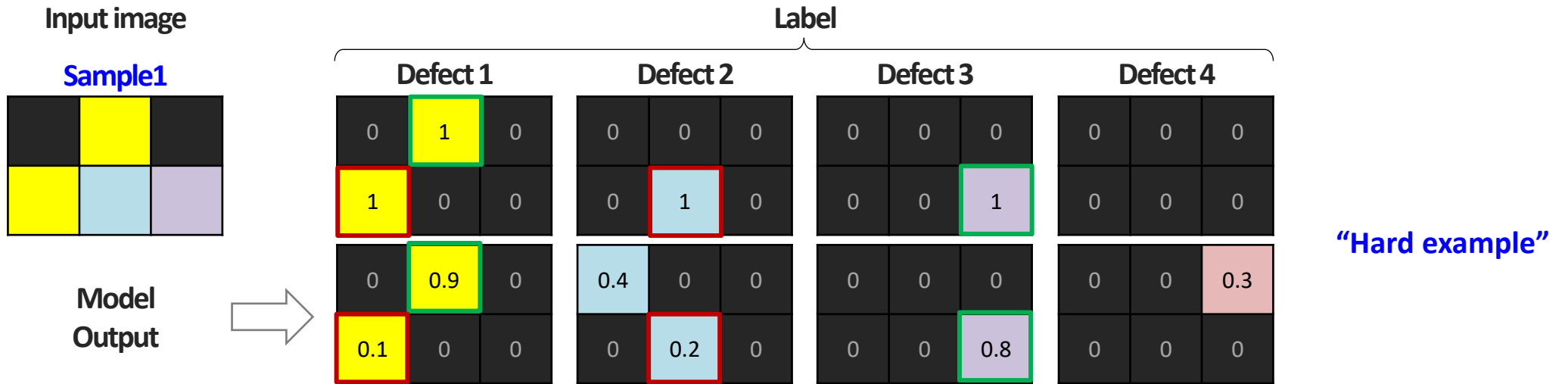
III. Data imbalance problem

❖ Easy example and Hard example – BCE loss



III. Data imbalance problem

❖ Example : Focal loss ($\alpha=0.75, \gamma=2$)



Focal loss

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t)$$

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases}$$

$$L_{\text{Focal}} = -(\alpha(1 - \hat{y})^\gamma y \log(\hat{y}) + (1 - \alpha)(1 - (1 - \hat{y}))^\gamma (1 - y) \log(1 - \hat{y}))$$

$$L(1,0.9) = -(0.75(1 - 0.9)^2 1 \log(0.9) + 0) = 0.0008 \quad \downarrow$$

$$L(1,0.1) = -(0.75(1 - 0.1)^2 1 \log(0.1) + 0) = 1.3988 \quad \uparrow$$

$$L(0,0.4) = -(0 + (1 - 0.75)(0.4)^2 (1 - 0) \log(1 - 0.4)) = 0.0204$$

$$L(1,0.2) = -(0.75(1 - 0.2)^2 1 \log(0.2) + 0) = 0.7725 \quad \uparrow$$

$$L(1,0.8) = -(0.75(1 - 0.8)^2 1 \log(0.8) + 0) = 0.0067 \quad \downarrow$$

$$L(0,0.3) = -(0 + (1 - 0.75)(0.3)^2 (1 - 0) \log(1 - 0.3)) = 0.0080$$

$$L_{\text{Focal}} = (0.0008 + 1.3988 + 0.0204 + 0.7725 + 0.0067 + 0.0080) / 24 = \mathbf{0.0920}$$

BCE loss

0.1054

2.3026

0.5108

1.6094

0.2231

0.3567

Defect 1

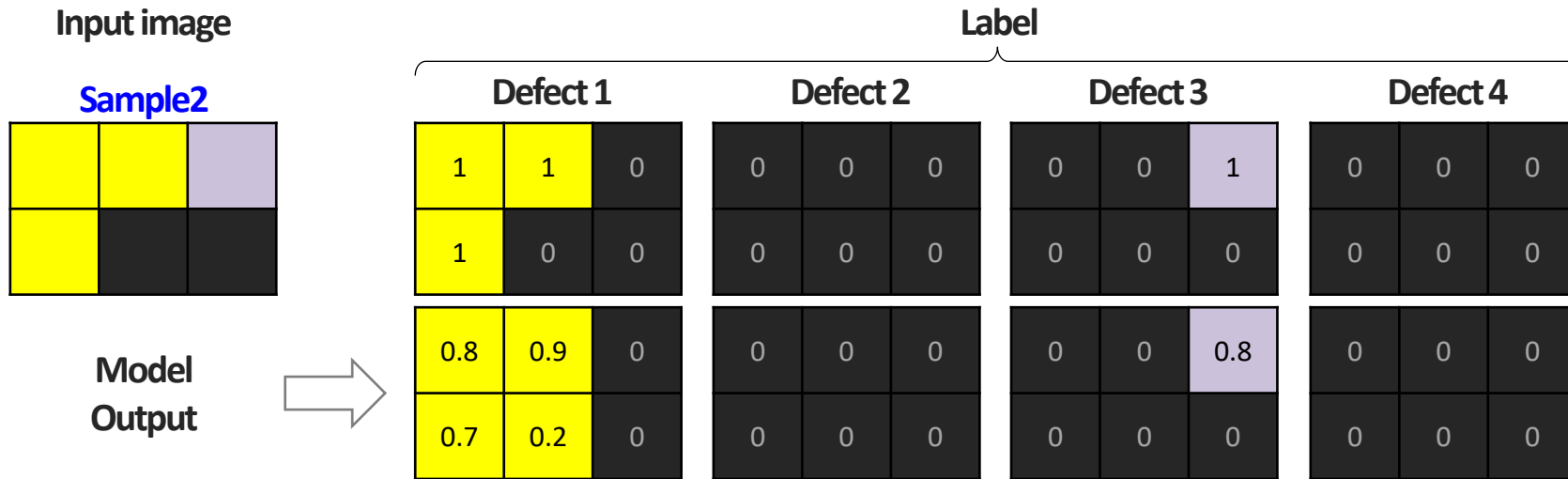
Defect 2

Defect 3

Defect 4

III. Data imbalance problem

❖ Example : Focal loss ($\alpha=0.75, \gamma=2$)



“Easy example”

Focal loss

$$L_{\text{Focal}} = -(\alpha(1 - \hat{y})^\gamma y \log(\hat{y}) + (1 - \alpha)(1 - (1 - \hat{y}))^\gamma (1 - y) \log(1 - \hat{y}))$$

$$FL(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t)$$

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise} \end{cases}$$

$$L(1,0.8) = -(0.75(1 - 0.8)^2 1 \log(0.8) + 0) = 0.0067$$

$$L(1,0.9) = -(0.75(1 - 0.9)^2 1 \log(0.9) + 0) = 0.0008$$

$$L(1,0.7) = -(0.75(1 - 0.7)^2 1 \log(0.7) + 0) = 0.0241$$

$$L(0,0.2) = -(0 + (1 - 0.75)(0.2)^2 (1 - 0) \log(1 - 0.2)) = 0.0022$$

$$L(1,0.8) = -(0.75(1 - 0.8)^2 1 \log(0.8) + 0) = 0.0067$$

Defect 1

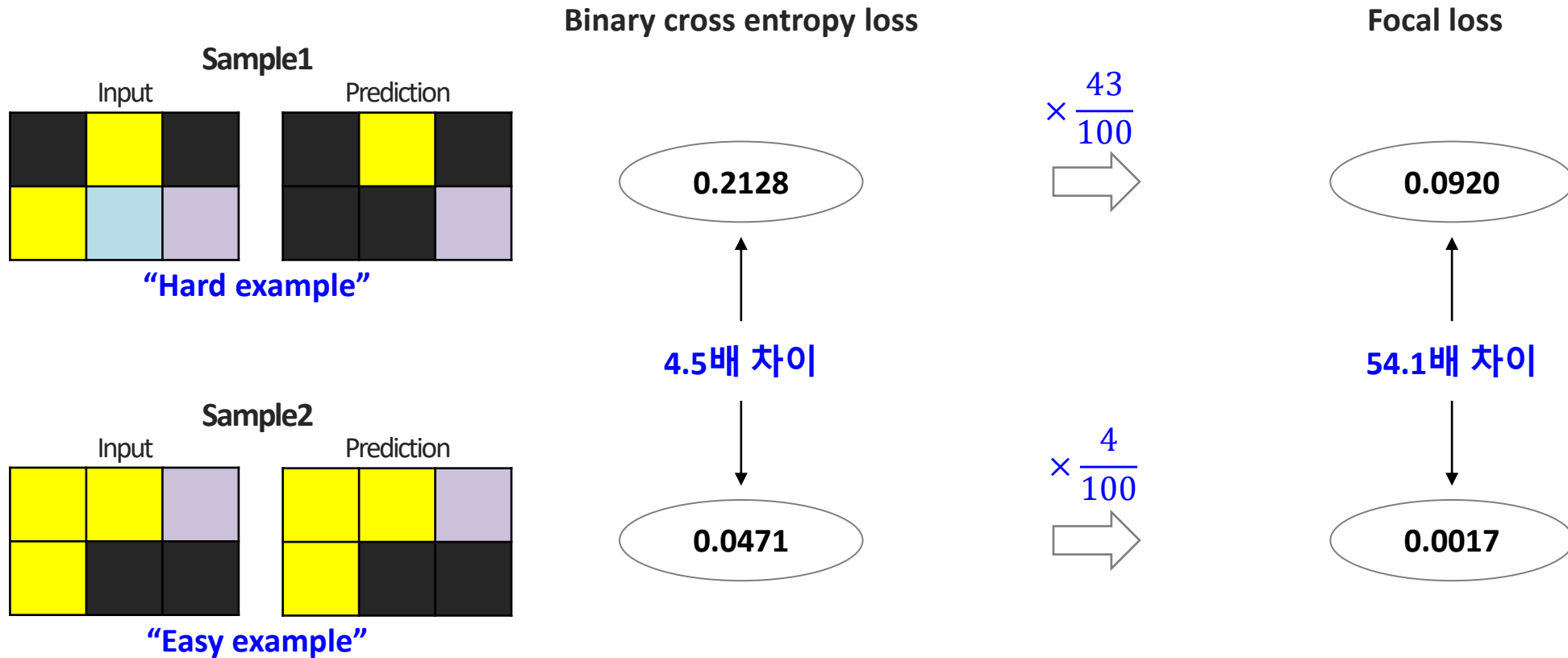
Defect 3

$$L_{\text{Focal}} = (0.0067 + 0.0008 + 0.0241 + 0.0022 + 0.0067) / 24 = 0.0017$$

Ⅲ. Data imbalance problem

❖ BCE vs. Focal

- Focal loss는 데이터 불균형에 기인한 Easy/Hard example에 대해 Hard example 대비 Easy example의 loss값을 훨씬 많이 감소시켜 Hard example의 loss가 모델 학습에 더 많이 반영될 수 있도록 함



Ⅲ. Data imbalance problem

❖ BCE vs. Weighted BCE vs. Focal

- Weighted BCE loss는 특정 Class에 대한 가중치를 부여하여 해당 Class가 포함된 Sample의 Loss를 증대
- Focal loss는 Easy example에 대한 Loss를 상대적으로 더 많이 감소시켜 Hard example에 집중함으로써 불균형 문제를 해소

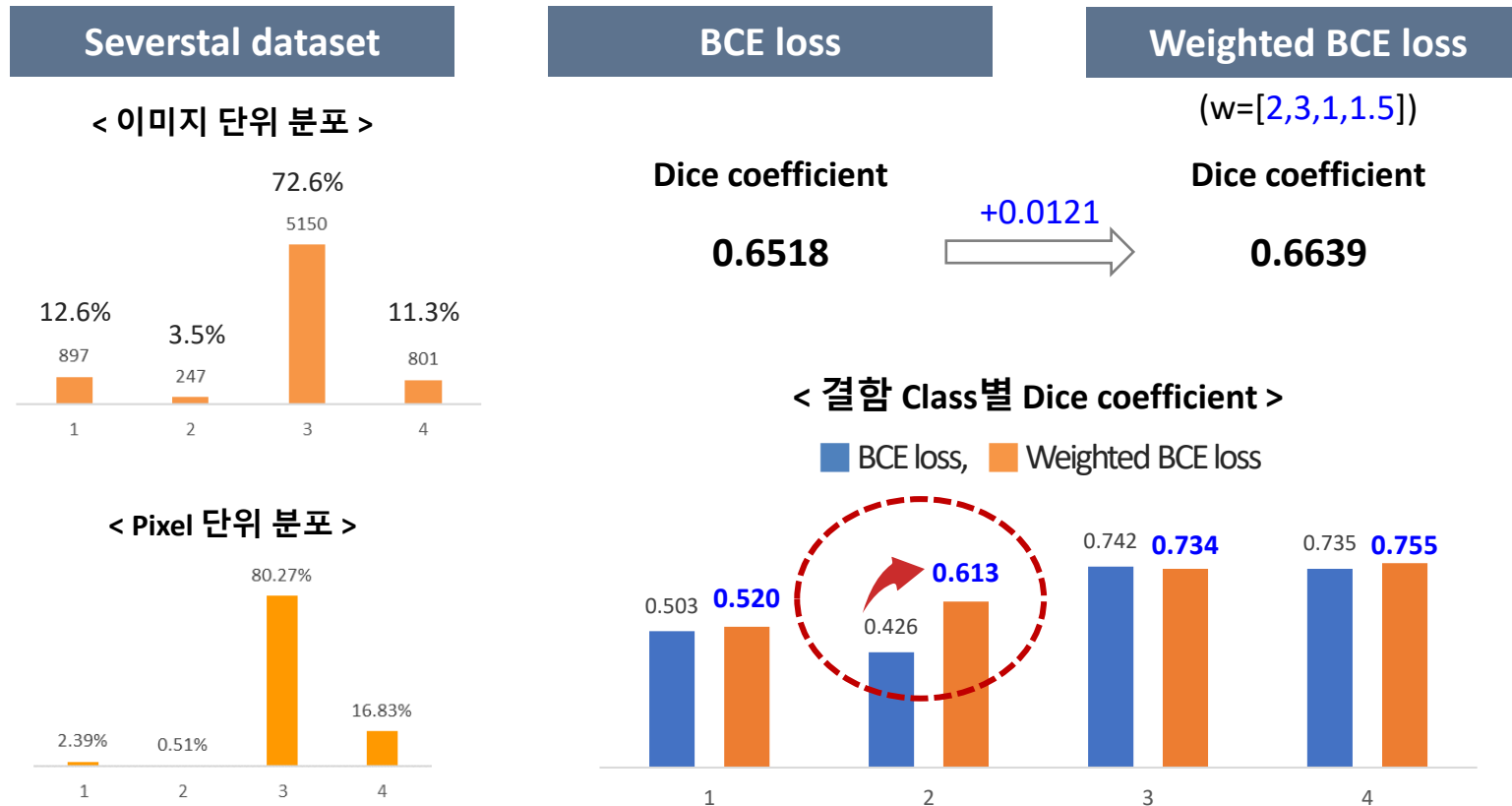
	BCE	Weighted BCE (positive weight=[1,3,1,1])	Focal ($\gamma=2, \alpha=0.75$)
<p>Sample1</p> <p>Input Prediction</p>  <p>“Hard example”</p>	0.2128	0.3470	0.0920
<p>Sample2</p> <p>Input Prediction</p>  <p>“Easy example”</p>	0.0471	0.0471	0.0017

III. Data imbalance problem

❖ BCE vs. Weighted BCE – Severstal dataset

- Severstal dataset에 대한 Segmentation task 성능 비교 결과, Weight를 부여한 결함 Class의 Dice 성능이 향상됨을 확인

* 비교 모델 : Loss function 외에는 모두 동일한 조건의 모델을 사용함



※ Severstal Kaggle competition 1st place

1st Place Solution
Posted in severstal-steel-defect-detection 2 years ago

R Guo
1st place

Congratulations to all winners in this competition competition for my teammates @zhubonbo and

Segmentation

We have to admit that we used models from @lightforev private LB to our current score.

Train data: 256x512 crop images

Augmentations: Hflip, Vflip, RandomBrightnessContrast

Batchsize: 12 or 24 (both accumulate gradients for 24 s)

Optimizer: Rectified Adam

Models: Unet (efficientnet-b3), FPN (efficientnet-b3) fr

Loss:

BCE (with pos_weight = (2.0,2.0,1.0,1.5))

0.75BCE+0.25DICE (with pos_weight = (2.0,2.0,1.0,1.5))

Model Ensemble:

<https://www.kaggle.com/c/severstal-steel-defect-detection/discussion/114254>

IV. Deep learning for steel surface defects detection

❖ “Automatic Detection and Classification of Steel Surface Defect Using Deep Convolutional Neural Networks”

- Wang, Shuai, et al. (2021), *Metals*
- Severstal 데이터셋에 대해 Improved faster R-CNN 등을 적용하여 효율적이면서 정확도가 높은 Object detection 모델을 제안
- Loss function으로 Weighted BCE loss를 사용하고, Minor class에 대해 Data augmentation을 적용하여 Sample수 확보



metals



Article

Automatic Detection and Classification of Steel Surface Defect Using Deep Convolutional Neural Networks

Shuai Wang ^{1,2,*}, Xiaojun Xia ^{1,2}, Lanqing Ye ^{1,2} and Binbin Yang ^{1,2}

¹ School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China; xiaojun@sict.ac.cn (X.X.); yelanqing18@mails.ucas.ac.cn (L.Y.); yangbinbin181@mails.ucas.ac.cn (B.Y.)

² Shenyang Institute of Computing Technology, Chinese Academy of Sciences, Shenyang 110168, China

* Correspondence: wangshuai14@mails.ucas.ac.cn; Tel.: +86-1980-032-1230

Abstract: Automatic detection of steel surface defects is very important for product quality control in the steel industry. However, the traditional method cannot be well applied in the production line, because of its low accuracy and slow running speed. The current, popular algorithm (based on deep

IV. Deep learning for steel surface defects detection

❖ Model Framework 및 Dataset

- Framework : 2-stage로 Classification에서 결함 유/무를 먼저 분류하고, 결함 대상 Sample에 대해 Object detection 모델 적용
- Dataset : 12,568개의 1600x256 Image를 400x256 Image 4개로 분할하여 50,272개의 데이터를 사용하였고, 원래의 데이터셋은 Pixel 단위 결함 Label이 제공되나, 별도로 Annotation을 적용하여 Bounding box 형태로 Label 변경

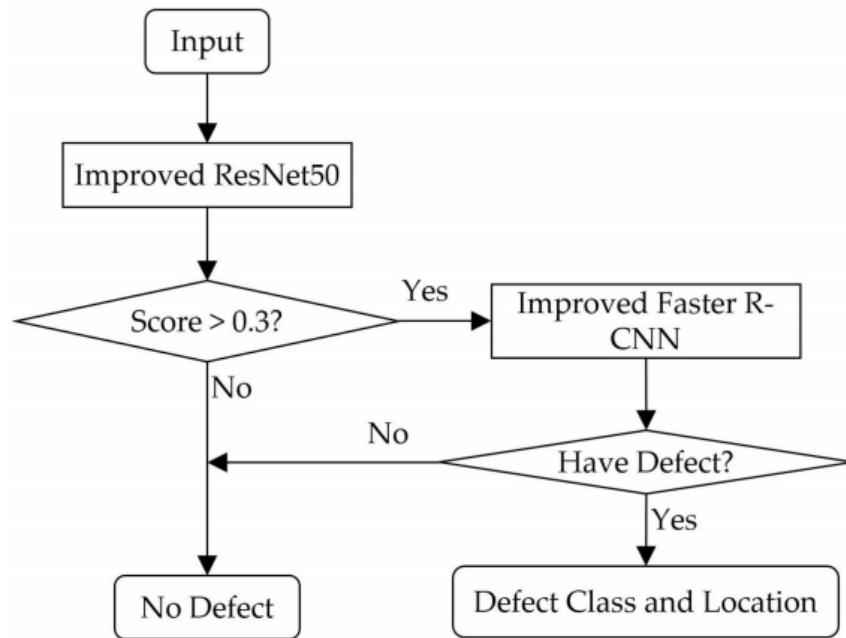


Figure 1. Algorithm Framework.

Table 1. Classification model dataset.

Class	Train	Valid	Total
With Defects	10,553	2639	13,192
Without Defects	29,664	7416	37,080
Total	40,217	10,055	50,272

Table 2. Object detection model dataset.

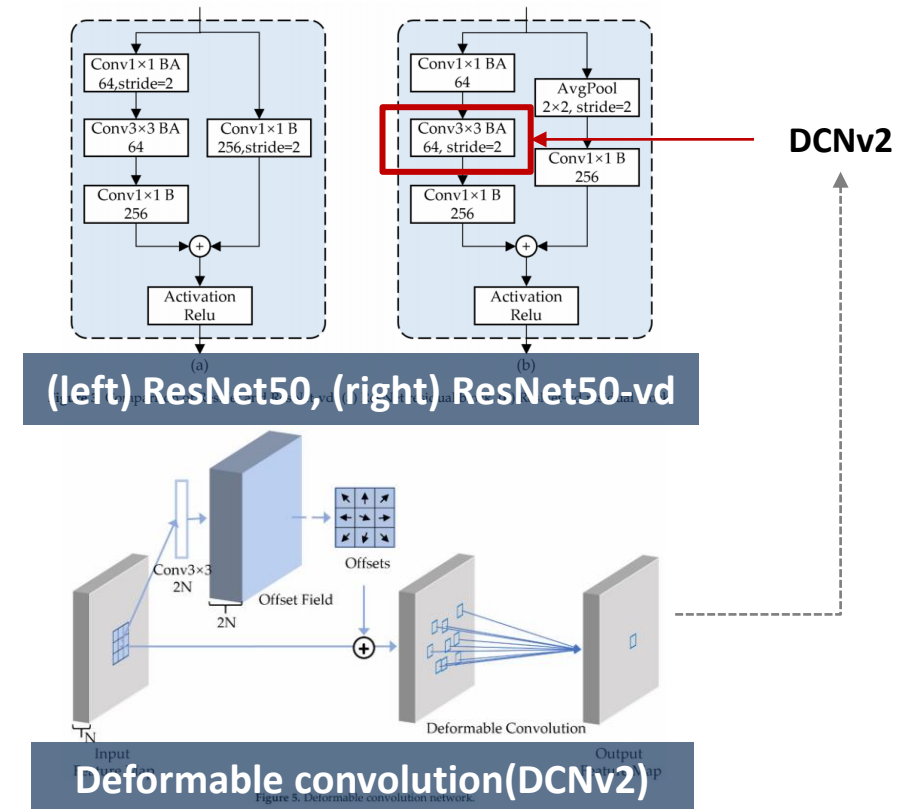
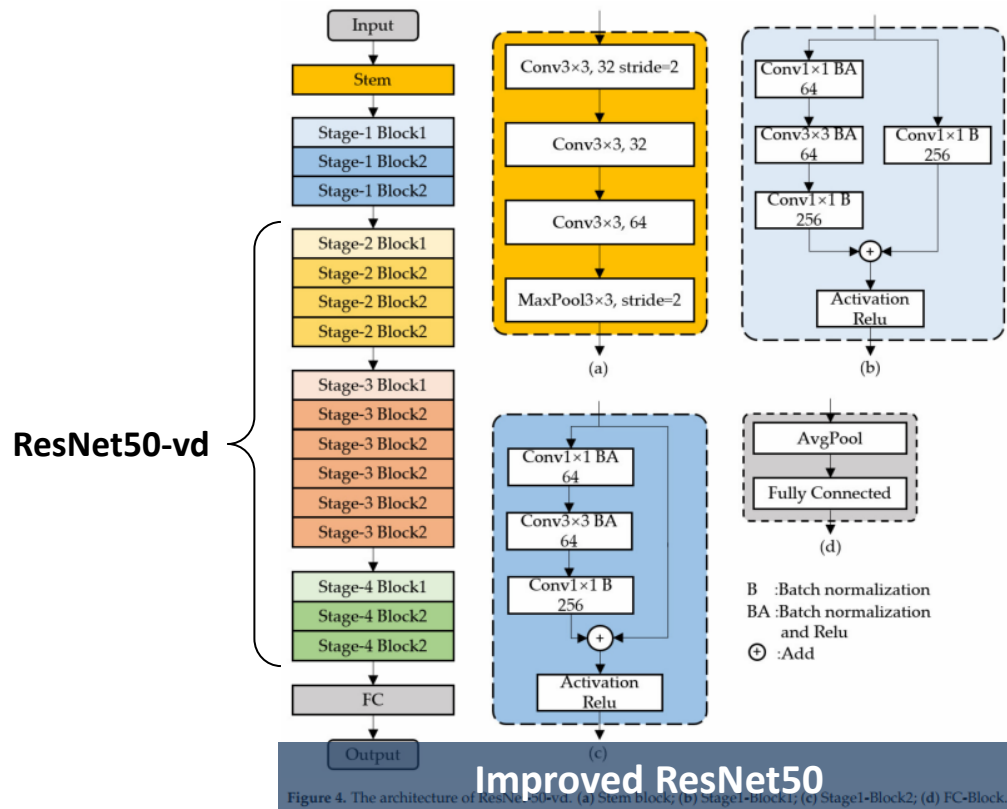
Class	Train	Valid	Total
Pitted surface	1044	262	1306
Crazing	171	43	214
Scratches	7984	1996	9980
Patches	1100	276	1376
Multi Class Defect	252	64	316
Without Defect	29,664	7416	37,080
Total	40,215	10,057	50,272



IV. Deep learning for steel surface defects detection

❖ Classification Model - Improved ResNet50

- 기존 ResNet50 block을 변형하여 향상된 성능을 보인 ResNet50-vd 기반
- Object의 기하학적 변형에 적응할 수 있는 Module인 Deformable convolution network(DCNv2)를 적용



IV. Deep learning for steel surface defects detection

❖ Object Detection Model - Improved Faster R-CNN

- Classification 모델 구조인 ResNet50-vd+DCNv2를 기반으로, Multi-scale defect detection을 위해 FPN(Feature Pyramid Network)과 SPP(Spatial Pyramid Pooling)가 적용된 Improved Faster R-CNN을 제안

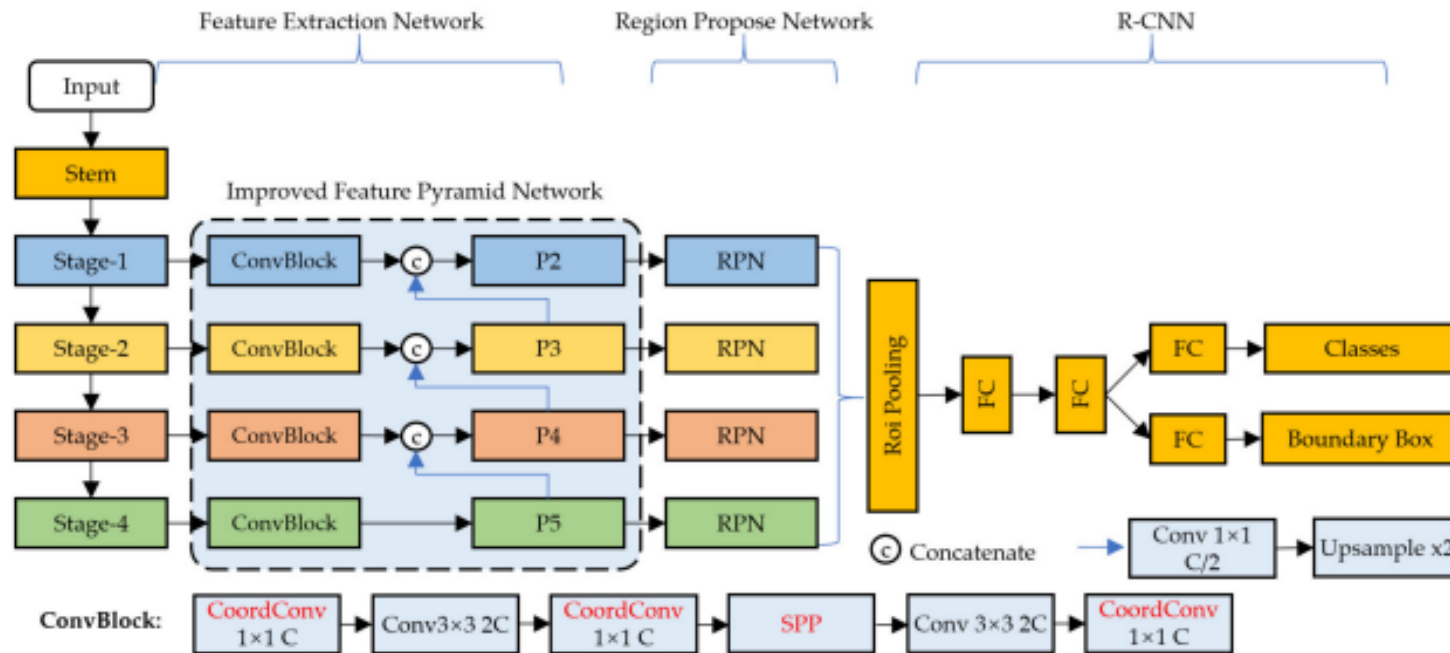


Figure 6. The architecture of enhanced faster R-CNN.

Improved Faster R-CNN

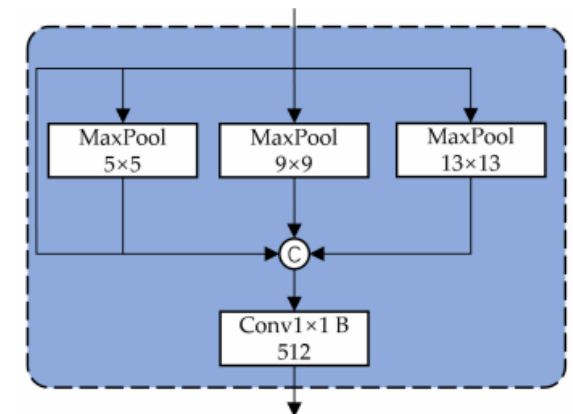


Figure 7. Spatial pyramid pooling (SPP).

Spatial Pyramid Pooling

IV. Deep learning for steel surface defects detection

❖ Loss function and Data augmentation

- 데이터 불균형 문제를 위해 Weighted BCE loss를 사용(Positive weight : (Detection) [1.5, 2.0, 1.0, 1.5], (Classification) [1.0, 2.0]) 하였고, Minor class에 대한 데이터수를 Augmentation을 통해 증대시킴(Crazing class : 214→800)
- Classification task를 위한 Data augmentation으로 Improved cutout 방법을 제안
- * Improved cutout : Original cutout과 달리, 결함 Label(Bounding box)을 이용하여 Random하게 선택된 Label box를 Cutout 시킴

Weighted binary cross entropy loss

$$L(y, \hat{y}) = -(\beta \times y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}))$$

↑
[1.5, 2.0, 1.0, 1.5] / [1.0, 2.0]
(segmentation / classification)

800으로 증대

Table 2. Object detection model dataset.

Class	Train	Valid	Total
Pitted surface	1044	262	1306
Crazing	171	43	214
Scratches	7984	1996	9980
Patches	1100	276	1376
Multi Class Defect	252	64	316
Without Defect	29,664	7416	37,080
Total	40,215	10,057	50,272

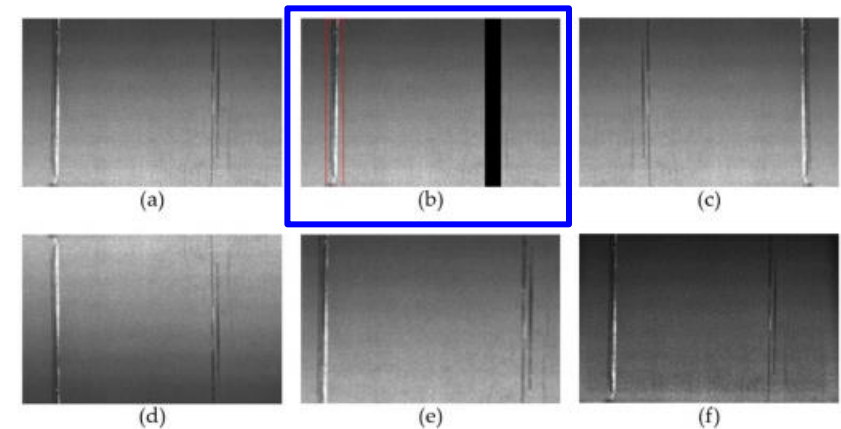


Figure 9. Data augmentation. (a) Original image; (b) improved cutout; (c) horizontal flip; (d) vertical flip; (e) random crop; (f) random contrast and brightness transformation.

Improved Cutout

IV. Deep learning for steel surface defects detection

❖ Results and Discussion

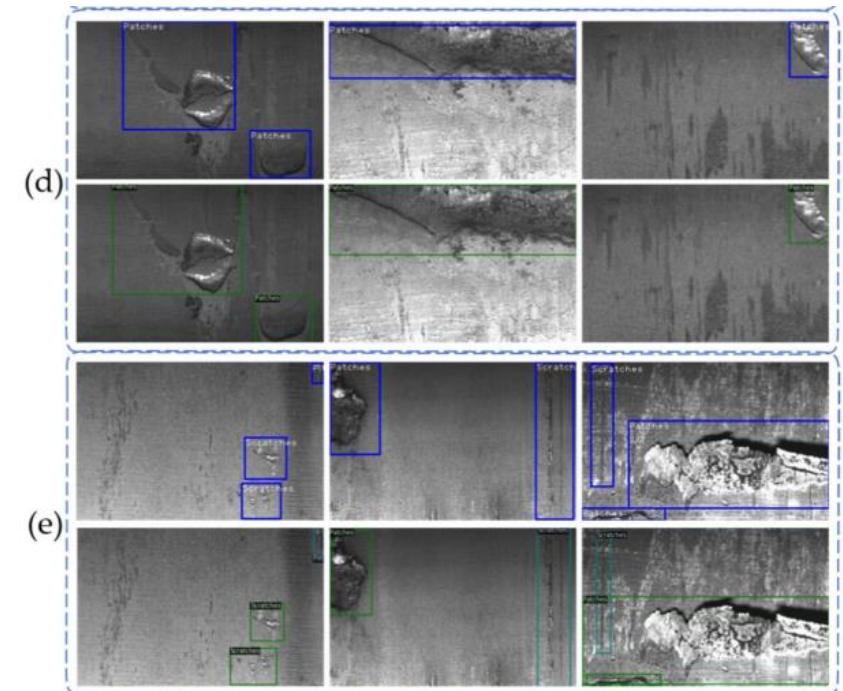
- Classification(결함 유/무 분류) : ResNet-vd+DCNv2+ImprovedCutout 모델로 성능 향상(Accuracy 0.9752)
- Object detection : 최종 모델 기준 mAP 0.876을 달성하였으나, Minor class인 Pitted surface와 Crazing의 성능은 상대적으로 낮음

Table 5. Comparison of the classification models.

Model	Original Image		Horizontal Flip		Vertical Flip		TTA		Running Time
	F1	Acc	F1	Acc	F1	Acc	F1	Acc	
Fadli et al. [37]	-	0.94	-	-	-	-	-	-	
ResNet	0.968	0.969	0.9627	0.9628	0.9626	0.9627	0.9673	0.9675	2.40 ms
Konovalenko et al. [38]	-	0.9691	-	-	-	-	-	-	
ResNet_vd	0.9707	0.9708	0.9694	0.9695	0.9689	0.9690	0.9710	0.9711	2.44 ms
ResNet_vd_dcnV2	0.9732	0.9732	0.9726	0.9726	0.9730	0.9731	0.9739	0.9739	2.9 ms
ResNet_vd_dcnV2_ImprovedCutout	0.9747	0.9747	0.9744	0.9744	0.9745	0.9745	0.9752	0.9752	2.9 ms

Table 6. The AP of each type of defect and the total mAP of the model.

Model	Type of Defect				mAP
	Pitted Surface	Crazing	Scratches	Patches	
YOLOv3	0.700	0.643	0.749	0.724	0.704
Faster-RCNN	0.801	0.767	0.874	0.865	0.827
Faster-RCNN + DCN	0.806	0.782	0.881	0.876	0.836
Faster-RCNN + DCN + FPN&SPP	0.840	0.818	0.902	0.895	0.864
Faster-RCNN + DCN + FPN&SPP + Coordconv	0.845	0.829	0.913	0.916	0.876



V. Conclusion

철강 제조과정

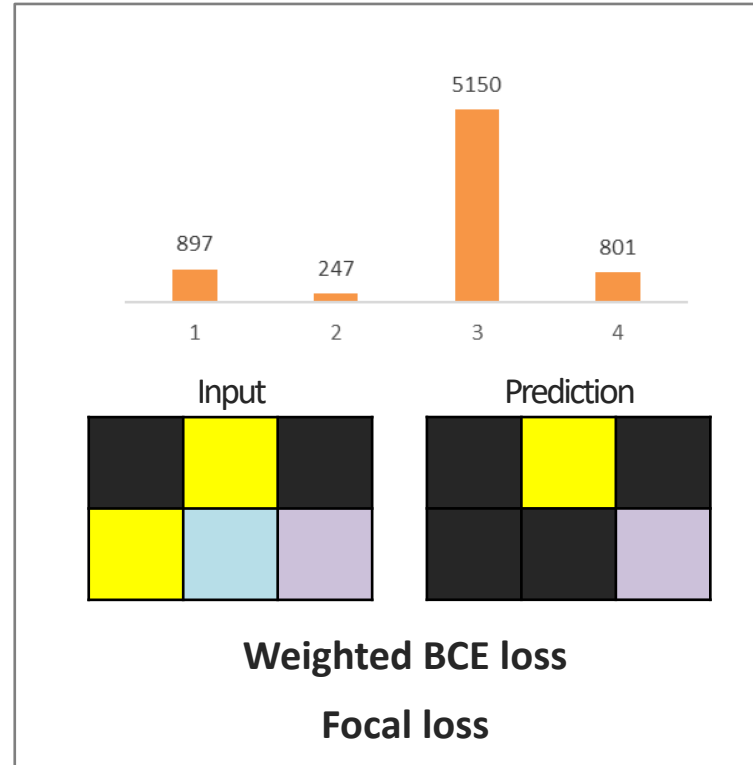
Iron Man Man of Steel (Superman)

Iron Man 3 V.s Man Of Steel ... Who wins?

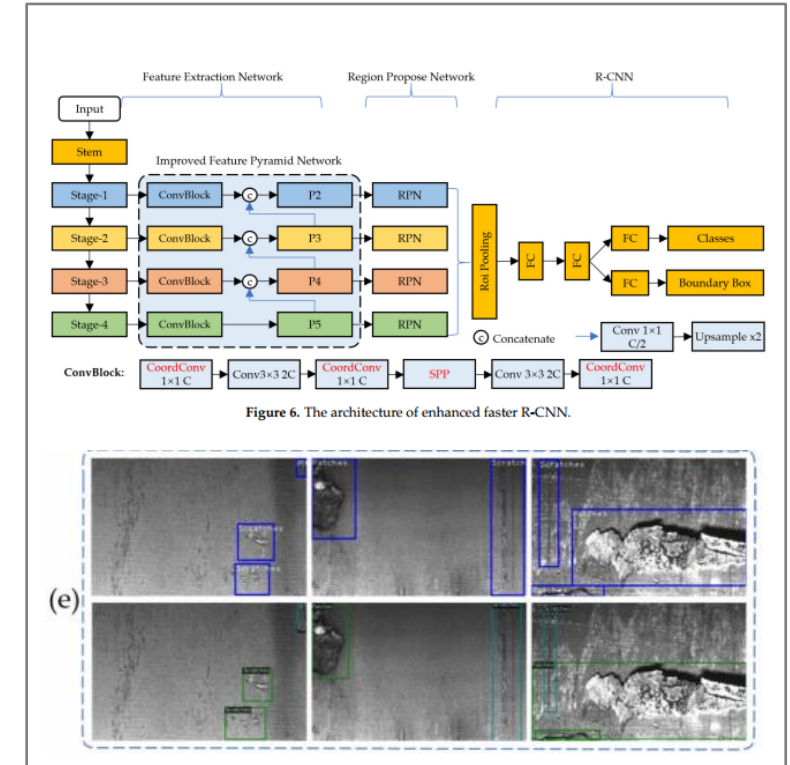
<http://jdsblog88.blogspot.com/2013/06/super-hero-movies-2013-iron-man-3-or.html>

제선 제강 연주 압연 도금

표면 결함과 데이터 불균형 문제



표면 결함 탐지 딥러닝 모델



❖ 철강 표면 결함 탐지 연구계획

- ✓ 여전히 해결되지 않은 데이터 불균형 문제에 대한 지속 연구 (Loss function, Data augmentation, Model...)
- ✓ 보다 정확한 결함 분류 및 결함 평점 산정을 위해 Semantic segmentation 모델 구현 추진

감사합니다.