

CNN을 활용한 OLED 디스플레이 결함 검출 시스템 구현

조국한*¹, 허영욱*², 조현우*³, 송영준**

Implementation of OLED Display Defect Detection System using CNN

Guk-Han Jo*¹, Young-Wook Heo*², Hyun-Woo Cho*³, and Young-Joon Song**

이 연구는 금오공과대학교 학술연구비로 지원되었음(202000970001)

요 약

최근에는 머신러닝 기술이 발전함에 따라 다양한 산업에서 이를 적용한 기술들이 개발되고 있다. 하지만 패널을 생산하는 공정에서는 아직 머신러닝을 결합하여 Mura를 검출하는 방법이 보편화되어 있지 않다. 현재 생산 공정에서 주로 사용되고 있는 AOI(Automated Optical Inspection) 기법은 사전에 입력된 Mura를 검출하는데 효과적이지만 새로운 Mura가 입력되면 검출하지 못하는 문제가 발생한다. 본 논문에서는 이러한 문제를 해결하기 위해 CNN(Convolutional Neural Networks)을 이용한 Mura 검출 시스템을 구현하였다. 패널의 이미지 데이터들은 29M 카메라와 Frame Grabber를 이용하여 수집하고 총 5단계로 구성된 데이터 가공 기법을 적용하여 정형화된 데이터를 생성하였다. 이후 생성된 데이터들을 CNN 모델의 입력으로 사용하여 Mura 검출을 수행하였다. 실험 결과 50개의 테스트 데이터에서 100%의 정확도를 얻을 수 있었다.

Abstract

Recently, as machine learning technology develops, technologies applying it are being developed in various industries. However, in the process of producing panels, Mura detecting method using machine learning is not yet common. The AOI(Automated Optical Inspection) technique, which is mainly used in the current production process, is effective in detecting typical mura pattern, but has a problem not detecting when a new mura pattern. In this paper, we implemented a mura detection system using CNN(Convolutional Neural Networks) to solve this problem. The image data of the panel was collected using a 29M camera and a frame grabber, and standardized data was generated by applying a data processing technique consisting of a total of 5 steps. Then, This generated data is used for CNN model for Mura detection. As a result, we confirmed the 100% accuracy Mura detection for 50 test data.

Keywords

machine learning, mura, AOI, CNN

* 금오공과대학교 전자공학과

- ORCID¹: <http://orcid.org/0000-0002-4384-2852>

- ORCID²: <http://orcid.org/0000-0003-2249-5318>

- ORCID³: <http://orcid.org/0000-0003-4504-0965>

** 금오공과대학교 전자공학과 교수(교신저자)

- ORCID: <http://orcid.org/0000-0002-9736-602X>

• Received: Jan. 17, 2022, Revised: Feb. 25, 2022, Accepted: Feb. 28, 2022

• Corresponding Author: Young-Joon Song

Dept. of EE, Kumoh National Inst of Tech, Korea

Tel.: + 82-54-478-7486, Email: yjsong@kumoh.ac.kr

1. 서 론

최근 코로나로 인해 화상회의나 강의와 같은 비대면 시스템이 활성화되면서 디스플레이가 탑재된 각종 IT 기기들의 활용성이 더 높아지고 있으며 화상회의와 강의뿐만 아니라 각종 소비활동에서도 언택트가 일상화되었다[1]. 또한 최근에는 스마트 이러닝과 같은 다양한 원격 교육시스템이 제안되고 있다[2]. 이러한 언택트 환경에서 필수적으로 사용되는 IT기기들은 대부분 사용자에게 시각적으로 데이터를 전달하는 디스플레이를 탑재하고 있다.

현재 디스플레이 생산과정에서는 다양한 공정들을 통해 디스플레이의 생산성을 높이고 있다. 그 중에서도 결함을 빠르게 검출하고 이를 보상하는 작업은 필수적으로 포함되는 공정이다. 디스플레이 생산에서 발생하는 결함은 Mura라고 불리며 발생 원인으로서는 작업 과정에서 실수나 부적합한 환경으로 인해 발생한다. 이러한 Mura의 발생은 디스플레이의 생산율을 낮추는 요인 중에 하나이며 대부분의 공정에서 이를 해결하기 위해 몇몇의 검출방법 및 보상방법이 수행되고 있다. 대표적으로 사용되는 검출기법으로는 AOI 검출 기법이 있으며 육안으로 검출을 수행하기도 한다[3].

AOI 검출 기법은 레이저 등과 같은 광학장치에서 출력되는 빛을 필름에 투과하여 발생하는 빛의 양을 이미지 카메라를 통해 Mura를 검출하는 방법이다. 육안검출 기법은 작업자가 직접 눈으로 관찰하여 검출하는 방법이다. 현재에는 AOI 검출기법과 머신러닝을 결합하여 결함을 검출하는 기술들도 개발되고 있으며, 가버웨이플릿 변환, Polynomial models 기법을 통한 Mura 검출 기법들도 제안되었다[4]-[6].

하지만 아직 머신러닝을 결합하여 Mura를 검출하는 방법은 생산 공정에서 보편화되어있지 않으며 대부분 AOI 검출이나 육안 검출을 사용하고 있다. 이러한 검출 방법은 문제점이 있는데 먼저 육안 검출 방법은 작업자의 주관적인 판단으로 인해 검출 결과가 달라질 수 있고 작업자의 눈에 상당한 피로를 줄 수 있다. AOI 검출은 사전에 Mura의 패턴을 입력하고 동작하기 때문에 새로운 패턴의 Mura가

발생하게 된다면 Mura가 없는 양품으로 인식하는 문제가 발생한다. 따라서 본 논문에서는 이러한 AOI, 육안 검출에서 발생하는 문제점을 해결하기 위해 머신러닝을 이용하여 Mura를 검출하는 시스템을 구현하였다.

II. 데이터 추출을 위한 주요 부품 및 기능

Mura 검출에 사용될 데이터들은 실제 공정에서 패널 이미지를 수집하여 가공한다. 수집된 데이터는 CNN의 학습데이터로 사용되기 때문에 가공과정에서 일정한 규격의 이미지 데이터를 추출해야한다. 이를 위해서 본 논문에서는 이미지 카메라와 아날로그 및 디지털 비디오 스트림에서 개별 디지털 스틸 프레임을 캡처하는 Frame grabber를 사용하였다.

그림 1의 카메라 렌즈는 6 inch의 모듈 대응을 위해서 약 1.5M까지 Working Distance가 필요하기 때문에 이에 적합한 OVM0504MN 모델[7]을 사용하였다. 그림 2의 카메라는 RMV-29 CL 모델[8]이며 해당 모델은 최대 QHD+(1440×2880)까지 보상이 가능하다. 또한, 디지털 변환을 위해서는 1:2의 픽셀 캡처가 요구되며 (1440×2)×(2880×2)의 CCD 카메라 해상도가 필요하다. 따라서 최종적으로 29M 카메라를 선정하였다.



그림 1. OVM0504MN 카메라 렌즈
Fig. 1. OVM0504MN camera lens



그림 2. RMV-29 CL 29M 카메라
Fig. 2. RMV-29 CL 29M camera



그림 3 PCIe 타입 Frame grabber
Fig. 3. PCIe type frame grabber

그림 3은 개별 디지털 스틸 프레임을 추출하기 위해 사용한 Solios eV-CL Frame grabber 모델[9]이며 전송속도를 고려하여 PCIe 타입을 사용하였다.

III. 패널 이미지 데이터 추출 및 가공

3.1 패널 이미지 추출 과정

본 논문 2장에서 소개한 카메라 장비와 Frame grabber를 활용하여 패널 이미지를 추출한다. 그림 4는 패널 이미지 추출 과정을 Flow chart를 이용하여 나타낸 것이다[10]. 먼저 준비된 카메라를 구동하면 패널 Positioning을 수행하고 이어 패널의 휘도 및 카메라 셋팅 과정을 수행한다. 이후 캡처를 시작하는데 캡처된 패널 이미지는 카메라 Setting과 결합되어 수집된다. 수집하는 과정에서 이미지 데이터가 정상적으로 캡처 되지 않았을 경우에는 캡처 시작 지점으로 다시 돌아가 캡처를 수행한다.

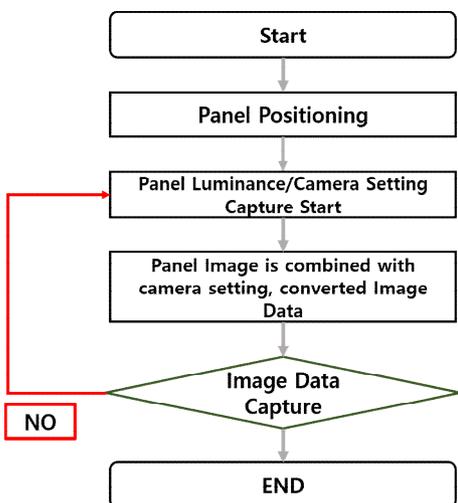


그림 4. 패널 이미지 추출 과정 flow chart
Fig. 4. Flow chart for panel image extraction process

3.2 패널 이미지 데이터 가공

Frame grabber를 사용하여 카메라에서 촬영한 이미지 데이터는 총 5단계의 과정을 통해 가공되어 생성된다. 1단계는 카메라에서 촬영한 이미지 데이터가 PC로 전송되어 bmp 파일로 변환하는 과정이며, 2단계는 노이즈 제거를 위해 필터를 적용하고 이진화를 수행하는 과정이다. 3단계는 이진화를 통해 패널이 발생한 비정상적인 부분을 복원한다. 마지막으로 4단계에서는 패널의 코너를 검출하고 5단계는 3차원 상의 위치와 회전을 제거하여 2차원의 이미지 형태로 변환한다.

3.3 BMP 파일 변환 및 이진화

그림 5는 패널 이미지 파일을 bmp 파일로 변환 후 이진화를 수행한 결과이다. (a)는 이진화를 수행하기 전 패널 이미지이며 (b)는 Otsu 알고리즘 [11]-[13]을 통해 이진화를 수행한 후의 결과이다. 먼저 Frame grabber를 이용하여 카메라에서 촬영된 데이터를 PC로 전송하면 MIL library를 사용하여 12 Bit binary 파일로 변환한다. 이후 12 Bit binary 파일로부터 패널 size 영상을 획득하여 BMP 파일로 변환한다.

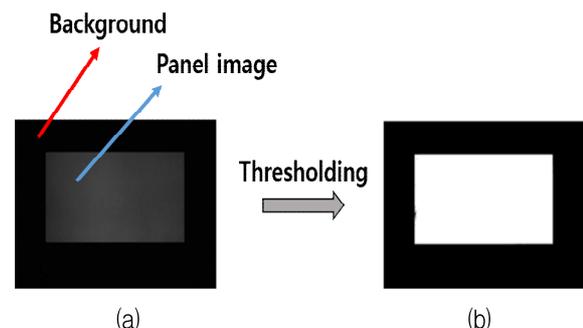


그림 5. 패널 이미지 파일의 threshold 이진화
Fig. 5. Threshold binarization of panel image file

BMP 파일로 변환된 이미지 파일은 패널 영역과 배경을 분리하기 위해 Otsu threshold 알고리즘을 사용한다. Otsu 이진화 방법은 이미지나 영상을 두 개의 클래스로 나누고 클래스 간의 편차가 최대가 되도록 임계값을 설정하는 방법이다.

4 CNN을 활용한 OLED 디스플레이 결함 검출 시스템 구현

두 개의 클래스는 이미지나 영상을 배경과 객체로 나누어 C_0, C_1 으로 정의하면 각 클래스의 확률 분포 w_0, w_1 와 평균 μ_0, μ_1 은 아래 수식과 같다.

$$w_0 = \Pr(C_0) = \sum_{i=1}^k p_i = w(k) \quad (1)$$

$$w_1 = \Pr(C_1) = \sum_{i=k+1}^L p_i = 1 - w(k) \quad (2)$$

$$\mu_0 = \sum_{i=1}^k i \Pr(i|C_0) = \sum_{i=1}^k i p_i / w_0 = \mu(k) / w(k) \quad (3)$$

$$\mu_1 = \sum_{i=k+1}^L i \Pr(i|C_1) = \sum_{i=k+1}^L i p_i / w_1 = \frac{\mu_T - \mu(k)}{1 - w(k)} \quad (4)$$

L 은 영상이 가지고 있는 Gray Level 개수이며, i 는 Gray level이다. 전체평균 μ_T 는 식 (5)와 같다.

$$\mu_T = \mu(L) = \sum_{i=1}^L i p_i \quad (5)$$

각 클래스 C_0, C_1 의 분산 σ_0^2, σ_1^2 은 아래와 같다.

$$\sigma_0^2 = \sum_{i=1}^k (i - \mu_0)^2 \Pr(i|C_0) = \sum_{i=1}^k (i - \mu_0)^2 p_i / w_0 \quad (6)$$

$$\sigma_1^2 = \sum_{i=k+1}^L (i - \mu_1)^2 \Pr(i|C_1) = \sum_{i=k+1}^L (i - \mu_1)^2 p_i / w_1 \quad (7)$$

각 클래스의 분산을 구했다면 이를 이용하여 클래스 내 분산 σ_W^2 와 클래스 간 분산 σ_B^2 , 전체 분산 σ_T^2 을 계산한다. 수식은 다음과 같다.

$$\sigma_W^2 = w_0 \sigma_0^2 + w_1 \sigma_1^2 \quad (8)$$

$$\begin{aligned} \sigma_B^2 &= w_0 (\mu_0 - \mu_T)^2 + w_1 (\mu_1 - \mu_T)^2 \\ &= w_0 w_1 (\mu_1 - \mu_0)^2 \end{aligned} \quad (9)$$

$$\sigma_T^2 = \sum_{i=1}^L (i - \mu_T)^2 p_i \quad (10)$$

클래스 간 분산 σ_B^2 에 w_0, w_1, μ_0, μ_1 을 대입하여 계산하면 아래 수식과 같다.

$$\sigma_B^2(k) = \frac{[\mu_T w(k) - \mu(k)]^2}{w(k)[1 - w(k)]} \quad (11)$$

식 (9)를 살펴보면 μ_0, μ_1 의 차가 클수록 σ_B^2 의 분산이 커짐을 알 수 있다. 즉 σ_B^2 의 분산이 커질수록 클래스간의 분리도가 커진다는 것이다. 클래스 간 분산 $\sigma_B^2(k)$ 이 최대일 때 최적의 임계값을 가지므로 다음과 같은 식으로 표현할 수 있다.

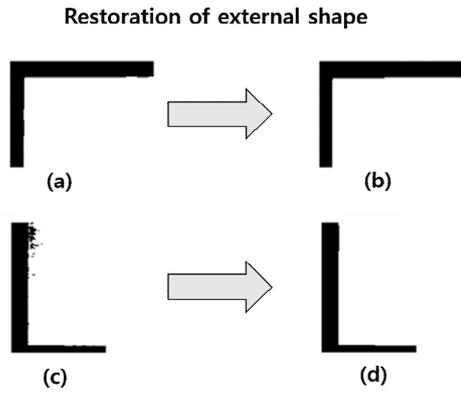
$$\sigma_B^2(k^*) = \max \sigma_B^2(k), 1 \leq k \leq L \quad (12)$$

식 (12)에서 확인할 수 있듯이 식 (11)을 최대화 하는 k^* 을 찾으면 Otsu 임계값을 얻을 수 있다.

3.4 패널 형태 복원

패널 이미지의 이진화를 수행하면 출력된 이미지에서 얼룩과 같은 형태의 비정상적인 부분을 확인할 수 있다. 이러한 얼룩과 같은 부분을 처리하기 위해 형태 복원 과정이 필요하다. 형태 복원과정은 내부 형태 복원과 외부 형태 복원 2가지를 수행한다.

그림 6은 패널의 형태 복원 과정 및 결과이다. (a)는 외부 형태 복원 전 패널 이미지이며, (b)는 외부 형태 복원 후 패널 이미지이다. 그림 (a)을 보면 내부의 프레임이 매끄럽지 않게 나타나는 것을 볼 수 있다. 하지만 외부 형태를 복원하여 출력한 (b)의 결과를 살펴보면 내부 프레임이 (a)에 비해 많이 매끄러워진 것을 볼 수 있다. 내부 형태 복원은 패널 내부에 발생하는 얼룩을 제거하는 것으로 (c)의 내부에 얼룩과 같은 비정상적인 부분이 있는 것을 확인할 수 있다. 이를 내부 형태 복원을 통해 복원한 결과는 그림에서 (d)와 같다. 이러한 얼룩 형태의 잡음들은 OpenCV의 Erosion과 Dilation 함수를 통해 복원한다.



Restoration of internal shape

그림 6. 패널 복원 과정 및 결과

Fig. 6. Panel restoration process and its results

3.5 패널 코너 검출

3.6절에서 수행할 3차원 위치 및 회전 제거를 위해서 패널의 코너 포인트를 검출하는 과정이 필요하다. 코너 검출 과정은 다음과 같다.

그림 7은 패널의 코너 검출 과정을 나타낸 것이다. 코너 검출과정은 패널의 이미지를 4분할하여 중심에서부터 분할된 영역별로 코너를 검색하고 각 영역별로 거리가 가장 먼 4개의 포인트를 코너 포인트로 검출한다.

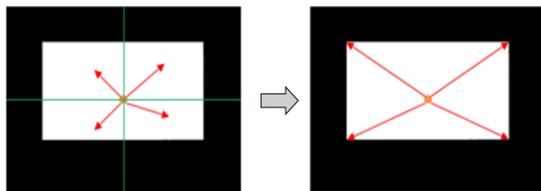


그림 7. 패널의 코너 검출 과정

Fig. 7. Panel corner detection process

3.6 패널의 3차원 위치 및 회전 제거

실제 카메라를 통해 촬영된 영상은 다양한 환경에 따라 필연적으로 미세한 회전이나 위치 변화가 반영된다. 이는 규격화된 데이터를 생성하는데 있어 필수적으로 제거해야 하는 부분이다. 본 논문에서는 IPT(Inverse Perspective Transform)[14]을 이용하여 3차원 위치 및 회전을 제거하고 평면 이미지 데이터를 생성하였다.

그림 8은 IPT를 간단한 그림으로 나타낸 것이다.

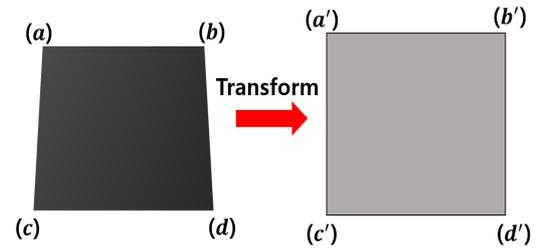


그림 8. Inverse perspective transform

Fig. 8. Inverse perspective transform

IPT는 주로 카메라로 촬영한 이미지나 영상에서 3차원 위치나 회전을 제거하여 2차원 평면 형태로 변환할 때 사용된다. 수식은 아래와 같다.

$$Q' = wHQ \quad (12)$$

Q 는 실제 영상이며 Q' 는 변환된 영상, w 는 스케일 인자, H 는 평면 호모그래피 행렬이다. 3차원 위치, 회전을 제거하면 최종적으로 규격화된 데이터가 생성된다. 생성된 패널 이미지는 720×1920 사이즈를 가지고 있으며 이 데이터를 CNN 모델에 입력하여 학습한다.

IV. 패널 이미지 데이터 학습 및 테스트

3장에서 생성한 패널 이미지를 검증하기 위해 CNN 모델을 활용하여 학습 후 Mura 검출을 수행하였다. CNN의 입력데이터로는 3장에서 생성한 720×1920 사이즈의 패널 이미지를 480×480로 resize 하고 0과 1사이의 값으로 정규화하였다.

초기 모델 학습에는 총 350개의 패널 이미지 데이터가 사용되었으며 Mura가 포함된 패널 이미지 175개, 정상적인 패널 이미지 175개를 사용하여 학습을 진행하였다. 모든 패널 이미지 데이터는 Mura가 포함된 패널 이미지와 정상적인 패널 이미지로 육안으로 구별되지 않는 데이터를 사용하였다.

그림 9는 Mura 검출을 위해 구성한 CNN 모델의 구조이다. 구성한 모델은 크게 2개의 Convolutional layer와 Pooling layer, 1개의 Flatten layer와 2개의 Fully connected layer로 구성되어 있다. 또한, 각 Layer들 사이에는 Dropout을 적용하여 과적합을 방지하였다. Convolutional layer는 입력된 패널 이미지

데이터들의 특징을 추출하며, 추출된 정보들은 Maxpooling을 통해 특징 표본을 추출한다. 이 과정이 끝나면 데이터를 펼치는 Flatten layer를 거쳐 최종적으로 Fully connected layer를 통해 결과를 출력한다. 사용된 활성화함수는 딥러닝에서 많이 사용되고 있는 ReLU이며, Optimazer로는 Adam을 사용하였다. 마지막 Output layer는 앞선 노드의 모든 결과를 받아 Softmax를 통해 분류를 수행한다.

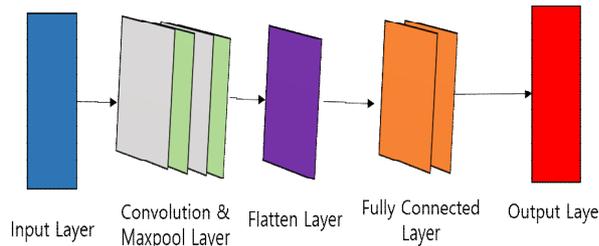


그림 9. 구성한 CNN 모델
Fig. 9. Constructed CNN model

그림 10은 시뮬레이션 결과를 표현한 히트맵이다. 학습 후 테스트에서는 Mura가 포함된 패널 이미지 25개, 정상적인 패널 이미지 25개로 총 50개의 데이터를 사용하였다. 결과를 보면 모든 데이터를 정확하게 분류한 것을 확인할 수 있다.

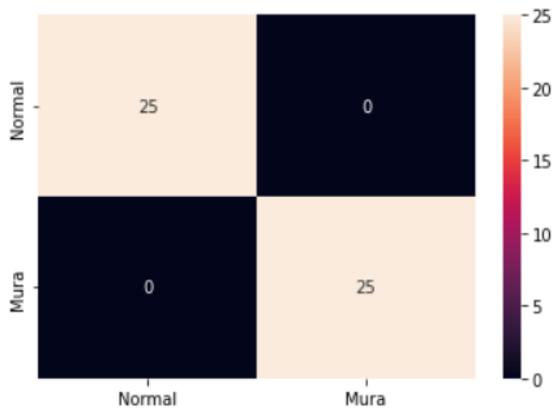


그림 10. 시뮬레이션 결과 히트맵
Fig. 10. Hitmap of simulation result

[3]에서는 패널 이미지를 받아 5개의 Convolutional layer와 2개의 Fully connected layer를 이용하여 학습을 수행하였다. 이때 테스트 수행 결과로 96%의 정확도를 얻을 수 있었다. 하지만 본 논문에서는 오히려 Convolutional layer를 줄였으며 데이터를 펴는

Flatten layer를 추가하고 Dropout를 적절히 활용한 결과 50개의 테스트 데이터를 모두 정확하게 분류하여 100% 정확도를 얻을 수 있었다.

V. 결론 및 향후 과제

머신러닝 기술이 발전함에 따라 생산 공정, 서비스업, 차량 등 모든 분야에서 예외 없이 머신러닝 기술이 사용되고 있다. 본 논문에서는 현재 각광받고 있는 딥러닝 기술 CNN을 이용하여 생산 공정에서 디스플레이 패널의 결함을 검출하는 시스템을 구현하였다. 이미지 수집을 위해 29M 카메라와 OVM0504MN 모델, PCIe 타입 Frame grabber를 사용하고 수집한 데이터들을 총 5단계의 전처리 과정을 통해 가공하여 규격화된 데이터들을 생성하였다. 전처리 과정은 카메라 장비를 통해 수집한 데이터들을 BMP 파일로 변환하고 Otsu 이진화 알고리즘을 통해 노이즈를 제거하였다. 이후 패널 이미지의 코너를 영역별로 검출하고 IPT를 적용하여 3차원 위치 및 회전을 제거하였다. 이렇게 생성된 이미지 데이터들은 CNN 모델의 입력으로 사용되어 학습하였다. 그 결과, [3]보다 향상된 정확도를 얻을 수 있었다. 하지만 생산 공정을 통해 대량으로 데이터를 수집해야 하는 패널 데이터 특성상 현재 논문에서 수집한 데이터보다 더 많은 양의 데이터를 활용해 시뮬레이션을 수행하는 것이 필요하다. 또한, AOI 기법에서 주로 수행하는 Mura 패턴 분석 및 보상과 같은 시스템 구현을 위해서는 충분한 양의 데이터 수집과 세부적인 모델 개선이 필요할 것으로 예상된다.

References

- [1] <https://www.hankookilbo.com/News/Read/202003201071329289>. [accessed: Jan. 29, 2022]
- [2] YouSik Hong, "Implementation of Smart E-learning based on Blended Learning", *JIBC*, Vol. 20, No. 2, pp. 171-178, Apr. 2020. <https://doi.org/10.7236/JIBC.2020.20.2.171>.
- [3] Guk-Han Jo, Hyun-Woo Cho, Youngwook Heo,

- and Young-Joon Song, "Applications of CNN to LCD-Panel Mura Detection", Summer Annual Conference of IEIE, Vol. 2020, No. 8, pp. 1992-1995, Aug. 2020.
- [4] H. Y. Sung and Y. J. Jeon, "MURA inspection and correction using Polynomial Models", ICROS, Vol. 2019, No. 5, pp. 13-14, May 2019.
- [5] Hwang-Ju Lee and Tae-Hyoung Park, "Mura Defect Detection of Display Panel Using Gabor Wavelet Transformation and Image Post-Processing", Journal of Institute of Control, Robotics and Systems, Vol. 23, No. 4, pp. 287-293, Apr. 2017. <https://doi.org/10.5302/J.ICROS.2017.16.0209>.
- [6] <http://laonpeople.com/main/main.php>. [accessed : Jan. 29, 2022]
- [7] <https://www.illunis.com/wp-content/uploads/29MP-CL-CCD.pdf>. [accessed : Jan. 19, 2022]
- [8] https://www.nikon.com/products/industrial-lenses/assets/pdf/rayfact-pamphlet_en.pdf. [accessed : Jan. 3, 2022]
- [9] <https://www.matrox.com/sites/default/files/matrox-solis-os-ev-cl.pdf>. [accessed : Jan. 10, 2022]
- [10] Guk-han Jo, Young-wook Heo, and Young-Joon Song, "Image Extraction Method for Defect Detection of Display Panel", The proceeding the 2021 KIIT Autumn Conference, Vol. 16, No. 2, pp. 246-248, Nov. 2021.
- [11] N. Otsu, "A Threshold Selection Method from Gray Level Histograms", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, pp. 62-66, Jan. 1979. <https://doi.org/10.1109/TSMC.1979.4310076>.
- [12] Ho-Min Jang, Eui-Ryoung Kim, Se-Chun Oh, Sin-Ryeong Kim and Young-Gon Kim, "Embedded Multi-LED Display System based on Wireless Internet using Otsu Algorithm", JIIBC, Vol. 16, No. 6, pp. 329-336, Dec. 2016. <https://doi.org/10.7236/JIIBC.2016.16.6.329>.
- [13] Young Woo Lee and Jin Heon Kim, "A

Computational Improvement of Otsu's Algorithm by Estimating Approximate Threshold", Journal of Korea Multimedia Society, Vol. 20, No. 2, pp. 163-169, Jan. 2017. <https://doi.org/10.9717/kmms.2017.20.2.163>.

- [14] H. Hwan and Gi-Tae Han, "A Robust Real-Time Lane Detection for Sloping Roads", KIPS Tr. Software and Data Eng, Vol. 2, No. 6, pp. 413-422, Feb. 2013. <https://doi.org/10.3745/KTSDE.2013.2.6.413>.

저자소개

조 국 한 (Guk-Han Jo)



2016년 2월 : 금오공과대학교
컴퓨터공학과 공학사
2019년 2월 : 금오공과대학교
전자공학과 공학석사
2019년 3월 ~ 현재 : 금오공과
대학교 전자공학과 박사과정
관심분야 : 머신러닝, 채널코딩

허 영 욱 (Young-Wook Heo)



2002년 2월 : 성공회대학교
정보통신학과 공학사
2022년 2월 : 금오공과대학교
전자공학과 공학석사
2001년 10월 ~ 현재 :
(주)에이피솔루션 이사
관심분야 : OLED, 딥러닝

조 현 우 (Hyun-Woo Cho)



2018년 8월 : 금오공과대학교
컴퓨터공학과 공학사
2018년 9월 : 금오공과대학교
전자공학과 공학석사
2020년 3월 ~ 현재 : 금오공과
대학교 전자공학과 박사과정
관심분야 : 머신러닝, 채널코딩

8 CNN을 활용한 OLED 디스플레이 결함 검출 시스템 구현

송 영 준 (Young-Joon Song)



1987년 2월 : 한양대학교

전자통신공학과 공학사

1994년 2월 : 한양대학교

전자통신공학과 공학석사

1999년 2월 : 한양대학교

전자통신공학과 공학박사

1994년 10월 ~ 2002년 2월 :

LG전자 책임연구원

2002년 3월 ~ 현재 : 금오공과대학교 전자공학부 교수

IEEE senior member

관심분야 : 머신러닝, 채널코딩, 이동통신 시스템