



(19) 대한민국특허청(KR)

(12) 등록특허공보(B1)

(45) 공고일자 2024년07월12일

(11) 등록번호 10-2684087

(24) 등록일자 2024년07월08일

(51) 국제특허분류(Int. Cl.)

H04B 10/116 (2013.01) G06K 7/14 (2006.01)
G06N 3/0464 (2023.01) G06N 3/08 (2023.01)
G06N 3/094 (2023.01) G06V 10/56 (2022.01)
G06V 10/764 (2022.01) G06V 10/82 (2022.01)
H04B 10/50 (2013.01) H04N 5/77 (2006.01)

(52) CPC특허분류

H04B 10/116 (2013.01)
G06K 7/1443 (2013.01)

(21) 출원번호 10-2023-0033888

(22) 출원일자 2023년03월15일

심사청구일자 2023년03월15일

(56) 선행기술조사문헌

KR102269750 B1

(뒷면에 계속)

전체 청구항 수 : 총 14 항

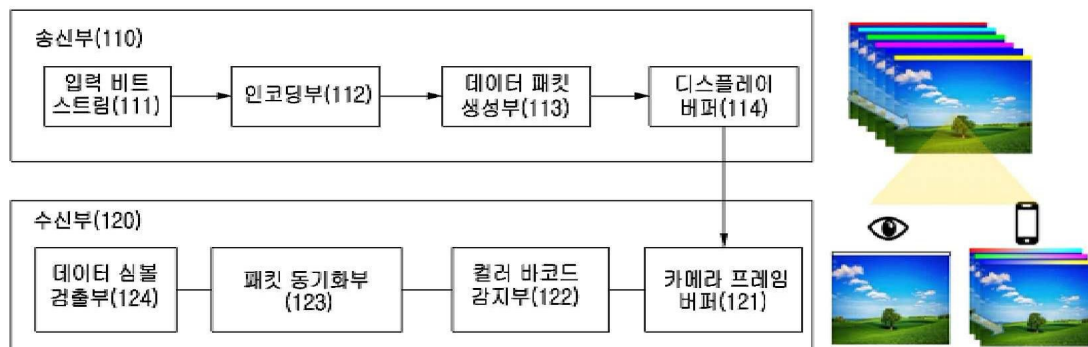
심사관 : 신상길

(54) 발명의 명칭 딥러닝을 이용한 보색 바코드 기반 광카메라 통신 방법 및 시스템

(57) 요약

딥러닝을 이용한 보색 바코드 기반 광카메라 통신 방법 및 시스템이 제시된다. 본 발명에서 제안하는 딥러닝을 이용한 보색 바코드 기반 광카메라 통신 방법은 광카메라 통신을 위한 송신부의 인코딩부를 통해 입력 비트 스트림의 이진 데이터를 컬러 바코드로 인코딩하여 지정된 색상 값에 매핑하고 데이터 패킷으로 구성하는 단계, 상기

대표도



인코딩된 컬러 바코드에 대하여 데이터 패킷 생성부를 통해 모든 심볼이 보색 쌍으로 구성되는 연속적인 데이터 패킷을 생성하고, 디스플레이 버퍼를 통해 송신하는 단계, 광카메라 통신을 위한 수신부를 통해 상기 송신되는 데이터 패킷에 대하여 서터 카메라를 이용하여 표시 영역을 캡처하고, 캡처된 이미지는 카메라 프레임 버퍼에 저장하는 단계, 상기 저장된 이미지에 대해 컬러 바코드 감지부를 통해 딥 러닝 모델을 사용하여 컬러 바코드 영역이 감지되면 컬러 바코드 이미지를 이미지 목록에 저장하는 단계, 패킷 동기화부를 통해 지속적인 패킷 동기화를 위한 딥 러닝 기반 파일럿 심볼 검출 프로세스를 수행하여 각 패킷의 시작점을 식별하고 동기화하는 단계 및 상기 동기화가 완료되면, 각 패킷 내의 데이터에 대해 딥 러닝을 사용하여 데이터 심볼 검출부를 통해 연속적으로 디코딩하는 단계를 포함한다.

(52) CPC특허분류	(56) 선행기술조사문헌
<i>G06N 3/0464</i> (2023.01)	KR1020230053347 A
<i>G06N 3/082</i> (2023.01)	KR102324926 B1
<i>G06N 3/094</i> (2023.01)	KR102224861 B1
<i>G06V 10/56</i> (2023.08)	Jung, S.-Y.; Lee, J.-H.; Nam, W.; Kim, B.W.
<i>G06V 10/765</i> (2023.08)	“Complementary Color Barcode-Based Optical
<i>G06V 10/82</i> (2022.01)	Camera Communications,” Wirel. Commun. Mob.
<i>H04B 10/502</i> (2013.01)	Comput. (2020)
<i>H04N 5/77</i> (2013.01)	

이 발명을 지원한 국가연구개발사업	
과제고유번호	1711153443
과제번호	2022R1A2B5B01001543
부처명	과학기술정보통신부
과제관리(전문)기관명	한국연구재단
연구사업명	중견연구자지원사업
연구과제명	교통정보 서비스 고도화를 위한 자유공간 광전송 기반 저궤도 위성통신의 빔 정밀조
향 기술 연구	
기 여 율	1/1
과제수행기관명	창원대학교
연구기간	2022.03.01 ~ 2023.02.28
공지예외적용	: 있음

명세서

청구범위

청구항 1

광카메라 통신을 위한 송신부의 인코딩부를 통해 입력 비트 스트림의 이진 데이터를 컬러 바코드로 인코딩하여 지정된 색상 값에 매핑하고 데이터 패킷으로 구성하는 단계;

상기 인코딩된 컬러 바코드에 대하여 데이터 패킷 생성부를 통해 모든 심볼이 보색 쌍으로 구성되는 연속적인 데이터 패킷을 생성하고, 디스플레이 버퍼를 통해 송신하는 단계;

광카메라 통신을 위한 수신부를 통해 상기 송신되는 데이터 패킷에 대하여 셔터 카메라를 이용하여 표시 영역을 캡처하고, 캡처된 이미지는 카메라 프레임 버퍼에 저장하는 단계;

상기 저장된 이미지에 대해 컬러 바코드 감지부를 통해 딥 러닝 모델을 사용하여 컬러 바코드 영역이 감지되면 컬러 바코드 이미지를 이미지 목록에 저장하는 단계;

패킷 동기화부를 통해 지속적인 패킷 동기화를 위한 딥 러닝 기반 파일럿 심볼 검출 프로세스를 수행하여 각 패킷의 시작점을 식별하고 동기화하는 단계; 및

상기 동기화가 완료되면, 각 패킷 내의 데이터에 대해 딥 러닝을 사용하여 데이터 심볼 검출부를 통해 연속적으로 디코딩하는 단계

를 포함하는 광카메라 통신 방법.

청구항 2

제1항에 있어서,

상기 저장된 이미지에 대해 컬러 바코드 감지부를 통해 딥 러닝 모델을 사용하여 컬러 바코드 영역이 감지되면 컬러 바코드 이미지를 이미지 목록에 저장하는 단계는,

상기 각 패킷 내의 데이터에 대한 디코딩을 수행하기 위해 상기 저장된 이미지 내에서 컬러 바코드 영역만을 감지하고, 디스플레이 버퍼와 셔터 카메라 사이의 비동기를 감소시키기 위해 컬러 바코드 영역의 주석을 포함하는 데이터를 이용하여 YOLO 모델을 통해 컬러 바코드 영역을 감지하는

광카메라 통신 방법.

청구항 3

제2항에 있어서,

상기 저장된 이미지에 대해 컬러 바코드 감지부를 통해 딥 러닝 모델을 사용하여 컬러 바코드 영역이 감지되면 컬러 바코드 이미지를 이미지 목록에 저장하는 단계는,

상기 YOLO 모델의 넥(neck) 구조는 공간 피라미드 풀링(SPP)과 경로 집합 네트워크(PAN)를 사용하고, 훈련 비용을 증가시켜 학습 정확도를 향상시키고, 예측 정확도를 높이기 위해 BoF(Bag of Freebies) 및 BoS(Bag of Specials) 기법을 사용하며, 이미지의 일부 영역에 노이즈를 추가하고, 컷 아웃(CutOut)을 통해 일부 영역을 제거하며, 믹스 업(MixUp)을 통해 레이블이 다른 두 이미지를 혼합하고, 컷 믹스(CutMix)를 통해 레이블이 있는 두 이미지를 잘라내어 붙여 넣는

광카메라 통신 방법.

청구항 4

제2항에 있어서,

상기 저장된 이미지에 대해 컬러 바코드 감지부를 통해 딥 러닝 모델을 사용하여 컬러 바코드 영역이 감지되면 컬러 바코드 이미지를 이미지 목록에 저장하는 단계는,

컬러 바코드 영역에서 추출된 좌표를 이미지 목록에 저장하고, 저장된 좌표에서 x축 및 y축 좌표의 최소값과 높이 및 너비의 최대값을 찾는 과정을 통해 컬러 바코드 정보의 손실 없이 가능한 가장 작은 크기의 영역을 획득하는

광카메라 통신 방법.

청구항 5

제1항에 있어서,

상기 패킷 동기화부를 통해 지속적인 패킷 동기화를 위한 딥 러닝 기반 파일럿 심볼 검출 프로세스를 수행하여 각 패킷의 시작점을 식별하고 동기화하는 단계는,

3개의 파일럿 심볼(R, G, B) 및 8개의 데이터 심볼(0-7)을 갖는 총 11개의 컬러 바코드의 클래스는 디스플레이 버퍼와 셔터 카메라 사이의 비동기를 감소시키기 위해 각 클래스에 대하여 상기 감지된 컬러 바코드의 패턴에 대한 추가 레이블링을 수행하는

광카메라 통신 방법.

청구항 6

제5항에 있어서,

상기 패킷 동기화부를 통해 지속적인 패킷 동기화를 위한 딥 러닝 기반 파일럿 심볼 검출 프로세스를 수행하여 각 패킷의 시작점을 식별하고 동기화하는 단계는,

패킷 동기화를 위해 상기 감지된 컬러 바코드의 패턴의 입력으로부터 파일럿 심볼(R, G, B)의 패턴을 개별 클래스로 지정하고 데이터 심볼에 나타나는 패턴을 하나의 클래스로 묶어 총 4개의 동기화 클래스를 분류하도록 훈련함으로써 파일럿 심볼과 데이터 심볼을 식별하고, 파일럿 심볼의 종류를 인식하는

광카메라 통신 방법.

청구항 7

제6항에 있어서,

상기 동기화가 완료되면, 각 패킷 내의 데이터에 대해 딥 러닝을 사용하여 데이터 심볼 검출부를 통해 연속적으로 디코딩하는 단계는,

총 11개의 컬러 바코드의 클래스, 총 4개의 동기화 클래스 및 복수의 이미지 캡처 각수에 따라 데이터 심볼 검출을 위한 딥 러닝 훈련 데이터 셋을 구축하고, 확대, 축소, 회전, 위치 변환, 수직 반전을 포함하는 데이터 증강 기법을 이용하여 합성 데이터를 생성하여 심볼 검출을 위한 딥 러닝 수행하는

광카메라 통신 방법.

청구항 8

광카메라 통신시스템에 있어서,

송신부 및 수신부를 포함하고,

상기 송신부는,

입력 비트 스트림의 이진 데이터를 컬러 바코드로 인코딩하여 지정된 색상 값에 매핑하고 데이터 패킷으로 구성하는 인코딩부;

상기 인코딩된 컬러 바코드에 대하여 모든 심볼이 보색 쌍으로 구성되는 연속적인 데이터 패킷을 생성하는 데이터 패킷 생성부; 및

상기 생성된 연속적인 데이터 패킷을 송신하는 디스플레이 버퍼

를 포함하고,

상기 수신부는,

상기 송신되는 데이터 패킷에 대하여 셔터 카메라를 이용하여 표시 영역을 캡처하고, 캡처된 이미지를 저장하는 카메라 프레임 버퍼;

상기 저장된 이미지에 대해 딥 러닝 모델을 사용하여 컬러 바코드 영역이 감지하고, 감지된 컬러 바코드의 이미지를 이미지 목록에 저장하는 컬러 바코드 감지부;

지속적인 패킷 동기화를 위한 딥 러닝 기반 파일럿 심볼 검출 프로세스를 수행하여 각 패킷의 시작점을 식별하고 동기화하는 패킷 동기화부; 및

상기 동기화가 완료되면, 각 패킷 내의 데이터에 대해 딥 러닝을 사용하여 연속적으로 디코딩을 수행하는 데이터 심볼 검출부

를 포함하는 광카메라 통신 시스템.

청구항 9

제8항에 있어서,

상기 컬러 바코드 감지부는,

상기 각 패킷 내의 데이터에 대한 디코딩을 수행하기 위해 상기 저장된 이미지 내에서 컬러 바코드 영역만을 감지하고, 디스플레이 버퍼와 셔터 카메라 사이의 비동기를 감소시키기 위해 컬러 바코드 영역의 주석을 포함하는 데이터를 이용하여 YOLO 모델을 통해 컬러 바코드 영역을 감지하는

광카메라 통신 시스템.

청구항 10

제9항에 있어서,

상기 컬러 바코드 감지부는,

상기 YOLO 모델의 넥(neck) 구조는 공간 피라미드 풀링(SPP)과 경로 집합 네트워크(PAN)를 사용하고, 혼련 비용을 증가시켜 학습 정확도를 향상시키고, 예측 정확도를 높이기 위해 BoF(Bag of Freebies) 및 BoS(Bag of Specials) 기법을 사용하며, 이미지의 일부 영역에 노이즈를 추가하고, 컷 아웃(CutOut)을 통해 일부 영역을 제거하며, 믹스 업(MixUp)을 통해 레이블이 다른 두 이미지를 혼합하고, 컷 믹스(CutMix)를 통해 레이블이 있는 두 이미지를 잘라내어 붙여 넣는

광카메라 통신 시스템.

청구항 11

제9항에 있어서,

상기 컬러 바코드 감지부는,

컬러 바코드 영역에서 추출된 좌표를 이미지 목록에 저장하고, 저장된 좌표에서 x축 및 y축 좌표의 최소값과 높이 및 너비의 최대값을 찾는 과정을 통해 컬러 바코드 정보의 손실 없이 가능한 가장 작은 크기의 영역을 획득하는

광카메라 통신 시스템.

청구항 12

제8항에 있어서,

상기 패킷 동기화부는,

3개의 파일럿 심볼(R, G, B) 및 8개의 데이터 심볼(0-7)을 갖는 총 11개의 컬러 바코드의 클래스는 디스플레이 버퍼와 셔터 카메라 사이의 비동기를 감소시키기 위해 각 클래스에 대하여 상기 감지된 컬러 바코드의 패턴에 대한 추가 레이블링을 수행하는

광카메라 통신 시스템.

청구항 13

제12항에 있어서,

상기 패킷 동기화부는,

패킷 동기화를 위해 상기 감지된 컬러 바코드의 패턴의 입력으로부터 파일럿 심볼(R, G, B)의 패턴을 개별 클래스로 지정하고 데이터 심볼에 나타나는 패턴을 하나의 클래스로 묶어 총 4개의 동기화 클래스를 분류하도록 훈련함으로써 파일럿 심볼과 데이터 심볼을 식별하고, 파일럿 심볼의 종류를 인식하는

광카메라 통신 시스템.

청구항 14

제13항에 있어서,

상기 데이터 심볼 검출부는,

총 11개의 컬러 바코드의 클래스, 총 4개의 동기화 클래스 및 복수의 이미지 캡처 각수에 따라 데이터 심볼 검출을 위한 딥 러닝 훈련 데이터 셋을 구축하고, 확대, 축소, 회전, 위치 변환, 수직 반전을 포함하는 데이터 증강 기법을 이용하여 합성 데이터를 생성하여 심볼 검출을 위한 딥 러닝 수행하는

광카메라 통신 시스템.

발명의 설명

기술 분야

[0001] 본 발명은 딥러닝을 이용한 보색 바코드 기반 광카메라 통신 방법 및 시스템에 관한 것이다.

배경 기술

[0002] 최근 몇 년 동안 비디오 콘텐츠는 광범위한 비즈니스, 광고 및 문화에서 그 어느 때보다 중요해졌다. 광고판이나 다이렉트 메일과 같은 전통적인 마케팅 방법과 달리 비디오 콘텐츠는 보다 개인화되고 몰입감 있는 큐레이션된 정보를 제공할 수 있다. 이러한 영상 콘텐츠에 대한 수요가 증가함에 따라 스마트폰, 모바일 태블릿 등의 휴대기기로 영상을 시청하면서 동시에 부가적인 정보를 전달하는 새로운 패러다임에 대한 요구가 높아지고 있다. 이 서비스를 제공하기 위한 한 가지 직관적인 솔루션은 기존의 RF(Radio Frequency) 기반 통신을 사용하는 것이다. 그러나 무선 연결에는 명시적인 RF 장치가 필요하며 RF 신호의 브로드캐스팅 특성으로 인해 원치 않는 장치로 정보가 스페밍되거나 누출될 수 있다. 이러한 문제를 해결할 수 있는 후보로 카메라 수신기로 디코딩할 수 있는 화면 이미지의 정보를 인코딩하는 D2C(Display-to-Camera Communications)가 주목 받고 있다. 기성 전자 디스플레이의 높은 재생률($\geq 120\text{Hz}$) 덕분에 프레임률이 높은 비디오에서 데이터를 숨겨 인식률이 훨씬 낮은($40\sim 50\text{Hz}$) 사람의 눈을 속일 수 있다. D2C 기술은 주로 통신 거리가 짧은 전자 디스플레이와 카메라 센서가 있는 환경이라면 어디든 적용할 수 있는 가시광선 통신의 일종이다.

[0003] 종래기술에서는 디스플레이의 밝기와 질감에 따라 변하는 적응형 임베딩 방식인 크로마코드(ChromaCode)를 제안하였다. 기존 방식과 달리 CCB OCC(Complementary Color Barcode-based Optical Camera Communications)는 인간의 눈에 흰색 색상 막대로 인지될 수 있도록 세심하게 설계된 보색 쌍으로 기호를 전송하는 방식이지만 개별 색상 패턴은 카메라로 감지할 수 있다.

[0004] 그러나 기존의 CCB-OCC 시스템은 차분 영상을 이용하여 컬러 바코드 영역을 감지하지만 카메라 거리와 각도에 크게 영향을 받아 바코드 영역을 정확하게 감지하지 못할 수 있다. 또한 휴리스틱 히스토그램 분석을 통한 색상 값 추출은 패킷 동기화 및 데이터 디코딩에서 강력한 성능을 보장할 수 없다.

선행기술문헌

비특허문헌

[0005] (비특허문헌 0001) Jung, S.-Y.; Lee, J.-H.; Nam, W.; Kim, B.W. Complementary Color Barcode-Based Optical Camera Communications. Wirel. Commun. Mob. Comput. 2020, 2020, 3898427.

발명의 내용

해결하려는 과제

[0006] 본 발명이 이루고자 하는 기술적 과제는 안정적인 데이터 전송을 지원하는 감지 불가능한 D2C 통신 시스템인 딥 러닝을 이용한 보색 바코드 기반 광카메라 통신(Deep learning-driven Complementary Color Barcode-based Optical Camera Communications; DeepCCB-OCC) 방법 및 시스템을 제공하는데 있다. D2C 링크에서 끊김 없는 바코드 이미지 영역을 감지하기 위해 YOLO(You Only Look Once) 기반 바코드 영역 감지 모델이 구현되고, CNN(Convolutional Neural Network) 모델은 컬러 바코드에 내장된 데이터를 추출하여 패킷 동기화 및 데이터 디코딩을 수행한다.

과제의 해결 수단

[0007] 일 측면에 있어서, 본 발명에서 제안하는 딥러닝을 이용한 보색 바코드 기반 광카메라 통신 방법은 광카메라 통신을 위한 송신부의 인코딩부를 통해 입력 비트 스트림의 이진 데이터를 컬러 바코드로 인코딩하여 지정된 색상 값에 매핑하고 데이터 패킷으로 구성하는 단계, 상기 인코딩된 컬러 바코드에 대하여 데이터 패킷 생성부를 통해 모든 심볼이 보색 쌍으로 구성되는 연속적인 데이터 패킷을 생성하고, 디스플레이 버퍼를 통해 송신하는 단계, 광카메라 통신을 위한 수신부를 통해 상기 송신되는 데이터 패킷에 대하여 셔터 카메라를 이용하여 표시 영역을 캡처하고, 캡처된 이미지는 카메라 프레임 버퍼에 저장하는 단계, 상기 저장된 이미지에 대해 컬러 바코드 감지부를 통해 딥 러닝 모델을 사용하여 컬러 바코드 영역이 감지되면 컬러 바코드 이미지를 이미지 목록에 저장하는 단계, 패킷 동기화부를 통해 지속적인 패킷 동기화를 위한 딥 러닝 기반 파일럿 심볼 검출 프로세스를 수행하여 각 패킷의 시작점을 식별하고 동기화하는 단계 및 상기 동기화가 완료되면, 각 패킷 내의 데이터에 대해 딥 러닝을 사용하여 데이터 심볼 검출부를 통해 연속적으로 디코딩하는 단계를 포함한다.

[0008] 상기 저장된 이미지에 대해 컬러 바코드 감지부를 통해 딥 러닝 모델을 사용하여 컬러 바코드 영역이 감지되면 컬러 바코드 이미지를 이미지 목록에 저장하는 단계는 상기 각 패킷 내의 데이터에 대한 디코딩을 수행하기 위해 상기 저장된 이미지 내에서 컬러 바코드 영역만을 감지하고, 디스플레이 버퍼와 셔터 카메라 사이의 비동기를 감소시키기 위해 컬러 바코드 영역의 주석을 포함하는 데이터를 이용하여 YOLO 모델을 통해 컬러 바코드 영역을 감지한다.

[0009] 상기 저장된 이미지에 대해 컬러 바코드 감지부를 통해 딥 러닝 모델을 사용하여 컬러 바코드 영역이 감지되면 컬러 바코드 이미지를 이미지 목록에 저장하는 단계는 상기 YOLO 모델의 넥(neck) 구조는 공간 피라미드 풀링(SPP)과 경로 집합 네트워크(PAN)를 사용하고, 훈련 비용을 증가시켜 학습 정확도를 향상시키기 위해 BoF(Bag of Freebies) 및 BoS(Bag of Specials) 기법을 사용하며, 이미지의 일부 영역에 노이즈를 추가하고, 컷 아웃(CutOut)을 통해 일부 영역을 제거하며, 믹스 업(MixUp)을 통해 레이블이 다른 두 이미지를 혼합하고, 컷 믹스(CutMix)를 통해 레이블이 있는 두 이미지를 잘라내어 붙여 넣는다.

[0010] 상기 저장된 이미지에 대해 컬러 바코드 감지부를 통해 딥 러닝 모델을 사용하여 컬러 바코드 영역이 감지되면 컬러 바코드 이미지를 이미지 목록에 저장하는 단계는 컬러 바코드 영역에서 추출된 좌표를 이미지 목록에 저장하고, 저장된 좌표에서 x축 및 y축 좌표의 최소값과 높이 및 너비의 최대값을 찾는 과정을 통해 컬러 바코드 정보의 손실 없이 가능한 가장 작은 크기의 영역을 획득한다.

[0011] 상기 패킷 동기화부를 통해 지속적인 패킷 동기화를 위한 딥 러닝 기반 파일럿 심볼 검출 프로세스를 수행하여 각 패킷의 시작점을 식별하고 동기화하는 단계는 3개의 파일럿 심볼(R, G, B) 및 8개의 데이터 심볼(0-7)을 갖는 총 11개의 컬러 바코드의 클래스는 디스플레이 버퍼와 셔터 카메라 사이의 비동기를 감소시키기 위해 각 클래스에 대하여 상기 감지된 컬러 바코드의 패턴에 대한 추가 레이블링을 수행한다.

[0012] 상기 패킷 동기화부를 통해 지속적인 패킷 동기화를 위한 딥 러닝 기반 파일럿 심볼 검출 프로세스를 수행하여 각 패킷의 시작점을 식별하고 동기화하는 단계는 패킷 동기화를 위해 상기 감지된 컬러 바코드의 패턴의 입력으로부터 파일럿 심볼(R, G, B)의 패턴을 개별 클래스로 지정하고 데이터 심볼에 나타나는 패턴을 하나의 클래스로 묶어 총 4개의 동기화 클래스를 분류하도록 훈련함으로써 파일럿 심볼과 데이터 심볼을 식별하고, 파일럿 심볼의 종류를 인식한다.

[0013] 상기 동기화가 완료되면, 각 패킷 내의 데이터에 대해 딥 러닝을 사용하여 데이터 심볼 검출부를 통해 연속적으

로 디코딩하는 단계는 총 11개의 컬러 바코드의 클래스, 총 4개의 동기화 클래스 및 복수의 이미지 캡처 각수에 따라 데이터 심볼 검출을 위한 딥 러닝 훈련 데이터 셋을 구축하고, 확대, 축소, 회전, 위치 변환, 수직 반전을 포함하는 데이터 증강 기법을 이용하여 합성 데이터를 생성하여 심볼 검출을 위한 딥 러닝 수행한다.

[0014] 또 다른 일 측면에 있어서, 본 발명에서 제안하는 딥러닝을 이용한 보색 바코드 기반 광카메라 통신 시스템은 송신부 및 수신부를 포함하고, 상기 송신부는 입력 비트 스트림의 이진 데이터를 컬러 바코드로 인코딩하여 지정된 색상 값에 매핑하고 데이터 패킷으로 구성하는 인코딩부, 상기 인코딩된 컬러 바코드에 대하여 모든 심볼이 보색 쌍으로 구성되는 연속적인 데이터 패킷을 생성하는 데이터 패킷 생성부 및 상기 생성된 연속적인 데이터 패킷을 송신하는 디스플레이 버퍼를 포함하고, 상기 수신부는 상기 송신되는 데이터 패킷에 대하여 셔터 카메라를 이용하여 표시 영역을 캡처하고, 캡처된 이미지를 저장하는 카메라 프레임 버퍼, 상기 저장된 이미지에 대해 딥 러닝 모델을 사용하여 컬러 바코드 영역이 감지하고, 감지된 컬러 바코드의 이미지를 이미지 목록에 저장하는 컬러 바코드 감지부, 지속적인 패킷 동기화를 위한 딥 러닝 기반 파일럿 심볼 검출 프로세스를 수행하여 각 패킷의 시작점을 식별하고 동기화하는 패킷 동기화부 및 상기 동기화가 완료되면, 각 패킷 내의 데이터에 대해 딥 러닝을 사용하여 연속적으로 디코딩을 수행하는 데이터 심볼 검출부를 포함한다.

발명의 효과

[0015] 본 발명의 실시예들에 따르면 감지 불가능한 D2C 통신 시스템인 딥러닝을 이용한 보색 바코드 기반 광카메라 통신(Deep learning-driven Complementary Color Barcode-based Optical Camera Communications; DeepCCB-OCC) 방법 및 시스템을 통해 안정적인 데이터 전송을 지원할 수 있다. D2C 링크에서 끊김 없는 바코드 이미지 영역을 감지하기 위해 YOLO(You Only Look Once) 기반 바코드 영역 감지 모델이 구현되고, CNN(Convolutional Neural Network) 모델은 컬러 바코드에 내장된 데이터를 추출하여 패킷 동기화 및 데이터 디코딩을 수행함으로써 DeepCCB-OCC는 고유한 심층 신경망(DNN) 속성의 일반화 성능으로 인해 안정적인 통신을 제공하여 기존 방식의 비신뢰성의 근본 원인을 해결할 수 있다.

도면의 간단한 설명

[0016] 도 1은 본 발명의 일 실시예에 따른 DeepCCB-OCC 시스템의 구성을 나타내는 도면이다.
 도 2는 본 발명의 일 실시예에 따른 DeepCCB-OCC 시스템의 동작 방법을 설명하기 위한 흐름도이다.
 도 3은 본 발명의 일 실시예에 따른 보색 컬러 바코드 기반 OCC를 위한 색상값을 나타내는 도면이다.
 도 4는 본 발명의 일 실시예에 따른 CCB-OCC의 전송 패킷 구조를 나타내는 도면이다.
 도 5는 본 발명의 일 실시예에 따른 YOLO 기반 컬러 바코드 감지 아키텍처를 나타내는 도면이다.
 도 6은 본 발명의 일 실시예에 따른 컬러 바코드 영역 추출 과정을 나타내는 도면이다.
 도 7은 본 발명의 일 실시예에 따른 컬러 바코드의 4가지 패턴을 나타내는 도면이다.
 도 8은 본 발명의 일 실시예에 따른 수신된 컬러 바코드 심볼 001에 대한 4가지 패턴을 나타내는 도면이다.
 도 9는 본 발명의 일 실시예에 따른 데이터 증강 기술을 사용한 합성 데이터를 나타내는 도면이다.
 도 10은 본 발명의 일 실시예에 따른 패킷 동기화 및 데이터 인코딩을 위한 CNN 모델 구조를 나타내는 도면이다.
 도 11은 본 발명의 일 실시예에 따른 DeepCCB-OCC의 전체 알고리즘을 나타내는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0017] D2C(Display-to-camera) 통신은 일반적인 시청 중에 카메라가 장착된 장치에 부가 정보를 제공하는 차세대 비디오의 핵심 방법으로 부상했다. 본 발명에서는 D2C 링크에서 감지할 수 없는 전송과 안정적인 통신을 위해 구축된 다중 심층 신경망을 사용하는 D2C 시스템인 딥 러닝 기반 보색 컬러 바코드 기반 광학 카메라 통신(Deep learning-driven Complementary Color Barcode-based Optical Camera Communications; DeepCCB-OCC)을 제안한다. DeepCCB-OCC는 YOLO(You Only Look Once) 모델을 활용하여 전자 디스플레이의 컬러 바코드 영역을 완벽하게 감지한다. 수신된 이미지에서 전송된 컬러 바코드 기호를 식별하기 위해 카메라와 디스플레이 간의 동기화 지터로 인해 발생하는 다양한 컬러 바코드 패턴을 정의한다. 그런 다음 DeepCCB-OCC는 CNN(Convolutional Neural Network) 모델을 통합하여 다양한 D2C 환경에 관계없이 전송 패킷에서 파일럿 및 데이터 기호를 정확하

게 감지한다. 상업용 모니터와 스마트폰을 사용한 실험에서 DeepCCB-OCC가 D2C 링크의 다양한 거리와 각도에서 기존 CCB-OCC 시스템을 능가한다는 것을 확인했다. 실험 결과 디스플레이와 카메라 사이의 거리 90cm에서 정렬 각도가 20도일 때 제안 방식이 약 79.1bps를 달성하여 기존 방식에 비해 14.1%의 성능 향상을 보였다. 이하, 본 발명의 실시 예를 첨부된 도면을 참조하여 상세하게 설명한다.

- [0019] 도 1은 본 발명의 일 실시예에 따른 DeepCCB-OCC 시스템의 구성을 나타내는 도면이다.
- [0020] 제안하는 DeepCCB-OCC 시스템은 송신부(110) 및 수신부(120)를 포함한다.
- [0021] DeepCCB-OCC 시스템의 송신부(110)는 인코딩부(112), 데이터 패킷 생성부(113) 및 디스플레이 버퍼(114)를 포함한다.
- [0022] 본 발명의 실시예에 따른 인코딩부(112)는 입력 비트 스트림의 이진 데이터를 컬러 바코드로 인코딩하여 지정된 색상 값에 매핑하고 데이터 패킷으로 구성한다.
- [0023] 송신부(110)에서, 이진 데이터의 입력 비트 스트림(111)은 인코딩부(112)를 통해 컬러 바코드로 인코딩된다.
- [0024] 본 발명의 실시예에 따른 데이터 패킷 생성부(113)는 상기 인코딩된 컬러 바코드에 대하여 모든 심볼이 보색 쌍으로 구성되는 연속적인 데이터 패킷을 생성한다.
- [0025] 데이터가 색상 형태로 인코딩되기 위해서는 사전 설계된 컬러 콘스텔레이션(constellation) 다이어그램을 사용한 색상 매핑이 필요하다. 그런 다음 파일럿 및 데이터 심볼은 지정된 색상 값에 매핑되고 데이터 패킷으로 구성된다. 패킷 내의 모든 심볼은 보색 쌍으로 구성되며, 디스플레이는 지정된 컬러 바코드 영역에서 보색 쌍으로 구성된 심볼을 연속적인 이미지 프레임에 표시한다.
- [0026] 본 발명의 실시예에 따른 디스플레이 버퍼(114)는 상기 생성된 연속적인 데이터 패킷을 송신한다.
- [0027] DeepCCB-OCC 시스템의 수신부(120)는 카메라 프레임 버퍼(121), 컬러 바코드 감지부(122), 패킷 동기화부(123) 및 데이터 심볼 검출부(124)를 포함한다.
- [0028] 본 발명의 실시예에 따른 카메라 프레임 버퍼(121)는 상기 송신되는 데이터 패킷에 대하여 서터 카메라를 이용하여 표시 영역을 캡처하고, 캡처된 이미지를 저장한다. 이때 사람의 눈에는 컬러 바코드 영역이 흰색으로 나타나지만, 컬러 바코드 영역에서는 지속적인 색상 변화가 관찰된다.
- [0029] 본 발명의 실시예에 따른 컬러 바코드 감지부(122)는 상기 저장된 이미지에 대해 딥 러닝 모델을 사용하여 컬러 바코드 영역이 감지하고, 감지된 컬러 바코드의 이미지를 이미지 목록에 저장한다.
- [0030] 본 발명의 실시예에 따른 패킷 동기화부(123)는 지속적인 패킷 동기화를 위한 딥 러닝 기반 파일럿 심볼 검출 프로세스를 수행하여 각 패킷의 시작점을 식별하고 동기화한다.
- [0031] 본 발명의 실시예에 따른 데이터 심볼 검출부(124)는 상기 동기화가 완료되면, 각 패킷 내의 데이터에 대해 딥 러닝을 사용하여 연속적으로 디코딩을 수행한다.
- [0033] 도 2는 본 발명의 일 실시예에 따른 DeepCCB-OCC 시스템의 동작 방법을 설명하기 위한 흐름도이다.
- [0034] 제안하는 DeepCCB-OCC 시스템의 광카메라 통신 방법은 광카메라 통신을 위한 송신부의 인코딩부를 통해 입력 비트 스트림의 이진 데이터를 컬러 바코드로 인코딩하여 지정된 색상 값에 매핑하고 데이터 패킷으로 구성하는 단계(210), 상기 인코딩된 컬러 바코드에 대하여 데이터 패킷 생성부를 통해 모든 심볼이 보색 쌍으로 구성되는 연속적인 데이터 패킷을 생성하고, 디스플레이 버퍼를 통해 송신하는 단계(220), 광카메라 통신을 위한 수신부를 통해 상기 송신되는 데이터 패킷에 대하여 서터 카메라를 이용하여 표시 영역을 캡처하고, 캡처된 이미지는 카메라 프레임 버퍼에 저장하는 단계(230), 상기 저장된 이미지에 대해 컬러 바코드 감지부를 통해 딥 러닝 모델을 사용하여 컬러 바코드 영역이 감지되면 컬러 바코드 이미지를 이미지 목록에 저장하는 단계(240), 패킷 동기화부를 통해 지속적인 패킷 동기화를 위한 딥 러닝 기반 파일럿 심볼 검출 프로세스를 수행하여 각 패킷의 시작점을 식별하고 동기화하는 단계(250) 및 상기 동기화가 완료되면, 각 패킷 내의 데이터에 대해 딥 러닝을 사용하여 데이터 심볼 검출부를 통해 연속적으로 디코딩하는 단계(260)를 포함한다.
- [0035] 단계(210)에서, 광카메라 통신을 위한 송신부의 인코딩부를 통해 입력 비트 스트림의 이진 데이터를 컬러 바코드로 인코딩하여 지정된 색상 값에 매핑하고 데이터 패킷으로 구성한다. 이진 데이터의 입력 비트 스트림은 인코딩부를 통해 컬러 바코드로 인코딩된다.
- [0036] 단계(220)에서, 상기 인코딩된 컬러 바코드에 대하여 데이터 패킷 생성부를 통해 모든 심볼이 보색 쌍으로 구성

되는 연속적인 데이터 패킷을 생성하고, 디스플레이 버퍼를 통해 송신한다.

- [0037] 데이터가 색상 형태로 인코딩되기 위해서는 사전 설계된 컬러 콘스텔레이션(constellation) 다이어그램을 사용한 색상 매핑이 필요하다. 그런 다음 파일럿 및 데이터 심볼은 지정된 색상 값에 매핑되고 데이터 패킷으로 구성된다. 패킷 내의 모든 심볼은 보색 쌍으로 구성되며, 디스플레이는 지정된 컬러 바코드 영역에서 보색 쌍으로 구성된 심볼을 연속적인 이미지 프레임에 표시한다.
- [0038] 단계(230)에서, 광카메라 통신을 위한 수신부를 통해 상기 송신되는 데이터 패킷에 대하여 셔터 카메라를 이용하여 표시 영역을 캡처하고, 캡처된 이미지는 카메라 프레임 버퍼에 저장한다. 이때 사람의 눈에는 컬러 바코드 영역이 흰색으로 나타나지만, 컬러 바코드 영역에서는 지속적인 색상 변화가 관찰된다.
- [0039] 단계(240)에서, 상기 저장된 이미지에 대해 컬러 바코드 감지부를 통해 딥 러닝 모델을 사용하여 컬러 바코드 영역이 감지되면 컬러 바코드 이미지를 이미지 목록에 저장한다.
- [0040] 단계(250)에서, 패킷 동기화부를 통해 지속적인 패킷 동기화를 위한 딥 러닝 기반 파일럿 심볼 검출 프로세스를 수행하여 각 패킷의 시작점을 식별하고 동기화한다.
- [0041] 단계(260)에서, 상기 동기화가 완료되면, 각 패킷 내의 데이터에 대해 딥 러닝을 사용하여 데이터 심볼 검출부를 통해 연속적으로 디코딩을 수행한다.
- [0043] 도 3은 본 발명의 일 실시예에 따른 보색 컬러 바코드 기반 OCC를 위한 색상값을 나타내는 도면이다.
- [0044] 본 발명의 실시예에 따른 DeepCCB-OCC 시스템을 사용하여 정보를 전송하려면 전송 심볼과 색상 값을 일치시켜야 한다. DeepCCB-OC 시스템의 보색 관계는 도 3에 표시된 색상 원을 사용하여 구성되었다. 원색과 보색은 색상 원에서 서로 180도 떨어져 있다. 원색(빨강, 녹색, 파랑)의 경우 보색은 청록색, 노랑색, 주황색이다. 원색과 보색을 더하면 R, G, B 채널 값이 255가 되어 흰색이 된다. 디스플레이의 빠른 리프레시 속도를 통해 원색과 보색이 연속적으로 나타날 때 인간의 눈은 두 색의 조합을 흰색으로 본다. 이를 통해 인간에게 방해가 되지 않는 추가 정보가 디스플레이를 통해 전송될 수 있다. 도 3에서 볼 수 있듯이, 8개의 심볼은 색상 원을 구성하며, 각 심볼은 3비트로 구성된다. 그레이 코딩은 디코딩 프로세스에서 비트 오류율을 최소화하기 위해 사용된다. 각각의 심볼은 빨강(R), 초록(G), 파랑(B) 채널의 강도값으로 표현되며, CCB-OCC에서 사용되는 8개의 심볼에 대한 각 채널의 색상 값은 도 3와 같다.
- [0045] 본 발명의 실시예에 따르면, 디스플레이를 전송할 때, 컬러 바코드는 원색과 보색이 쌍을 이루도록 연속적으로 구성되어야 한다. R, G, B 인코드 심볼이 전송되면 실제 데이터 스트림은 RR' GG' BB'이며, 여기서 R, G, B는 인코드 심볼을 나타내고 R', G', B'는 보색을 나타낸다. 수신기는 롤링 셔터 카메라를 사용하여 디스플레이를 촬영하는 동안 캡처된 이미지를 사용한다. 이론적으로 디스플레이 출력의 절반 속도로 이미지를 캡처할 때 원색과 보색의 쌍으로 구성된 컬러 바코드가 하나의 캡처된 이미지로 나타난다. 예를 들어, 도 1에 보여진 바와 같이, 송신부의 화면 리프레시 비율이 60 Hz이고, 수신부가 30 fps로 이를 캡처하는 경우, 송신된 6개의 프레임(R, R', G, G', B, B')은 수신기에 RR', GG', BB'의 3개의 수신 이미지로 나타나지만, 사람의 눈에는 흰색 막대로 인식된다.
- [0047] 도 4는 본 발명의 일 실시예에 따른 CCB-OCC의 전송 패킷 구조를 나타내는 도면이다.
- [0048] 본 발명의 실시예에 따른 데이터 인코딩된 컬러 바코드 심볼은 도 4와 같이 패킷 형태로 전송된다. CCB-OCC에서 사용되는 전송 패킷은 파일럿 심볼과 데이터 심볼로 구성되며, 각 심볼은 보색과 쌍으로 구성된다. 동기화 및 채널 추정에 사용되는 파일럿 심볼은 RR 'GG'BB' 형식의 6개 이미지 프레임으로 설계되었다. 파일럿 심볼을 따라 데이터는 인코딩된 색상 심볼과 보색 쌍으로 구성된 프레임 쌍으로 설계된다. 데이터 인코딩된 이미지 프레임은 롤링 셔터 카메라에 의해 캡처되며, 캡처된 이미지 내의 컬러 바코드 영역은 이미지 처리 단계에서 YOLO 객체 검출기를 통해 검출된다. 그 후, 동기화 및 디코딩은 패킷의 심볼에 대해 훈련된 CNN 모델을 사용하여 수행된다.
- [0050] 도 5는 본 발명의 일 실시예에 따른 YOLO 기반 컬러 바코드 감지 아키텍처를 나타내는 도면이다.
- [0051] 본 발명의 실시예에 따른 D2C 시스템에서 데이터 디코딩 프로세스를 수행하려면 모니터에 캡처된 영상 내에서 컬러 바코드 영역만 추출해야 한다. 종래기술에서, 차동 이미지에서 추출된 컬러 바코드 영역은 주변 빛의 영향과 영상 캡처 지터(jitter) 문제로 인해 부정확한 결과를 초래했다. 이 문제는 디스플레이와 카메라 사이의 정렬 불량으로 인해 악화된다. 따라서 본 발명에서는 실시간 객체 감지 모델인 YOLO를 이용한 컬러 바코드 영역

추출 방법을 제안한다.

- [0052] 본 발명에서는 다양한 YOLO 버전 중 컬러 바코드 영역 추출을 위해 YOLOv4를 적용하였다. 도 5와 같이 YOLOv4 모델은 입력 이미지(511)를 입력 받는 CSPDarknet53(510)을 백본 구조로 하고, YOLOv3 모델은 거의 그대로 사용하며, 넥(neck) 구조(520)는 공간 피라미드 풀링(Spatial Pyramid Pooling; SPP)과 경로 집합 네트워크(Path Aggregation Network; PAN)를 사용한다. 또한 기존 YOLOv3 모델을 개선하기 위해 YOLOv4(530)는 BoF(Bag of Freebies) 및 BoS(Bag of Specials)라는 기법을 사용하여 컬러 바코드(531)를 감지한다. BoF는 훈련 비용을 증가시켜 학습 정확도를 향상시키는 데 사용된다. 대표적인 예로 랜덤 삭제(Random Erase)를 사용하여 이미지의 일부 영역에 노이즈를 추가하고, 컷 아웃(CutOut)을 사용하여 일부 영역을 제거하며, 믹스 업(MixUp)을 사용하여 레이블이 다른 두 이미지를 혼합하고, 컷 믹스(CutMix)를 사용하여 레이블이 있는 두 이미지를 잘라내어 붙여 넣는다. BoS는 예측 정확도를 높이기 위한 방법으로, 수용 영역 확대, 주의 메커니즘 도입, 특징 통합 기능 강화, 예측 결과에 대한 경계 상자 선택 등 특정 특징을 강화하기 위해 다양한 방법이 사용된다.
- [0053] YOLOv4 기반의 컬러 바코드 영역 추출을 위한 입력은 컬러 바코드를 전송하는 디스플레이 장치에서 캡처한 이미지이다. 입력 이미지(511)가 CSPDarknet53(510)의 입력 크기인 512x512픽셀로 변환된다. 여기서, 변환된 이미지에서 추출될 컬러 바코드 영역이 크게 감소한다. 그러나 YOLOv4의 백본으로 사용되는 CSPDarknet53(510)은 수용 필드가 크고 파라미터가 많아 작은 물체에도 우수한 검출 성능을 보인다. 또한 넥 구조(520)에서 모델의 작동 속도에 영향을 미치지 않고 수용 영역을 증가시키는 SPP 블록과 다양한 백본 레벨의 정보를 수신하는 PAN을 통해 다양한 크기의 컬러 바코드를 성공적으로 추출할 수 있었다. 이를 통해 디스플레이와 카메라 사이의 다양한 거리에서 컬러 바코드 영역(531)을 강력하게 추출할 수 있다.
- [0054] 수신되는 컬러 바코드의 이상적인 형태는 데이터 인코딩된 원색과 보색이 컬러 바코드 영역 내에서 원활하게 연결되는 형태일 것이다. 그러나 실제 추출된 컬러 바코드는 직사각형이 아닌 사다리꼴로 나타나는 것이 일반적이다. 이는 디스플레이와 카메라 사이의 모양과 정렬 각도로 인해 캡처된 이미지에 큰 왜곡이 발생하기 때문일 수 있다. 이는 대부분의 캡처 결과에서 발생하며, 디스플레이와 카메라 사이의 거리는 왜곡을 더욱 크게 만들어 컬러 바코드의 크기에도 영향을 미친다.
- [0056] 도 6은 본 발명의 일 실시예에 따른 컬러 바코드 영역 추출 과정을 나타내는 도면이다.
- [0057] YOLO를 사용하여 추출된 영역에는 컬러 바코드 외에도 주변 노이즈가 포함되어 동기화 및 디코딩 시 성능 저하가 발생한다. 따라서 노이즈를 제거하는 과정이 필요하지만, 단순히 YOLO를 이용하여 추출된 패킷 영역이 연속 프레임에서 일정하지 않기 때문에 YOLO에 의해 추출된 영역을 줄이는 것은 효과적이지 않다. 효과적인 노이즈 제거를 위해 도 6과 같이 전체 공정에서 차동 이미지, 형태학적 연산 등의 방법을 수행한다.
- [0058] 도 6(a)에 보여진 바와 같이, 컬러 바코드 영역을 통해 패킷 정보가 전송되는 디스플레이를 캡처하고, 수신된 디스플레이 이미지로부터 컬러 바코드 영역의 대략적인 영역을 추출하기 위해 컬러 바코드 영역의 주석을 포함하는 데이터를 이용하여 YOLO 모델을 훈련한다. YOLO를 통해 추출된 데이터는 해당 영역의 높이와 폭에 대한 컬러 바코드의 대략적인 직사각형 영역의 x축 및 y축 좌표를 포함한다.
- [0059] 도 6(b)에 표시된 바와 같이, 추출된 컬러 바코드 영역은 들어오는 각 비디오 프레임마다 다르며 주변 노이즈를 포함한다. 디코딩 성능에 영향을 미치는 주변 노이즈를 최소화하기 위해 주변 영역 없이 컬러 바코드 영역만 포함된 이미지를 획득해야 한다. 이를 위해 컬러 바코드 영역에서 추출된 좌표가 연속 영상 프레임에 저장된다. 저장된 좌표에서 x축 및 y축 좌표의 최소값과 높이 및 너비의 최대값을 찾는다. 이 과정을 통해 컬러 바코드 정보의 큰 손실 없이 가능한 가장 작은 크기의 영역을 얻을 수 있지만, 여전히 약간의 주변 노이즈를 포함하고 있다. 캡처된 이미지에서 노이즈를 제거하는 과정은 데이터를 전송하기 위해 컬러 바코드가 지속적으로 변경되지만 주변 영역은 변경되지 않는다는 사실을 사용한다.
- [0060] 도 6(c)는 차동 및 형태학적 연산에 의해 얻어진 컬러 바코드 마스크 이미지를 나타낸다. 마스크 이미지를 생성하려면 컬러 바코드 이미지가 2진수로 표시된다. 여기서 특정 그레이 스케일 임계값 20을 초과하는 값은 값이 255인 흰색으로 표시되고 주변 영역은 0으로 설정된다(즉, 검은색으로 표시됨). 연속되는 3개 이미지의 차동 이미지를 획득하여 이전 차동 이미지에 추가함으로써, 배경 제거 컬러 바코드 마스크를 생성할 수 있다. 컬러 바코드 영역 외부의 노이즈가 남아 있기 때문에 차동 이미지를 사용한 결과에서도 이를 제거하기 위해 침식 및 확장 등 여러 형태학적 연산이 사용된다. 그 결과, 노이즈가 없는 컬러 바코드 영역의 추출이 가능하다.
- [0061] 도 6(d)에 보여지는 바와 같이, 최종적으로 획득된 컬러 바코드 마스크 이미지에 YOLO를 통해 추출된 이미지를 곱하면 노이즈가 제거된 순수한 컬러 바코드의 영역을 추출할 수 있다.

- [0063] 도 7은 본 발명의 일 실시예에 따른 컬러 바코드의 4가지 패턴을 나타내는 도면이다.
- [0064] CCB-OCC에서 컬러 바코드는 송신 장치와 수신 장치 간의 동기화 불일치 및 정렬 문제로 인해 동일한 전송 신호로부터 수신된 이미지에서 다양한 패턴으로 나타난다. 종래기술에서는 바코드의 색상을 식별하기 위해 캡처된 이미지에 나타날 수 있는 제한된 수의 패턴에 대해 히스토그램 분석 접근법을 사용했다. 컬러 바코드의 RGB 값은 각 채널에 대한 히스토그램의 최대값과 최소값을 구하여 추출할 수 있었으며, 이를 이용하여 파일럿 심볼과 데이터 심볼을 결정하였다. 이 방법은 다양한 히스토그램 패턴에 대해 RGB 값을 정확하게 검출하기 어렵다는 한계가 있으며, 최악의 경우 RGB 값을 추출하기 위한 휴리스틱 접근법이 필요했다. 특히, 컬러 바코드를 구성하는 RGB 색상 값의 히스토그램 분포는 캡처 환경이 변화하면 크게 변화하여 부정확한 색상 값 추출이 발생한다. 휴리스틱 히스토그램 분석에 의존하는 CCB-OCC 체계와 달리, 본 발명에서는 CNN 모델(객체 분류 분야에서 강력한 성능을 보여주는 잘 알려진 기술)을 사용하여 심볼 검출 성능의 신뢰성을 향상시킬 수 있는 방법을 제안한다.
- [0065] CNN 모델을 훈련시키려면 컬러 바코드의 주석이 필요하다. 제안된 DeepCCB-OCC의 컬러 바코드 클래스는 3개의 파일럿 심볼(R, G, B)과 8개의 데이터 심볼(0-7)로 구성되어 있으며, 총 11개의 클래스로 구성되어 있다. 수신단에서 추출된 컬러 바코드는 데이터로 인코딩된 원색과 보색이 정확히 반으로 줄면서 나타나는 것이 아니라 송신 장치 간의 비동기적인 문제로 인해 다양한 이동 패턴으로 나타나므로 추가적인 레이블링이 필요하다. CNN 모델을 교육하기 전에 각 심볼 클래스에 대한 컬러 바코드 패턴의 레이블링을 수행해야 한다.
- [0066] 본 발명에서는 도 7과 같이 수신된 컬러 바코드의 4가지 이동 패턴을 정의한다. 상기 제1 패턴은 원색과 보색의 패턴이 반감되는 이상적인 형태의 컬러 바코드이다. 여기서, 수신된 컬러 바코드의 이동 패턴은 파일럿 심볼 G로 표시된다. 이상적인 형태인 첫 번째 패턴은 도 7(a)에서 GG'로 표시된다. 도 7(b)에 보여지는 제2 패턴은 원색이 컬러 바코드의 중앙에 위치하고, 좌측에는 이전 컬러 바코드의 보색이 표시되고, 우측에는 원색의 보색이 표시된다. 결과적으로, 그것은 R'GG'로 나타난다. 도 7(c)는 좌측에 이전 컬러 바코드의 보색이 포착되고, 우측에 원색의 컬러 바코드가 표시된다: 도 7(c)에 R'G가 보여진다. 도 7(d)는 총 네 개의 패턴을 가지고 있다. 각 패턴은 3개의 파일럿 심볼과 8개의 데이터 심볼로 구성되어 총 44개의 클래스에서 데이터를 수신할 수 있다.
- [0067] 상기와 같이 연속적인 색상 패턴은 4가지로 나눌 수 있으나, 송신단과 수신단의 동기 타이밍에 따라 컬러 바코드 패턴이 달라질 수 있다. 이러한 상황에서 단일 딥 러닝 모델을 사용하여 44개 클래스를 분류하는 접근 방식은 색상 패턴 분류 정확도를 제한하며, 궁극적으로 DeepCCB-OCC 시스템의 통신 성능에 악영향을 미칠 수 있다. 따라서 다양한 색상 패턴에 대한 DeepCCB-OCC 시스템의 일반화 성능을 유지하면서 분류 성능을 극대화하기 위해 파일럿 심볼 분류 및 데이터 심볼 분류를 위한 CNN 모델을 독자적으로 개발하였다.
- [0069] 도 8은 본 발명의 일 실시예에 따른 수신된 컬러 바코드 심볼 001에 대한 4가지 패턴을 나타내는 도면이다.
- [0070] DeepCCB-OCC 시스템에서의 동기화를 위해서는 파일럿 심볼 분류 모델이 사용되며, 여러 컬러 바코드 패턴의 입력으로부터 파일럿 심볼만을 정확하게 식별해야 한다. 이 모델은 파일럿 심볼과 데이터 심볼을 구별할 수 있어야 하며 파일럿 심볼의 종류를 인식할 수 있어야 한다. 이를 위해 R, G, B 파일럿 심볼의 패턴을 개별 클래스로 지정하고 데이터 심볼에 나타나는 다양한 패턴을 하나의 클래스로 묶어 훈련한다. 따라서 파일럿 심볼 분류 모델은 총 4개의 클래스를 분류하도록 훈련된다.
- [0071] DeepCCB-OCC에서 컬러 바코드는 송신 장치와 수신 장치 간의 동기화 불일치 및 정렬 문제로 인해 동일한 전송 신호로부터 수신된 이미지에서 다양한 패턴으로 나타난다. 실제 CCB-OCC 시스템에서 수신된 컬러 바코드(810)는 도 8과 같이 특정 심볼에 대해 4가지 패턴(811, 812, 813, 814)으로 나타난다. 앞서 설명된 바와 같이, 파일럿 심볼 및 데이터 심볼을 포함하여 11개의 데이터 클래스(R, G, B 및 0-7)가 컬러 바코드로 인코딩된다. 또한, 5개의 서로 다른 각도에서 촬영된 이미지를 사용하였기 때문에, 수집된 컬러 바코드의 종류의 총합은 이미지 캡처 각도의 수(5) × 컬러 바코드 패턴의 수(4) × 인코딩된 데이터의 종류(11)이므로 220개의 종류가 있다. CNN 모델 훈련을 위한 데이터셋을 구축하기 위해, 각 컬러 바코드 유형에 대해 20개의 이미지 총 4400개를 수집했다.
- [0073] 도 9는 본 발명의 일 실시예에 따른 데이터 증강 기술을 사용한 합성 데이터를 나타내는 도면이다.
- [0074] DeepCCB-OCC 기법을 실제로 사용할 경우 디스플레이와 카메라 사이의 다양한 거리와 각도에 따라 컬러 바코드의 크기와 형태가 다양한 형태로 나타날 수 있다. 모델 일반화를 개선하기 위해 심층 신경망 훈련의 표준 구성 요소가 된 데이터 증강이 사용된다. 도 9는 DeepCCB-OCC 모델을 훈련하는 데 사용되는 합성 데이터를 보여준다. 이와 같이 확대/축소(도 9(a)), 회전(도 9(b)), 위치 변환(도 9(c)), 수직 반전(도 9(d)) 등의 데이터 증강 기법을 이용하여 합성 데이터를 생성하였다. 합성 데이터의 다양성을 지원하기 위해 컬러 바코드 유형별로 80개의

합성 컬러 바코드를 생성하여 22,000개의 데이터셋을 생성하였다. 22,000개의 컬러 바코드 이미지 중 R, G, B 파일럿 신호가 인코딩된 6000개가 파일럿 심볼 분류를 위해 CNN 모델을 훈련하는 데 사용되었다. 0-7개의 데이터 심볼이 인코딩된 나머지 16,000개의 컬러 바코드는 데이터 심볼 분류를 위한 CNN 모델을 훈련시키는 데 사용되었다.

[0076] 도 10은 본 발명의 일 실시예에 따른 패킷 동기화 및 데이터 인코딩을 위한 CNN 모델 구조를 나타내는 도면이다.

[0077] CNN 모델의 구조는 인코딩된 정보에 관계없이 동일했으며, 두 구조는 동일하게 구성되었다. 모델에 대한 입력(1011)으로 3개의 2D 컨볼루션 계층(1010), 2개의 최대 풀링 계층(1020, 2030), 완전 연결 계층, 드롭아웃 계층(1040)으로 구성된 100×900의 RGB 3채널 이미지가 사용되었다. 각 분류 모델에 대한 훈련 데이터가 다르게 구성되었기 때문에, 각 모델은 독립적으로 훈련되었다.

[0078] 도 10과 같이 파일럿 심볼 분류 모델은 4개의 클래스가 있으며, 데이터 심볼 분류 모델은 8개의 클래스가 있다(1050). 파일럿 심볼 검출을 위한 CNN 모델을 사용하여 RR 'GG'BB'의 연속 컬러 바코드 패턴을 얻을 때, 데이터 검출을 위한 CNN 모델은 패킷의 데이터 심볼 길이만큼 사용되며, 전송된 신호는 순차적으로 디코딩된다.

[0080] 도 11은 본 발명의 일 실시예에 따른 DeepCCB-OCC의 전체 알고리즘을 나타내는 도면이다.

[0081] DeepCCB-OCC를 사용하여 데이터를 디코딩하는 전체 프로세스는 도 11에 설명되어 있다.

[0083] 이상에서 설명된 장치는 하드웨어 구성요소, 소프트웨어 구성요소, 및/또는 하드웨어 구성요소 및 소프트웨어 구성요소의 조합으로 구현될 수 있다. 예를 들어, 실시예들에서 설명된 장치 및 구성요소는, 예를 들어, 프로세서, 콘트롤러, ALU(arithmetic logic unit), 디지털 신호 프로세서(digital signal processor), 마이크로컴퓨터, FPGA(field programmable gate array), PLU(programmable logic unit), 마이크로프로세서, 또는 명령(instruction)을 실행하고 응답할 수 있는 다른 어떠한 장치와 같이, 하나 이상의 범용 컴퓨터 또는 특수 목적 컴퓨터를 이용하여 구현될 수 있다. 처리 장치는 운영 체제(OS) 및 상기 운영 체제 상에서 수행되는 하나 이상의 소프트웨어 애플리케이션을 수행할 수 있다. 또한, 처리 장치는 소프트웨어의 실행에 응답하여, 데이터를 접근, 저장, 조작, 처리 및 생성할 수도 있다. 이해의 편의를 위하여, 처리 장치는 하나가 사용되는 것으로 설명된 경우도 있지만, 해당 기술분야에서 통상의 지식을 가진 자는, 처리 장치가 복수 개의 처리 요소(processing element) 및/또는 복수 유형의 처리 요소를 포함할 수 있음을 알 수 있다. 예를 들어, 처리 장치는 복수 개의 프로세서 또는 하나의 프로세서 및 하나의 콘트롤러를 포함할 수 있다. 또한, 병렬 프로세서(parallel processor)와 같은, 다른 처리 구성(processing configuration)도 가능하다.

[0084] 소프트웨어는 컴퓨터 프로그램(computer program), 코드(code), 명령(instruction), 또는 이들 중 하나 이상의 조합을 포함할 수 있으며, 원하는 대로 동작하도록 처리 장치를 구성하거나 독립적으로 또는 결합적으로(collectively) 처리 장치를 명령할 수 있다. 소프트웨어 및/또는 데이터는, 처리 장치에 의하여 해석되거나 처리 장치에 명령 또는 데이터를 제공하기 위하여, 어떤 유형의 기계, 구성요소(component), 물리적 장치, 가상 장치(virtual equipment), 컴퓨터 저장 매체 또는 장치에 구체화(embodiment)될 수 있다. 소프트웨어는 네트워크로 연결된 컴퓨터 시스템 상에 분산되어서, 분산된 방법으로 저장되거나 실행될 수도 있다. 소프트웨어 및 데이터는 하나 이상의 컴퓨터 판독 가능 기록 매체에 저장될 수 있다.

[0085] 실시예에 따른 방법은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 실시예를 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

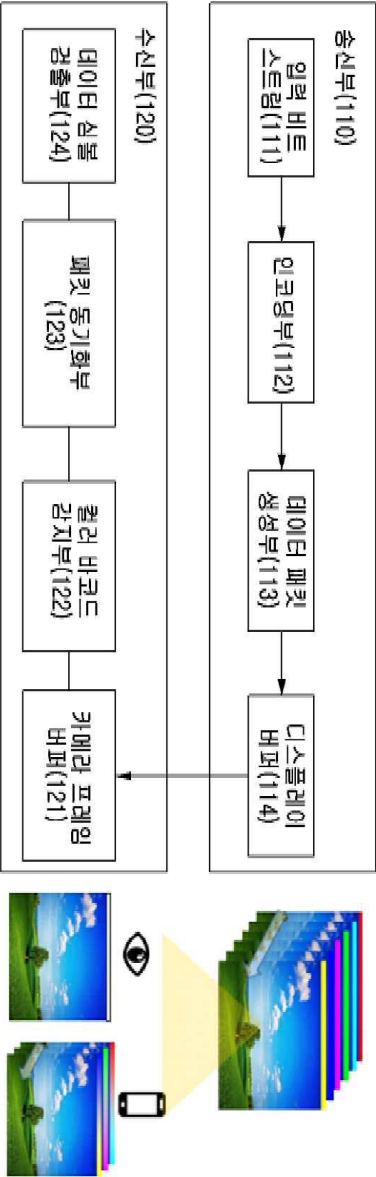
[0086] 이상과 같이 실시예들이 비록 한정된 실시예와 도면에 의해 설명되었으나, 해당 기술분야에서 통상의 지식을 가진 자라면 상기의 기재로부터 다양한 수정 및 변형이 가능하다. 예를 들어, 설명된 기술들이 설명된 방법과 다른 순서로 수행되거나, 및/또는 설명된 시스템, 구조, 장치, 회로 등의 구성요소들이 설명된 방법과 다른 형

태로 결합 또는 조합되거나, 다른 구성요소 또는 균등물에 의하여 대치되거나 치환되더라도 적절한 결과가 달성될 수 있다.

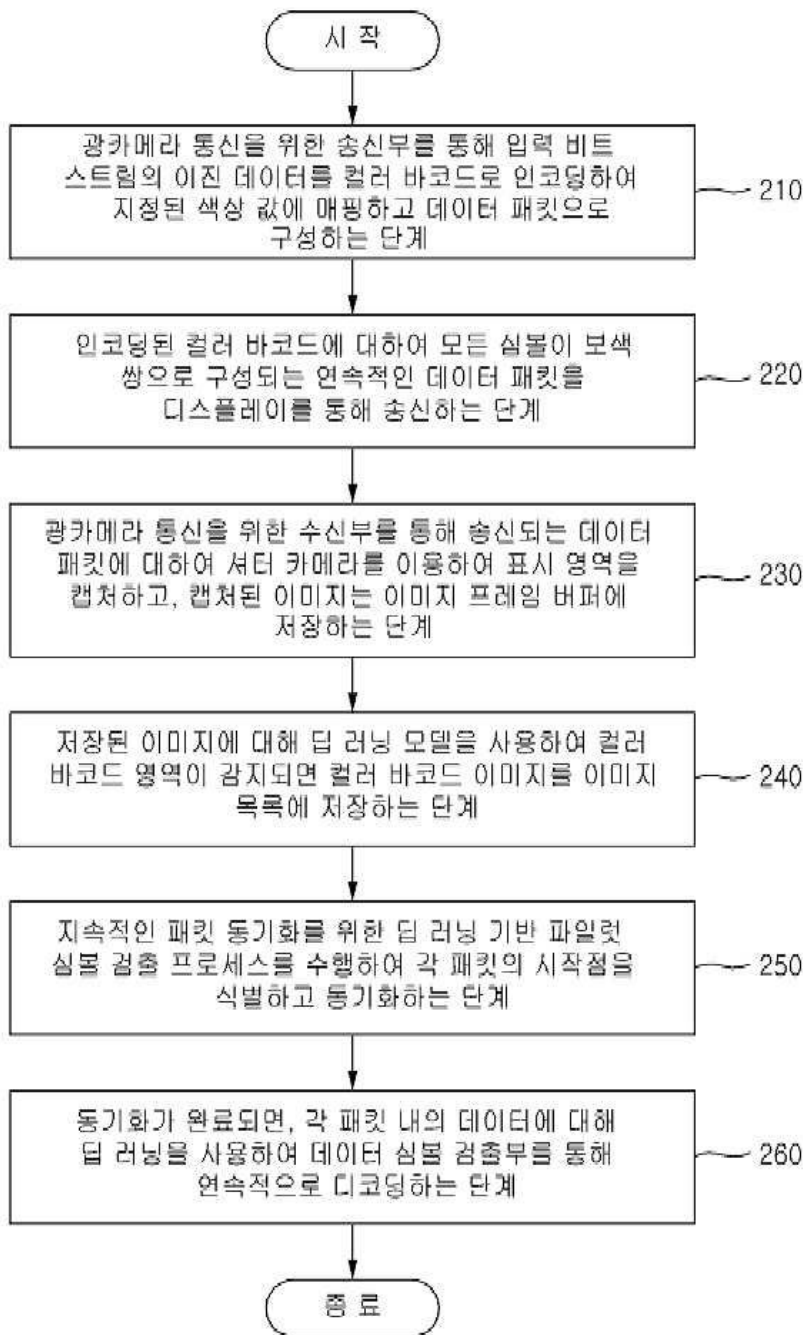
[0087] 그러므로, 다른 구현들, 다른 실시예들 및 특허청구범위와 균등한 것들도 후술하는 특허청구범위의 범위에 속한다.

도면

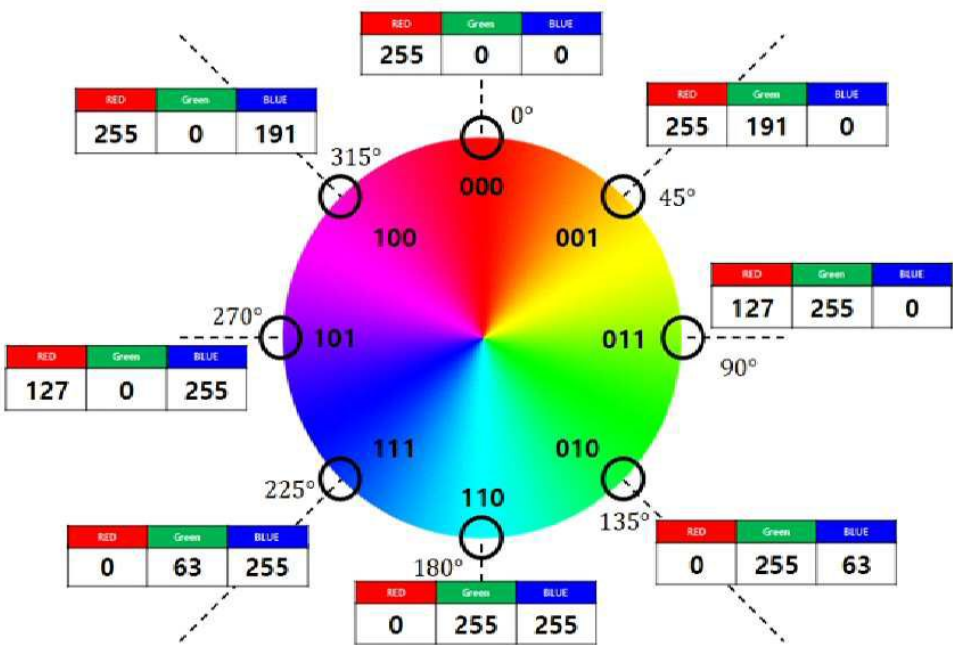
도면1



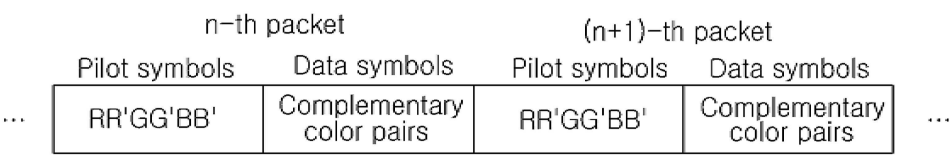
도면2



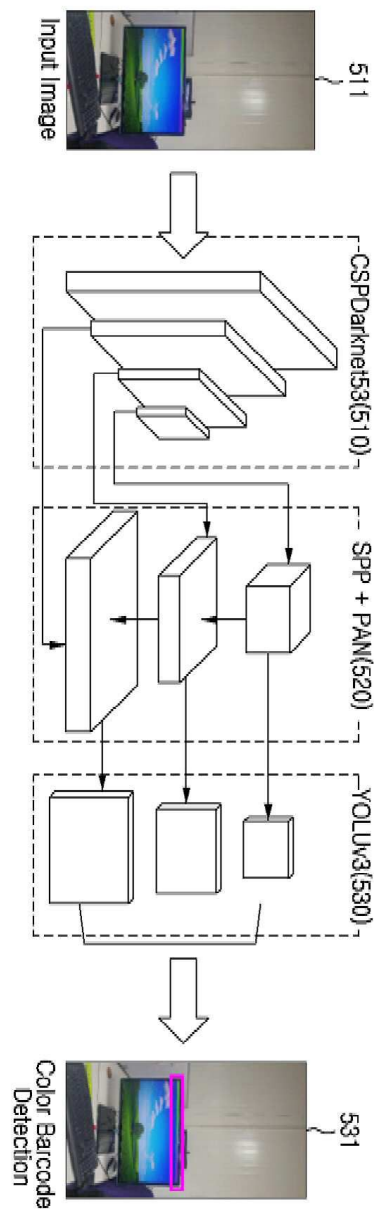
도면3



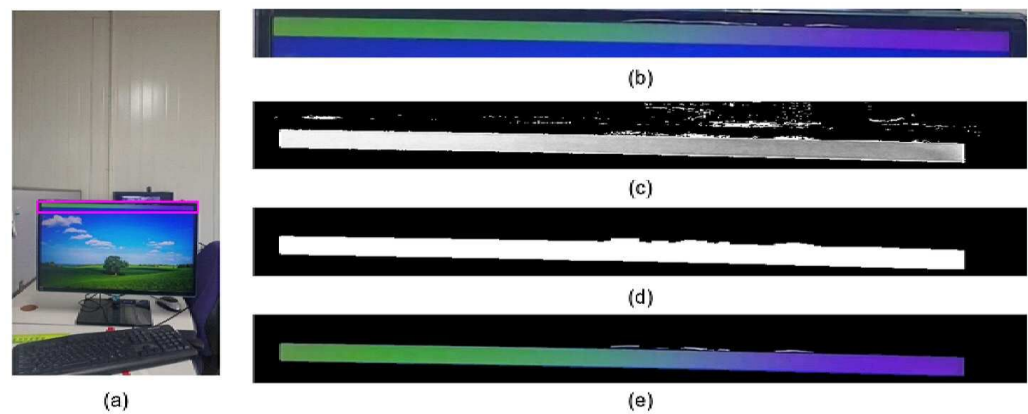
도면4



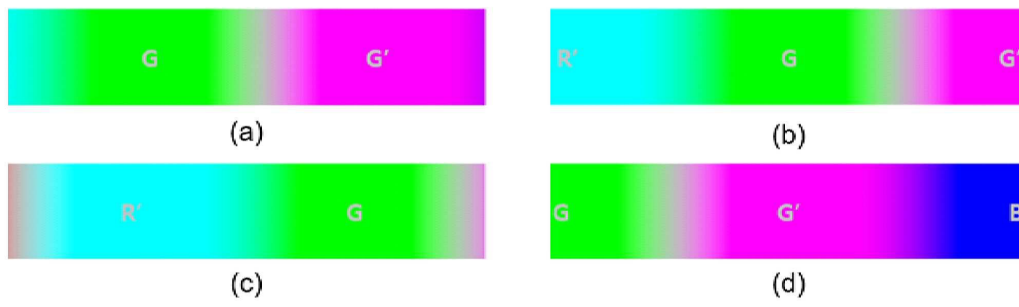
도면5



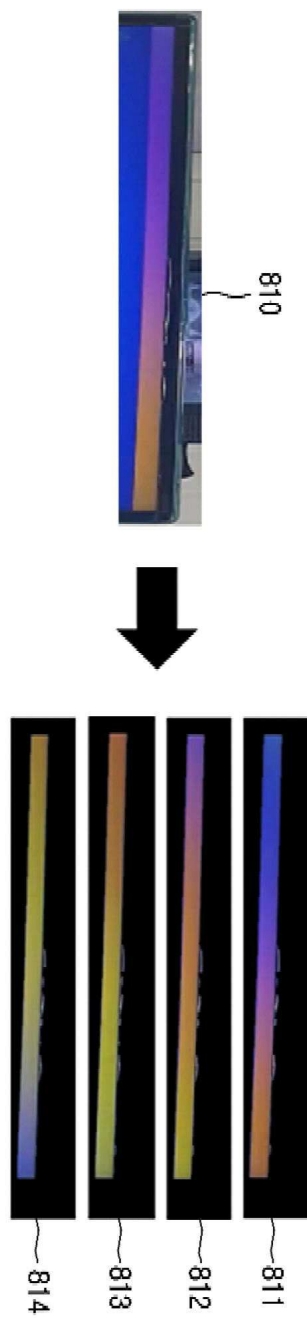
도면6



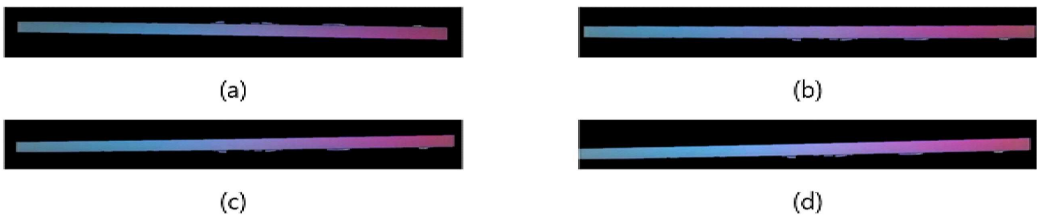
도면7



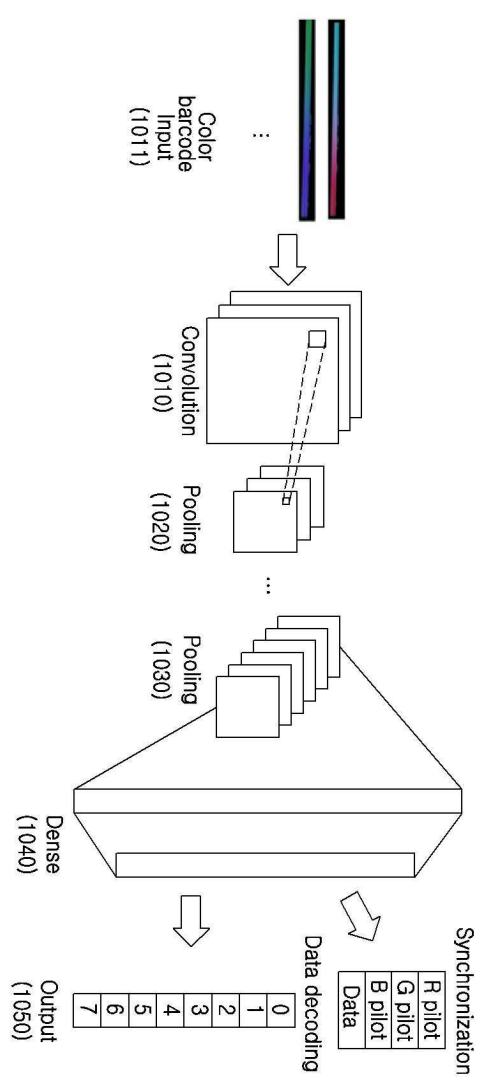
도면8



도면9



도면10



도면11

Algorithm 1: The entire process of DeepCCB-OCC.

Input: Captured Video with color barcodes

```
1: Synchronization ← False
2: while Captured Video do
3:   if Color barcode is detected then
4:     Area ← Color barcode area detected by YOLO
5:     Mask ← Differential current Area with previous Area
6:     Mask ← Morphological operation for Mask
7:     Barcode ← Extract Barcode using Mask
8:     if Synchronization then
9:       Data ← Barcode classification using CNN model
10:      Result.append(Data)
11:    else
12:      if Pilot symbols R, G, B are detected
        consecutively then
13:        Synchronization ← True
14:      end if
15:    end if
16:  end if
17: end while
Return Result
```
