

## 명세서

### 발명의 명칭: 전자 장치 및 그의 제어 방법

#### 기술분야

- [1] 본 발명은 전자 장치 및 그의 제어 방법에 관한 것이다.

#### 배경기술

- [2] 최근에는 코딩 교육이 의무화가 되면서 어린 시절부터 소프트웨어와 친해질 수 있도록, 각종 IT(Information Technology) 기술이 접목된 스마트 토이(Smart Toy)가 관심을 끌고 있고, 그 중 하나가 상기 코딩 로봇이다.
- [3] 그러나, 종래 코딩 로봇은, 코딩 작업에 의해서만 코딩 로봇이 작동됨으로써, 상기 코딩 로봇을 작동시키기 위해서는 코딩 교육이 꼭 선행되어야 하고, 그러한 코딩 교육이 선행되었더라도 만들어진 코딩에 오류가 있거나 코딩 실력이 부족하면 상기 코딩 로봇이 작동되지 않음에 따라, 상기 코딩 로봇에 대한 흥미가 급격히 감소되는 문제가 있었다.
- [4] 나아가, 종래의 코딩 로봇은 단순한 코딩 명령에 대한 단순한 동작만을 수행할 수 있어, 코딩 로봇의 사용자와의 다양한 인터랙션이 불가능하다는 한계가 존재하였다.
- [5] 이에 따라 쉽게 동작하되 사용자의 감정 등과 같은 다양한 입력에 따라 다양한 동작을 구현할 수 있는 코딩 로봇의 필요성이 대두되고 있다.

#### 발명의 상세한 설명

##### 기술적 과제

- [6] 본 발명이 해결하고자 하는 과제는 전자 장치 및 그의 제어 방법을 제공하는 것이다.
- [7] 본 발명이 해결하고자 하는 과제들은 이상에서 언급된 과제로 제한되지 않으며, 언급되지 않은 또 다른 과제들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

##### 기술적 해결방법

- [8] 상술한 과제를 해결하기 위한 본 발명의 일 면에 따른 코딩 로봇의 제어 방법은, 상기 코딩 로봇이, 인식 대상자의 정보를 획득하는 단계; 상기 코딩 로봇이, 상기 획득된 인식 대상자의 정보를 바탕으로 제어 데이터를 획득하는 단계; 상기 코딩 로봇이, 상기 제어 데이터가 기 설정된 제어 정보와 일치하는지 여부를 판단하는 단계; 상기 코딩 로봇이, 상기 제어 데이터가 기 설정된 제어 정보와 일치하는 경우, 상기 기 설정된 제어 명령에 대응되는 동작을 수행하는 단계; 및 상기 코딩 로봇이, 상기 제어 데이터가 기 설정된 제어 정보와 일치하지 않는 경우, 상기 제어 데이터를 분석하여, 기 설정된 제어 명령 중 어느 하나와 매칭 시키는 단계;를 포함하고, 상기 제어 데이터는, 상기 인식 대상자의 감정 데이터 및 음성 데이터 중 적어도 하나의 데이터인 것을 특징으로 한다.

### 발명의 효과

- [9] 상술한 본 발명의 다양한 실시예에 따라, 코딩 로봇은 인식 대상자의 감정 또는 음성을 판단하고, 판단된 감정 또는 음성에 대응되는 코딩 블록을 바탕으로 제어될 수 있다.
- [10] 본 발명의 효과들은 이상에서 언급된 효과로 제한되지 않으며, 언급되지 않은 또 다른 효과들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

### 도면의 간단한 설명

- [11] 도 1은 본 발명의 일 실시예에 따른 코딩 로봇을 설명하기 위한 예시도이다.
- [12] 도 2는 본 발명의 일 실시예에 따른 코딩 로봇의 구성을 나타낸 블록도이다.
- [13] 도 3은 본 발명의 일 실시예에 따른 코딩 로봇의 제어 방법을 설명하기 위한 흐름도이다.
- [14] 도 4 및 도 5는 본 발명의 일 실시예에 따른 인식된 사용자의 감정을 판단하는 방법을 설명하기 위한 예시도이다.

### 발명의 실시를 위한 최선의 형태

- [15] 상술한 과제를 해결하기 위한 본 발명의 일 면에 따른 코딩 로봇의 제어 방법은, 상기 코딩 로봇이, 인식 대상자의 정보를 획득하는 단계; 상기 코딩 로봇이, 상기 획득된 인식 대상자의 정보를 바탕으로 제어 데이터를 획득하는 단계; 상기 코딩 로봇이, 상기 제어 데이터가 기 설정된 제어 정보와 일치하는지 여부를 판단하는 단계; 상기 코딩 로봇이, 상기 제어 데이터가 기 설정된 제어 정보와 일치하는 경우, 상기 기 설정된 제어 명령에 대응되는 동작을 수행하는 단계; 및 상기 코딩 로봇이, 상기 제어 데이터가 기 설정된 제어 정보와 일치하지 않는 경우, 상기 제어 데이터를 분석하여, 기 설정된 제어 명령 중 어느 하나와 매칭 시키는 단계;를 포함하고, 상기 제어 데이터는, 상기 인식 대상자의 감정 데이터 및 음성 데이터 중 적어도 하나의 데이터인 것을 특징으로 한다.
- [16] 이때, 상기 제어 데이터가 상기 인식 대상자의 감정 데이터인 경우, 상기 제어 데이터를 획득하는 단계는, 상기 코딩 로봇이, 상기 인식 대상자의 얼굴 촬영 이미지를 획득하는 단계; 상기 코딩 로봇이, 상기 얼굴 촬영 이미지를 바탕으로 모델 변수를 획득하는 단계; 상기 코딩 로봇이, 상기 획득된 모델 변수를 바탕으로 상기 인식 대상자가 기 등록된 사용자의 얼굴과 일치하는지 판단하는 단계; 상기 인식 대상자가, 상기 기 등록된 사용자와 일치하는 경우, 상기 코딩 로봇이, 상기 얼굴 촬영 이미지를 바탕으로 상기 얼굴 촬영 이미지에 대응되는 특징값을 획득하는 단계; 상기 코딩 로봇이, 상기 획득된 특징값을 바탕으로 상기 인식 대상자의 눈, 코, 입을 포함하는 얼굴 부위를 판단하는 단계; 상기 코딩 로봇이, 상기 판단된 얼굴 부위를 바탕으로 상기 인식 대상자의 감정을 판단하는 단계; 및 상기 코딩 로봇이, 상기 판단된 감정을 바탕으로 상기 제어 데이터를 획득하는 단계;를 포함할 수 있다.

- [17] 이때, 상기 제어 명령에 대응되는 동작을 수행하는 단계는, 상기 코딩 로봇이, 상기 판단된 감정에 대응되는 2차원 벡터 공간상의 좌표 정보를 획득하는 단계; 상기 벡터 공간상의 좌표 정보에 대응되는 제어 명령을 획득하는 단계; 및 상기 제어 명령에 대응되는 동작을 수행하는 단계;를 포함할 수 있다.
- [18] 이때, 상기 제어 명령에 대응되는 동작을 수행하는 단계는, 상기 코딩 로봇이, 상기 인식 대상자의 얼굴 촬영 이미지를 바탕으로 상기 인식 대상자의 감정 변화 정보를 획득하는 단계; 상기 코딩 로봇이, 상기 인식 대상자의 감정이 제1 감정에서 제2 감정으로 변경된 경우, 상기 제1 감정에 대응되는 제1 좌표 정보 및 상기 제2 감정에 대응되는 제2 좌표 정보를 획득하는 단계; 및 상기 코딩 로봇이, 상기 제1 좌표 및 상기 제2 좌표에 대응되는 벡터값에 대응되는 이동 명령을 상기 제어 명령으로 획득하는 단계;를 포함할 수 있다.
- [19] 이때, 상기 제어 명령이, 상기 코딩 로봇을 이동시키기 위한 제어 명령인 경우, 상기 제어 방법은, 상기 코딩 로봇이, 상기 코딩 로봇 주변의 물체를 인식하는 단계; 상기 코딩 로봇이, 상기 인식된 물체가 기 저장된 표지판과 일치하는 경우, 상기 기 저장된 표지판에 대응되는 제어 정보를 획득하는 단계; 상기 코딩 로봇이, 상기 코딩 로봇 주변의 차선을 인식하는 단계; 및 상기 코딩 로봇이, 상기 인식된 차선을 따라 이동하되, 상기 제어 정보에 대응되게 이동하는 단계;를 포함하고, 상기 차선을 인식하는 단계는, 상기 코딩 로봇으로 입력되는 영상을 바탕으로 연속된 두개의 선분 정보를 획득하고, 획득된 선분 정보를 바탕으로 차선 영역을 획득하는 단계; 상기 획득된 선분과 상기 코딩 로봇의 거리 정보를 획득하는 단계; 및 상기 차선 영역 및 상기 거리 정보를 바탕으로 상기 코딩 로봇을 제어하는 단계;를 포함할 수 있다.
- [20] 이때, 상기 제어 데이터가 상기 인식 대상자의 음성 데이터인 경우, 상기 제어 방법은, 상기 코딩 로봇이, 상기 음성 데이터를 바탕으로 상기 코딩 로봇에 대한 동작 속도 제어 명령 및 동작 반복수 제어 명령을 포함하는 동작 제어 블록을 생성하는 단계; 상기 코딩 로봇이, 상기 음성 데이터를 바탕으로 동작 속도 정보 및 동작 반복수 정보 중 적어도 하나를 획득하는 단계; 및 상기 코딩 로봇이, 동작 속도 정보 및 동작 반복수 정보 중 적어도 하나를 바탕으로 상기 코딩 로봇의 동작 속도 및 동작 반복수 중 적어도 하나에 대응되는 제어 명령을 획득하는 단계;를 포함할 수 있다.
- [21] 이때, 상기 코딩 로봇의 동작 속도 및 동작 반복수 중 적어도 하나에 대응되는 제어 명령을 획득하는 단계는, 상기 코딩 로봇이, 상기 음성 데이터가 기 설정된 제어 정보와 일치하는 경우, 상기 기 설정된 제어 정보에 대응되는 동작 속도 정보 또는 동작 반복수 정보를 바탕으로 상기 코딩 로봇의 동작 속도 또는 동작 반복수를 상기 제어 명령으로 획득하는 단계; 상기 코딩 로봇이, 상기 음성 데이터가 기 설정된 제어 정보와 일치하지 않는 경우, 상기 음성 데이터를 바탕으로 음절 간격 정보를 생성하고, 상기 음절 간격 정보에 대응하여 상기 동작 속도 정보를 생성하는 단계; 상기 코딩 로봇이, 상기 음성 데이터가 기

설정된 제어 정보와 일치하지 않는 경우, 상기 음성 데이터를 바탕으로 신호 세기 정보를 생성하고, 상기 신호 세기 정보에 대응하여 상기 동작 반복수 정보를 생성하는 단계; 및 상기 생성된 동작 속도 정보 또는 동작 반복수를 바탕으로 상기 코딩 로봇의 동작 속도 또는 동작 반복수를 상기 제어 명령으로 획득하는 단계;를 포함할 수 있다.

[22] 본 발명의 기타 구체적인 사항들은 상세한 설명 및 도면들에 포함되어 있다.

#### **발명의 실시를 위한 형태**

[23] 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나, 본 발명은 이하에서 개시되는 실시예들에 제한되는 것이 아니라 서로 다른 다양한 형태로 구현될 수 있으며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하고, 본 발명이 속하는 기술 분야의 통상의 기술자에게 본 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다.

[24] 본 명세서에서 사용된 용어는 실시예들을 설명하기 위한 것이며 본 발명을 제한하고자 하는 것은 아니다. 본 명세서에서, 단수형은 문구에서 특별히 언급하지 않는 한 복수형도 포함한다. 명세서에서 사용되는 "포함한다(comprises)" 및/또는 "포함하는(comprising)"은 언급된 구성요소 외에 하나 이상의 다른 구성요소의 존재 또는 추가를 배제하지 않는다. 명세서 전체에 걸쳐 동일한 도면 부호는 동일한 구성 요소를 지칭하며, "및/또는"은 언급된 구성요소들의 각각 및 하나 이상의 모든 조합을 포함한다. 비록 "제1", "제2" 등이 다양한 구성요소들을 서술하기 위해서 사용되나, 이들 구성요소들은 이들 용어에 의해 제한되지 않음은 물론이다. 이들 용어들은 단지 하나의 구성요소를 다른 구성요소와 구별하기 위하여 사용하는 것이다. 따라서, 이하에서 언급되는 제1 구성요소는 본 발명의 기술적 사상 내에서 제2 구성요소일 수도 있음은 물론이다.

[25] 다른 정의가 없다면, 본 명세서에서 사용되는 모든 용어(기술 및 과학적 용어를 포함)는 본 발명이 속하는 기술분야의 통상의 기술자에게 공통적으로 이해될 수 있는 의미로 사용될 수 있을 것이다. 또한, 일반적으로 사용되는 사전에 정의되어 있는 용어들은 명백하게 특별히 정의되어 있지 않는 한 이상적으로 또는 과도하게 해석되지 않는다.

[26] 명세서에서 사용되는 "부" 또는 "모듈"이라는 용어는 소프트웨어, FPGA 또는 ASIC과 같은 하드웨어 구성요소를 의미하며, "부" 또는 "모듈"은 어떤 역할들을 수행한다. 그렇지만 "부" 또는 "모듈"은 소프트웨어 또는 하드웨어에 한정되는 의미는 아니다. "부" 또는 "모듈"은 어드레싱할 수 있는 저장 매체에 있도록 구성될 수도 있고 하나 또는 그 이상의 프로세서들을 재생시키도록 구성될 수도 있다. 따라서, 일 예로서 "부" 또는 "모듈"은 소프트웨어 구성요소들, 객체지향

소프트웨어 구성요소들, 클래스 구성요소들 및 태스크 구성요소들과 같은 구성요소들과, 프로세스들, 함수들, 속성들, 프로시저들, 서브루틴들, 프로그램 코드의 세그먼트들, 드라이버들, 펌웨어, 마이크로 코드, 회로, 데이터, 데이터베이스, 데이터 구조들, 테이블들, 어레이들 및 변수들을 포함한다. 구성요소들과 "부" 또는 "모듈"들 안에서 제공되는 기능은 더 작은 수의 구성요소들 및 "부" 또는 "모듈"들로 결합되거나 추가적인 구성요소들과 "부" 또는 "모듈"들로 더 분리될 수 있다.

- [27] 공간적으로 상대적인 용어인 "아래(below)", "아래(beneath)", "하부(lower)", "위(above)", "상부(upper)" 등은 도면에 도시되어 있는 바와 같이 하나의 구성요소와 다른 구성요소들과의 상관관계를 용이하게 기술하기 위해 사용될 수 있다. 공간적으로 상대적인 용어는 도면에 도시되어 있는 방향에 더하여 사용시 또는 동작시 구성요소들의 서로 다른 방향을 포함하는 용어로 이해되어야 한다. 예를 들어, 도면에 도시되어 있는 구성요소를 뒤집을 경우, 다른 구성요소의 "아래(below)"또는 "아래(beneath)"로 기술된 구성요소는 다른 구성요소의 "위(above)"에 놓여질 수 있다. 따라서, 예시적인 용어인 "아래"는 아래와 위의 방향을 모두 포함할 수 있다. 구성요소는 다른 방향으로도 배향될 수 있으며, 이에 따라 공간적으로 상대적인 용어들은 배향에 따라 해석될 수 있다.
- [28] 본 명세서에서, 컴퓨터는 적어도 하나의 프로세서를 포함하는 모든 종류의 하드웨어 장치를 의미하는 것이고, 실시 예에 따라 해당 하드웨어 장치에서 동작하는 소프트웨어적 구성도 포괄하는 의미로서 이해될 수 있다. 예를 들어, 컴퓨터는 스마트폰, 태블릿 PC, 데스크톱, 노트북 및 각 장치에서 구동되는 사용자 클라이언트 및 애플리케이션을 모두 포함하는 의미로서 이해될 수 있으며, 또한 이에 제한되는 것은 아니다.
- [29] 이하, 첨부된 도면을 참조하여 본 발명의 실시예를 상세하게 설명한다.
- [30] 본 명세서에서 설명되는 각 단계들은 컴퓨터에 의하여 수행되는 것으로 설명되나, 각 단계의 주체는 이에 제한되는 것은 아니며, 실시 예에 따라 각 단계들의 적어도 일부가 서로 다른 장치에서 수행될 수도 있다.
- [31] 도 1은 본 발명의 일 실시예에 따른 코딩 로봇을 설명하기 위한 예시도이다.
- [32] 본 발명에 따른 코딩 로봇(100)은 컴퓨터 언어로 프로그래밍된 코딩 언어 또는 프로그래밍된 코딩 블록에 의해 동작할 수 있다. 이때, 코딩이란 C언어 등의 컴퓨터 언어를 이용하여 순차적이고 논리적인 흐름을 프로그래밍(programming)하는 작업이고, 코딩 블록이란 특정 명령어를 수행하기 위하여 완성된 코딩 명령을 포함하는 블록을 의미할 수 있다. 즉, 코딩 로봇(100)은 작성된 코딩 또는 선택된 코딩 블록이 전달하는 명령에 따라 작동할 수 있다.
- [33] 본 발명에 따른 코딩 로봇(100)은, 카메라, 마이크, 메모리, 프로세서 및 구동부를 포함할 수 있다.

- [34] 도 2는 본 발명의 일 실시예에 따른 코딩 로봇의 구성을 상세히 나타내는 블록도이다.
- [35] 구체적으로, 코딩 로봇(100)은 입력부(110), 출력부(120), 프로세서(130)외에 통신부(140), 메모리(150) 및 감지부(160)를 더 포함할 수 있다.
- [36] 입력부(110)는 다양한 외부 입력을 획득하기 위하여 카메라(111), 마이크(112), 터치 패널(113) 등을 포함할 수 있다.
- [37] 카메라(111)는 코딩 로봇(100)주변의 영상 데이터를 획득하기 위한 구성이다. 카메라(111)는 정지 영상 및 동영상을 촬영할 수 있다. 예로, 카메라(111)는 하나 이상의 이미지 센서, 렌즈, 이미지 시그널 프로세서(ISP), 또는 플래시(예: LED 또는 xenon lamp 등)를 포함할 수 있으며, 복수의 카메라(111)로 구성될 수 있다. 이때, 불필요한 전력 소모를 막기 위해 카메라(111)는 특정 조건을 만족하는 경우에만 활성화될 수 있다. 예를 들어, 카메라(111)는 코딩 로봇(100) 주변에 사용자가 감지된 경우에만 활성화될 수 있다. 또는 코딩 로봇(100)이 IoT 시스템의 일부인 경우, 카메라(111)는 현관문이 열리거나, 실내 조명이 점등되는 조건을 만족하는 경우 활성화될 수 있다. 다만, 상술한 실시예에 한정되는 것은 아니고, 카메라(111)는 로봇에 전원이 인가되면 항상 활성화될 수도 있다. 카메라(111)가 항상 활성화된 경우, 카메라(111)는 특정 조건을 만족하는 경우에만, 영상을 저장하거나, 영상 처리를 수행할 수 있다. 특정 조건이란, 코딩 로봇(100) 주변에 사용자가 감지된 경우에 대한 조건, 코딩 로봇(100)이 IoT 시스템의 일부인 경우, 카메라(111)는 현관문이 열리거나, 실내 조명이 점등되는 조건일 수 있다.
- [38] 마이크(112)는 코딩 로봇(100) 주변의 소리를 획득하기 위한 구성이다. 마이크(112)는 외부의 음향 신호를 입력 받아 전기적인 음성 정보를 생성할 수 있으며, 코딩 로봇(100)은 복수개의 마이크(112)를 구비할 수 있다. 코딩 로봇(100)은 복수개의 마이크 각각에 입력되는 음향신호의 도착 시간 차이를 이용하여 음향 신호를 발생시키는 음원(Sound Source)의 방향을 추정할 수 있다. 마이크(112)는 외부의 음향 신호를 입력 받는 과정에서 발생하는 잡음(noise)를 제거하기 위한 다양한 잡음 제거 알고리즘을 이용할 수 있다. 카메라(111)의 경우와 같이, 마이크(112)는 불필요한 전력 소모를 막기 위해 특정 조건을 만족하는 경우에만 활성화될 수 있다. 즉, 마이크(112)는 코딩 로봇(100) 주변에 사용자가 감지된 경우에만 활성화될 수 있다. 또는 코딩 로봇(100)이 IoT 시스템의 일부인 경우, 마이크(112)는 현관문이 열리거나, 실내 조명이 점등되는 조건을 만족하는 경우 활성화될 수 있다. 다만, 상술한 실시예에 한정되는 것은 아니고, 카메라(111)의 경우와 같이 마이크(112)는 항상 활성화되어 있을 수도 있음은 물론이다.
- [39] 터치패널(113)은 다양한 사용자 입력을 입력 받을 수 있는 구성이다. 터치 패널(113)은 사용자 조작에 의한 사용자 데이터를 입력 받을 수 있다. 터치 패널 또한 불필요한 전력 소모를 막기 위해 상기와 같은 조건하에서 활성화될 수

- 있다. 또는 터치 패널(113)은 사용자의 터치가 감지된 경우에 한하여 활성화될 수도 있다. 터치 패널(113)은 후술하는 디스플레이와 결합하여 구성될 수도 있다.
- [40] 입력부(110)는 상술한 카메라(111), 마이크(112), 터치 패널(113) 외에도 다양한 사용자 데이터를 입력 받기 위한 다양한 구성일 수 있음은 물론이다.
- [41] 출력부(112)는 구동부(121), 오디오 출력부(122) 및 디스플레이(123)로 구성될 수 있다.
- [42] 구동부(121)는 입력부(110)를 통해 획득한 사용자 데이터에 대한 반응으로 다양한 모션을 출력하기 위한 구성이다. 본 발명의 일 실시예에 따른 코딩 로봇(100)이 인간형 코딩 로봇(100)인 경우, 구동부(121)는 인간의 관절에 대응되는 적어도 하나의 관절부를 구동할 수 있는 모터 또는 액추에이터(Actuator)를 포함할 수 있다. 또 다른 실시예로, 본 발명의 일 실시예에 따른 코딩 로봇(100)이 바퀴형 로봇인 경우, 구동부(121)은 바퀴를 구동할 수 있는 모터 또는 액추에이터를 포함할 수 있다.
- [43] 오디오 출력부(122)는 입력부(110)를 통해 획득한 사용자 데이터에 대한 반응으로, 다양한 오디오 신호를 출력하기 위한 구성이다. 오디오 출력부(122)는 오디오 처리부(미도시)에 의해 디코딩이나 증폭, 노이즈 필터링과 같은 다양한 처리 작업이 수행된 각종 오디오 데이터뿐만 아니라 각종 알림 음이나 음성 메시지를 출력할 수 있다. 특히, 오디오 출력부(122)는 스피커로 구현될 수 있으나, 이는 일 실시 예에 불과할 뿐, 오디오 데이터를 출력할 수 있는 출력 단자로 구현될 수 있다.
- [44] 디스플레이(123)는 입력부(110)를 통해 획득한 사용자 데이터에 대한 다양한 반응으로, 영상을 출력하기 위한 구성이다. 일 실시예로, 디스플레이(123)는 감정을 표현하기 위한 아바타(Avatar) 영상을 출력할 수 있다. 즉, 코딩 로봇(100)이 입력부(110)를 통해 획득된 사용자 데이터에 대한 반응으로 기뻐하는 감정을 출력하고자 하는 경우, 디스플레이(123)에 표시되는 아바타 영상을 통해 기뻐하는 감정을 출력할 수 있다. 한편, 다양한 영상을 제공하기 위한 디스플레이(123)는 다양한 형태의 디스플레이 패널로 구현될 수 있다. 예를 들어, 디스플레이 패널은 LCD(Liquid Crystal Display), OLED(Organic Light Emitting Diodes), AM-OLED(Active-Matrix Organic Light-Emitting Diode), LcoS(Liquid Crystal on Silicon) 또는 DLP(Digital Light Processing) 등과 같은 다양한 디스플레이 기술로 구현될 수 있다. 또한, 디스플레이(123)는 플렉서블 디스플레이(flexible display)의 형태로 코딩 로봇(100)의 전면 영역 및, 측면 영역 및 후면 영역 중 적어도 하나에 결합될 수도 있다.
- [45] 통신부(140)는 외부 장치와 통신을 수행할 수 있다. 특히, 통신부(140)는 와이파이 칩(141), 블루투스 칩(142), NFC 칩(143) 및 이동 통신 칩(144) 등과 같은 무선 통신을 수행하기 위한 다양한 통신 칩 또는 회로를 포함할 수 있다. 이때, 와이파이 칩(141), 블루투스 칩(142), NFC 칩(143)은 각각 LAN 방식, WiFi 방식, 블루투스 방식, NFC 방식으로 통신을 수행한다. 와이파이 칩(141)이나

블루투스칩(142)을 이용하는 경우에는 SSID 및 세션 키 등과 같은 각종 연결 정보를 먼저 송수신한 후, 이를 이용하여 통신을 연결하여 각종 정보들을 송수신할 수 있다. 이동 통신칩(144)은 3G(3rd Generation), 3GPP(3rd Generation Partnership Project), LTE(Long Term Evolution) 등과 같은 다양한 통신 규격에 따라 통신을 수행하는 칩을 의미한다. 한편, 통신부(140)는 USB(universal serial bus), HDMI(high definition multimedia interface), RS-232(recommended standard 232), 전력선 통신, 또는 POTS(plain old telephone service) 등 중 적어도 하나를 포함하는 유선 통신 방식으로 통신을 수행할 수도 있음은 물론이다. 특히 통신부는 외부 장치들과 통신을 수행하여 사용자 상태를 판단하기 위한 다양한 데이터를 송수신할 수 있다. 또는 사용자 상태 분석이 외부 서버에서 수행되는 경우, 통신부(140)는 수집한 학습 데이터를 외부 서버로 전송하고, 학습 데이터의 분석 결과를 수신할 수 있다.

- [46] 메모리(150)는, 예를 들면, 코딩 로봇(100)의 적어도 하나의 구성요소를 제어하기 위한 명령 또는 데이터를 저장할 수 있다. 일 실시예에 따르면, 메모리(150)는 소프트웨어 및/또는 프로그램을 저장할 수 있다. 프로그램은, 예를 들면, 커널(Kernel), 미들웨어(Middle-ware), 어플리케이션 프로그래밍 인터페이스(Application Programming Interface: API) 및/또는 어플리케이션 프로그램(또는 "어플리케이션") 등을 포함할 수 있다. 커널, 미들웨어 또는 API의 적어도 일부는, 운영 시스템(Operating System: OS)으로 지칭될 수 있다. 커널은, 예를 들면, 다른 프로그램들에 구현된 동작 또는 기능을 실행하는 데 사용되는 시스템 리소스들을 제어 또는 관리할 수 있다. 또한, 커널은 미들웨어, API, 또는 어플리케이션 프로그램에서 코딩 로봇(100)의 개별 구성요소에 접근할 때, 시스템 리소스들을 제어 또는 관리할 수 있는 인터페이스를 제공할 수 있다.
- [47] 미들웨어는, 예를 들면, API 또는 어플리케이션 프로그램이 커널과 통신하여 데이터를 주고받을 수 있도록 중개 역할을 수행할 수 있다. 또한, 미들웨어는 어플리케이션 프로그램으로부터 수신된 하나 이상의 작업 요청들을 우선 순위에 따라 처리할 수 있다. 예를 들면, 미들웨어는 어플리케이션 프로그램 중 적어도 하나에 코딩 로봇(100)의 시스템 리소스를 사용할 수 있는 우선 순위를 부여하고, 상기 하나 이상의 작업 요청들을 처리할 수 있다. API는 어플리케이션 프로그램이 커널 또는 미들웨어에서 제공되는 기능을 제어하기 위한 인터페이스로, 예를 들면, 파일 제어, 창 제어, 영상 처리, 또는 문자 제어 등을 위한 적어도 하나의 인터페이스 또는 함수(예: 명령어)를 포함할 수 있다.
- [48] 또한, 메모리(150)는, 내장 메모리 및 외장 메모리 중 적어도 하나를 포함할 수 있다. 내장 메모리는, 예를 들면, 휘발성 메모리(예: DRAM, SRAM, 또는 SDRAM 등), 비휘발성 메모리(예: OTPROM(one time programmable ROM), PROM, EPROM, EEPROM, mask ROM, flash ROM, 플래시 메모리, 하드 드라이브, 또는 솔리드 스테이트 드라이브(Solid State Drive: SSD) 중 적어도 하나를 포함할 수 있다. 외장 메모리는 플래시 드라이브(flash drive), 예를 들면, CF(compact flash),

SD(secure digital), Micro-SD, Mini-SD, xD(extreme digital), MMC(multi-media card) 또는 메모리 스틱 등을 포함할 수 있다. 외장 메모리는 다양한 인터페이스를 통하여 코딩 로봇(100)과 기능적으로 또는 물리적으로 연결될 수 있다.

- [49] 감지부(160)는 코딩 로봇(100)의 주변 정보를 획득하기 위한 구성이다. 상술한 바와 같이, 감지부(160)는 다양한 센서로 구성될 수 있으며, 센서에 의해 감지된 정보를 사용자 데이터로 획득할 수 있다. 감지부(160)는 적외선 센서, 압력 센서, 조도 센서, 습도 센서, 자이로 센서 등 다양한 센서를 포함할 수 있다. 예를 들어, 감지부(160)가 적외선 센서를 포함하는 경우, 적외선 센서를 통해 사용자의 존재 유무에 대한 데이터를 사용자 데이터로 획득하거나, 사용자의 체온 등에 대한 데이터를 획득할 수 있다. 또는, 감지부(160)가 압력 센서를 포함하는 경우, 사용자와의 물리적인 접촉에 대한 데이터를 사용자 데이터로 획득할 수 있다. 또는, 코딩 로봇(100)이 조도 센서, 습도 센서 등을 포함하는 경우, 사용자의 주변 환경과 관련된 습도 및 조도에 대한 정보를 사용자 데이터로 획득할 수 있다.
- [50] 프로세서(130)는 메모리(150)에 저장된 각종 프로그램을 이용하여 코딩 로봇(100)의 전반적인 동작을 제어할 수 있다.
- [51] 프로세서(130)는 RAM(131), ROM(132), 그래픽 처리부(133), 메인 CPU(134), 및 제1 내지 n 인터페이스(135-1~135-n) 및 버스(136)로 구성될 수 있다. 이때, RAM(131), ROM(132), 그래픽 처리부(133), 메인 CPU(134) 제1 내지 n 인터페이스(135-1 ~ 135-n) 등은 버스(136)를 통해 서로 연결될 수 있다.
- [52] RAM(131)은 OS 및 어플리케이션 프로그램을 저장한다. 구체적으로, 코딩 로봇(100)이 부팅되면 OS가 RAM(131)에 로딩(loading)되고, 사용자가 선택한 각종 어플리케이션 데이터가 RAM(131)에 로딩될 수 있다.
- [53] ROM(132)에는 시스템 부팅을 위한 명령어 세트 등이 저장된다. 턴 온(Turn-On) 명령이 입력되어 전원이 공급되면, 메인 CPU(134)는 ROM(132)에 저장된 명령어에 따라 메모리(150)에 저장된 OS를 RAM(131)에 복사하고, OS를 실행시켜 시스템을 부팅시킨다. 부팅이 완료되면, 메인 CPU(134)는 메모리(150)에 저장된 각종 어플리케이션 프로그램을 RAM(131)에 복사하고, RAM(131)에 복사된 어플리케이션 프로그램을 실행시켜 각종 동작을 수행한다.
- [54] 그래픽 처리부(133)는 연산부(미도시) 및 렌더링부(미도시)를 이용하여 GUI(Graphical User Interface) 아이템, 이미지, 텍스트 등과 같은 다양한 객체를 포함하는 화면을 생성한다. 여기서, 연산부는 입력부(140)로부터 수신된 제어 명령을 이용하여 화면의 레이아웃에 따라 각 객체들이 표시될 좌표값, 형태, 크기, 컬러 등과 같은 속성값을 연산하는 구성일 수 있다. 그리고, 렌더링부는 연산부에서 연산한 속성값에 기초하여 객체를 포함하는 다양한 레이아웃의 화면을 생성하는 구성일 수 있다. 이러한 렌더링부에서 생성된 화면은 디스플레이(130)의 디스플레이 영역 내에 표시될 수 있다.
- [55] 메인 CPU(134)는 메모리(150)에 액세스하여, 메모리(150)에 저장된 OS를 이용하여 부팅을 수행한다. 그리고, 메인 CPU(134)는 메모리(150)에 저장된 각종

프로그램, 콘텐츠, 데이터 등을 이용하여 다양한 동작을 수행한다.

- [56] 제1 내지  $n$  인터페이스(135-1 내지 135- $n$ )는 상술한 각종 구성요소들과 연결된다. 제1 내지  $n$  인터페이스(135-1 내지 135- $n$ ) 중 하나는 네트워크를 통해 외부 장치와 연결되는 네트워크 인터페이스가 될 수도 있다.
- [57] 이하에서는, 도 3 내지 도 5를 이용하여 본 발명에 따른 다양한 실시예를 설명한다.
- [58] 도 3은 본 발명의 일 실시예에 따른 코딩 로봇의 제어 방법을 설명하기 위한 흐름도이다.
- [59] 단계 S110에서, 코딩 로봇(100)은, 인식 대상자의 정보를 획득할 수 있다.
- [60] 일 실시예로, 인식 대상자의 정보는 상술한 입력부(110) 또는 감지부(160)에 의해 획득될 수 있다.
- [61] 단계 S120에서, 코딩 로봇(100)은, 획득된 인식 대상자의 정보를 바탕으로 제어 데이터를 획득할 수 있다.
- [62] 단계 S130에서, 코딩 로봇(100)은, 제어 데이터가 기 설정된 제어 정보와 일치하는지 여부를 판단할 수 있다.
- [63] 일 실시예로, 코딩 로봇(100)은, 제어 데이터와 매칭된 제어 정보를 메모리(150)에 미리 저장하고 있을 수 있다. 예를 들어, 제어 데이터가 화난 감정인 경우, 메모리(150)는 화난 감정에 대응되는 코딩 로봇(100)의 동작 정보를 제어 명령으로 저장하고 있을 수 있다.
- [64] 단계 S140에서, 코딩 로봇(100)은, 제어 데이터가 기 설정된 제어 정보와 일치하는 경우, 기 설정된 제어 명령에 대응되는 동작을 수행할 수 있다.
- [65] 단계 S150에서, 코딩 로봇(100)은, 제어 데이터가 기 설정된 제어 정보와 일치하지 않는 경우, 제어 데이터를 분석하여, 기 설정된 제어 명령 중 어느 하나와 매칭 시킬 수 있다.
- [66] 즉, 인식 대상자의 정보를 바탕으로 획득된 제어 데이터에 매칭된 제어 명령이 존재하지 않는 경우라도, 코딩 로봇(100)은 제어 데이터를 분석하여 기 저장된 제어 명령 중 적어도 하나의 제어 명령에 매칭시킬 수 있다.
- [67] 한편, 본 발명에서 제어 데이터는, 인식 대상자의 감정 데이터 및 음성 데이터 중 적어도 하나의 데이터를 의미할 수 있다.
- [68] 한편, 본 발명의 다양한 실시예에 따라, 제어 데이터가 인식 대상자의 감정 데이터인 경우, 코딩 로봇(100)은 다양한 방법을 통해 제어 데이터를 획득할 수 있다.
- [69] 구체적으로, 코딩 로봇(100)은, 인식 대상자의 얼굴 촬영 이미지를 획득할 수 있다. 이후, 코딩 로봇(100)은, 얼굴 촬영 이미지를 바탕으로 모델 변수를 획득할 수 있다. 이후, 코딩 로봇(100)은, 획득된 모델 변수를 바탕으로 인식 대상자가 기 등록된 사용자의 얼굴과 일치하는지 판단할 수 있다. 이후, 인식 대상자가, 기 등록된 사용자와 일치하는 경우, 코딩 로봇(100)은, 얼굴 촬영 이미지를 바탕으로 얼굴 촬영 이미지에 대응되는 특징값을 획득할 수 있다. 이후, 코딩 로봇(100)은,

획득된 특징값을 바탕으로 인식 대상자의 눈, 코, 입을 포함하는 얼굴 부위를 판단할 수 있다. 이후, 코딩 로봇(100)은, 판단된 얼굴 부위를 바탕으로 인식 대상자의 감정을 판단할 수 있다.

- [70] 일 실시예로, 코딩 로봇(100)은 OpenCV, Tensorflow, Dlib라는 라이브러리를 바탕으로 코딩로봇의 사용자를 인식하고, 인식 대상자의 감정을 판단할 수 있다.
- [71] 이때, OpenCV는 실시간 컴퓨터 비전을 목적으로 한 프로그래밍 라이브러리로, 로보미는 카메라를 다루고 카메라 영상을 처리할 수 있으며, TensorFlow는 다양한 작업에 대해 데이터 흐름 프로그래밍을 위한 오픈소스 소프트웨어 라이브러리로 심볼릭 수학 라이브러리이자, 뉴럴 네트워크같은 기계학습 응용프로그램에도 사용되며 얼굴 인식 및 물체 인식에 사용될 수 있으며, Dlib는 프로그래밍 언어 C++로 작성된 범용 크로스 플랫폼 소프트웨어 라이브러리로, 코딩 로봇(100)은 Dlib중 얼굴 랜드마크(눈, 코, 입 등 영역) 인식 기능을 이용해 감정 인식을 수행할 수 있다.
- [72] 구체적으로, 코딩 로봇(100)은 상술한 다양한 라이브러리를 바탕으로 얼굴 인식을 수행할 수 있다. 이를 위해, 코딩 로봇(100)은 기 작성된 코딩 블록을 포함할 수 있다.
- [73] 구체적으로, 얼굴 저장 블록은 인식된 얼굴을 OpenCV를 이용해 이미지 파일로 저장할 수 있다.
- [74] 또한, 얼굴 모델 생성 블록은 저장된 얼굴 사진을 기반으로, Tensorflow 라이브러리를 이용해 모델 변수를 생성할 수 있다. 생성된 모델 변수는 얼굴 인지 블록에서 사용될 수 있다.
- [75] 또한, 얼굴 인지 블록은 모델 변수를 이용해서, 얼굴이 등록된 얼굴과 얼마나 일치하는지 계산할 수 있다.
- [76] 또한, 얼굴 감정 인식 블록은 Dlib를 이용해, 도 4에 도시된 바와 같이, 얼굴의 꼭짓점 등을 점으로 나타냅니다. 이 점들의 위치값을 이용하여 눈의 크기나 눈썹 각도, 입의 크기 등을 계산해 감정을 추론할 수 있다. 이외에도 얼굴 감정 인식 모델은 눈, 코, 입의 위치를 계산할 수 있으므로, 눈코입 위치에 이미지를 덧그릴 수 있음은 물론이다.
- [77] 일 실시예에 따라, 코딩 로봇(100)은 하기 코드를 활용하여 인식 대상자의 얼굴 표정을 판단할 수 있다.
- [78] [알고리즘 1]
- [79] def detect\_face\_elem(o\_frame):
- [80] global detector
- [81] global predictor
- [82]
- [83] if detector is None or predictor is None:
- [84] if "dlib" not in sys.modules:
- [85] import dlib

```
[86]     detector = dlib.get_frontal_face_detector()
[87]     predictor = dlib.shape_predictor('shape_predictor_68_face_landmarks.dat')
[88]
[89]     frame = copy.deepcopy(o_frame)
[90]     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
[91]
[92]     line_brow_x = []
[93]     line_brow_y = []
[94]
[95]     faces = detector(gray)
[96]     for face in faces:
[97]
[98]         #200107
[99]         x1 = face.left()
[100]        y1 = face.top()
[101]        x2 = face.right()
[102]        y2 = face.bottom()
[103]        cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 1)
[104]
[105]
[106]        landmarks = predictor(gray, face)
[107]
[108]        ## "Jaw"
[109]        for n in range(0, 17):
[110]            x = landmarks.part(n).x
[111]            y = landmarks.part(n).y
[112]            cv2.circle(frame, (x, y), 1, (19, 199, 109), -1)
[113]        ## "Nose"
[114]        for n in range(27, 35):
[115]            x = landmarks.part(n).x
[116]            y = landmarks.part(n).y
[117]            cv2.circle(frame, (x, y), 1, (180, 42, 220), -1)
[118]            #cv2.putText(frame, str(n), (x, y), cv2.FONT_HERSHEY_SIMPLEX, 0.35, (0, 255,
255), 2, 4)
[119]            #31-34, 27-33
[120]
[121]        ## "Left Eye"
[122]        for n in range(42, 48):
```

```

[123]     x = landmarks.part(n).x
[124]     y = landmarks.part(n).y
[125]     cv2.circle(frame, (x, y), 1, (163, 38, 32), -1)
[126]
[127]     ## "Right Eye"
[128]     for n in range(36, 42):
[129]         x = landmarks.part(n).x
[130]         y = landmarks.part(n).y
[131]         cv2.circle(frame, (x, y), 1, (158, 163, 32), -1)
[132]         #cv2.putText(frame, str(n), (x, y), cv2.FONT_HERSHEY_SIMPLEX, 0.4, (0, 0,
255), 2, 4)
[133]
[134]     ## "Left Eyebrow"
[135]     for n in range(22, 27):
[136]         x = landmarks.part(n).x
[137]         y = landmarks.part(n).y
[138]         cv2.circle(frame, (x, y), 1, (168, 100, 168), -1)
[139]
[140]     ## "Right Eyebrow"
[141]     for n in range(17, 22):
[142]         x = landmarks.part(n).x
[143]         y = landmarks.part(n).y
[144]         cv2.circle(frame, (x, y), 1, (230, 159, 23), -1)
[145]
[146]     ## "Mouth"
[147]     for n in range(48, 68):
[148]         x = landmarks.part(n).x
[149]         y = landmarks.part(n).y
[150]         cv2.circle(frame, (x, y), 1, (79, 76, 240), -1)
[151]         #cv2.putText(frame, str(n), (x, y), cv2.FONT_HERSHEY_SIMPLEX, 0.35, (0, 0,
255), 2, 4)
[152]
[153]     face_width = x2 - x1
[154]     face_height = y2 - y1
[155]
[156]     # 표정인식의 근거로 임의 n포인트의 위치관계 분석
[157]     mouth_width = (landmarks.part(54).x - landmarks.part(48).x) / face_width # 입을
벌린 정도

```

```

[158]     mouth_height = (landmarks.part(66).y - landmarks.part(62).y) / face_width # 입을
        벌린 정도
[159]
[160]     # calc center pos
[161]     face_x = face_width / 2 + x1
[162]     face_y = face_height / 2 + y1
[163]     face_pos = [face_x, face_y, face_width, face_height, x1, y1]
[164]
[165]     mouth_x = (landmarks.part(54).x - landmarks.part(48).x) / 2 + landmarks.part(48).x
[166]     mouth_y = (landmarks.part(66).y - landmarks.part(62).y) / 2 + landmarks.part(62).y
[167]     mouth_pos = [mouth_x, mouth_y, landmarks.part(54).x - landmarks.part(48).x,
        landmarks.part(66).y - landmarks.part(62).y, landmarks.part(48).x,
        landmarks.part(50).y]
[168]
[169]     nose_x = (landmarks.part(34).x - landmarks.part(31).x) / 2 + landmarks.part(31).x
[170]     nose_y = (landmarks.part(33).y - landmarks.part(27).y) / 2 + landmarks.part(27).y
[171]     nose_pos = [mouth_x, mouth_y, landmarks.part(34).x - landmarks.part(31).x,
        landmarks.part(33).y - landmarks.part(27).y, landmarks.part(31).x,
        landmarks.part(27).y]
[172]
[173]     eye_x1 = (landmarks.part(39).x - landmarks.part(36).x) / 2 + landmarks.part(36).x
[174]     eye_y1 = (landmarks.part(41).y - landmarks.part(37).y) / 2 + landmarks.part(37).y
[175]     eye_pos1 = [eye_x1, eye_y1, landmarks.part(39).x - landmarks.part(36).x,
        landmarks.part(41).y - landmarks.part(37).y, landmarks.part(36).x,
        landmarks.part(37).y]
[176]
[177]     eye_x2 = (landmarks.part(45).x - landmarks.part(42).x) / 2 + landmarks.part(42).x
[178]     eye_y2 = (landmarks.part(47).y - landmarks.part(43).y) / 2 + landmarks.part(43).y
[179]     eye_pos2 = [eye_x2, eye_y2, landmarks.part(45).x - landmarks.part(42).x,
        landmarks.part(47).y - landmarks.part(43).y, landmarks.part(42).x,
        landmarks.part(43).y]
[180]
[181]     # 눈썹 위의 10가지 특징으로 눈썹을 고르는 정도와 눈살을 찌푸리는 정도를
        분석
[182]     brow_sum = 0 # 높이의 합
[183]     frown_sum = 0 # 양쪽 눈썹 거리의 합
[184]     for j in range(17, 21):
[185]         brow_sum += (landmarks.part(j).y - y1) + (landmarks.part(j + 5).y - y1)

```

```

[186]     frown_sum += landmarks.part(j + 5).x - landmarks.part(j).x
[187]     line_brow_x.append(landmarks.part(j).x)
[188]     line_brow_y.append(landmarks.part(j).y)
[189]
[190]     temp_x = np.array(line_brow_x)
[191]     temp_y = np.array(line_brow_y)
[192]     z1 = np.polyfit(temp_x, temp_y, 1) # 일차직선 합성
[193]     brow_k = -round(z1[0], 3) # 합 곡선의 기울기와 실제 눈썹의 기울기 방향은
반대
[194]
[195]     eye_sum = (landmarks.part(41).y - landmarks.part(37).y + landmarks.part(40).y -
landmarks.part(38).y +
[196]     landmarks.part(47).y - landmarks.part(43).y + landmarks.part(46).y -
landmarks.part(44).y)
[197]     eye_height = (eye_sum / 4) / face_width
[198]
[199]     return frame, face_pos, eye_pos1, eye_pos2, nose_pos, mouth_pos, mouth_width,
mouth_height, eye_height, brow_k
[200]     return frame, [], [], [], [], [], -1, -1, -1, -1
[201]
[202]     상술한 다양한 코딩 블록 이외에도, 코딩 로봇(100)은 다양한 코딩 블록을
포함할 수 있다. 예를 들어, 물체 인식 블록은 약 32개의 물체를 인식할 수 있는
코딩 명령어가 저장된 것으로 Tensorflow 기반으로 동작할 수 있다. 또는, 표지판
인식 블록은 사용자가 직접 표지판을 학습시키기 위한 코딩 블록 및 학습한
데이터를 기반으로 표지판을 인식하는 블록으로 구성될 수 있다. 또는, 차선
인식 블록은 화면에서 연속되는 두개의 선을 인식후, 선과 자동차 사이 거리를
측정해 한쪽으로 치우쳐져 있다면 차선 한가운데로 이동하여 이탈을 막기 위한
코딩 블록이다. 또는, 이외 0에서 9까지의 숫자를 인식하는 숫자 인식 블록, 영어
알파벳을 인식하는 글자 인식 블록이 본 발명의 다양한 실시예에 따라 응용될 수
있음은 물론이다.
[203]     이후, 코딩 로봇(100)은, 판단된 감정을 바탕으로 제어 데이터를 획득할 수
있다.
[204]     한편, 코딩 로봇(100)은 상술한 실시예에 의해 판단된 감정에 대응되는 제어
명령에 따라 제어될 수 있다.
[205]     일 실시예로, 코딩 로봇(100)은, 판단된 감정에 대응되는 2차원 벡터 공간상의
좌표 정보를 획득할 수 있다. 이후, 코딩 로봇(100)은, 벡터 공간상의 좌표 정보에
대응되는 제어 명령을 획득할 수 있다.
[206]     예를 들어, 2차원 벡터 공간은 도 5에 도시된 2차원 공간일 수 있다. 이때, 2차원

```

공간은 J. Russel의 원형 감정 모델 상에 표시될 수 있음은 물론이다. J. Russel의 원형 감정 모델의 가로축은 감정이 긍정적인지 부정적인지 여부를 나타내며, 세로축은 활동성을 나타낸다. 즉, 코딩 로봇(100)이 판단한 인식 대상자의 감정은 도 5에 도시된 J. Russel의 원형 감정 모델 상의 하나의 좌표 정보로 획득될 수 있다.

[207] 코딩 로봇(100)은 J. Russel의 원형 감정 모델 상의 각각의 좌표 정보에 대응되는 제어 명령을 저장할 수 있다. 일 실시예에 따라, 획득된 좌표 정보가 제1 사분면에 존재하면, 코딩 로봇(100)은 제1 제어 명령을 수행하고, 제2사분면에 존재하면, 코딩 로봇(100)은 제2 제어 명령을 수행하고, 제3사분면에 존재하면, 코딩 로봇(100)은 제3 제어 명령을 수행하고, 제4사분면에 존재하면, 코딩 로봇(100)은 제4 제어 명령을 수행할 수 있다.

[208] 일 실시예로, 제1 제어 명령 내지 제4 제어 명령은 코딩 로봇(100)의 이동과 관련된 명령일 수 있다.

[209] 한편, 본 발명의 또 다른 실시예에 따라, 코딩 로봇(100)은, 인식 대상자의 얼굴 촬영 이미지를 바탕으로 인식 대상자의 감정 변화 정보를 획득할 수 있다.

[210] 이후, 코딩 로봇(100)은, 인식 대상자의 감정이 제1 감정에서 제2 감정으로 변경된 경우, 제1 감정에 대응되는 제1 좌표 정보 및 제2 감정에 대응되는 제2 좌표 정보를 획득할 수 있다.

[211] 이후, 코딩 로봇(100)은, 제1 좌표 및 제2 좌표에 대응되는 벡터값에 대응되는 이동 명령을 제어 명령으로 획득할 수 있다.

[212] 예를 들어, 도 5에 도시된 바와 같이, 인식 대상자의 감정이 제1 좌표(510)에서 제2 좌표(520)으로 변경된 경우, 코딩 로봇(100)은 제1 좌표(510)를 시작점으로 하고 제2 좌표(520)를 종점으로 하는 벡터 방향으로 이동할 수 있다.

[213] 일 실시예에 따라, 코딩 로봇(100)의 초기 이동 속도는 원점부터 제1 좌표(510)까지의 거리에 비례하고, 코딩 로봇(100)의 이동 가속도는 제1 좌표(510)에서 제2 좌표(520)까지의 거리에 비례할 수 있다. 예를 들어, 제1 좌표(510)이 (a,b)이고, 제2 좌표가 (c,d)인 경우, 초기 이동 속도 V 및 이동 가속도 A는 하기 수식과 같이 표시될 수 있다.

[214] [수식1]

$$V=k\sqrt{a^2+b^2}$$

[215]  $A=l\sqrt{(a-c)^2+(b-d)^2}$

[216] 이때, k 및 l은 임의의 양의 상수일 수 있다.

[217] 한편, 본 발명의 또 다른 실시예에 따라, 제어 데이터가 인식 대상자의 음성 데이터인 경우, 코딩 로봇(100)은 다양한 방법을 통해 제어 데이터를 획득할 수 있다.

[218] 구체적으로, 코딩 로봇(100)은, 음성 데이터를 바탕으로 코딩 로봇에 대한 동작 속도 제어 명령 및 동작 반복수 제어 명령을 포함하는 동작 제어 블록을 생성할

수 있다.

- [219] 이후, 코딩 로봇(100)은, 음성 데이터를 바탕으로 동작 속도 정보 및 동작 반복수 정보 중 적어도 하나를 획득할 수 있다.
- [220] 이후, 코딩 로봇(100)은, 동작 속도 정보 및 동작 반복수 정보 중 적어도 하나를 바탕으로 코딩 로봇의 동작 속도 및 동작 반복수 중 적어도 하나에 대응되는 제어 명령을 획득할 수 있다.
- [221] 이때, 코딩 로봇(100)은, 음성 데이터가 기 설정된 제어 정보와 일치하는 경우, 기 설정된 제어 정보에 대응되는 동작 속도 정보 또는 동작 반복수 정보를 바탕으로 코딩 로봇의 동작 속도 또는 동작 반복수를 제어 명령으로 획득할 수 있다.
- [222] 예를 들어, 음성 데이터가 "오른쪽으로 빠르게 움직여"이고, 이에 대응되는 기 저장된 제어 명령(예를 들어, 오른쪽으로 3단계 속도만큼 이동)이 존재하면, 코딩 로봇(100)은 기 저장된 제어 명령에 따라 제어 될 수 있다.
- [223] 그러나, 코딩 로봇(100)은, 음성 데이터가 기 설정된 제어 정보와 일치하지 않는 경우, 음성 데이터를 바탕으로 음절 간격 정보를 생성하고, 음절 간격 정보에 대응하여 동작 속도 정보를 생성할 수 있다.
- [224] 예를 들어, 음성 데이터가 "오른쪽으로 빠르게 움직여"이고, "빠르게"에 대응되는 기 설정된 제어 정보가 존재하지 않는 경우, 코딩 로봇(100)은, 음성 데이터에 기초하여 음절 간격 정보를 생성하고, 음절 간격 정보에 대응하여 동작 속도 정보를 생성하고, 생성된 동작 속도 정보에 대응하는 이동 속도 제어 명령문을 생성할 수 있다.
- [225] 구체적으로, 코딩 로봇(100)은 음성 데이터에서 동작 속도 정보가 검색되지 않으면, 음성 데이터에 기초하여 음절 간격 정보를 생성하고, 음절 간격 정보에 대응하여 동작 속도 정보를 생성하고, 생성된 동작 속도 정보에 대응하는 이동 속도 제어 명령문을 생성할 수 있다.
- [226] 구체적으로, 코딩 로봇(100)은 음성 데이터에 포함된 음절 중에서 어느 하나의 음절의 종료 시점과 어느 하나의 음절의 다음 음절의 시작 시점 간에 제1 시간차를 산출하고, 복수의 제1 시간차의 평균 시간차를 음절 간격 정보로 생성할 수 있다.
- [227] 이후, 코딩 로봇(100)은 기준 시간차 대비 음절 간격 정보가 나타내는 평균 시간차의 시간차 비율을 산출하고, 기본 이동 속도에 시간차 비율을 적용하여 동작 속도 정보로 생성할 수 있다.
- [228] 예를 들어, 코딩 로봇(100)은 평균 시간차가 "12msec"이고, 기준 시간차가 "10msec"이며, 기본 이동 속도가 "1배속"인 경우, 시간차 비율을 "120%"로 산출하고, 기본 이동 속도 "1배속"에 시간차 비율 "120%"를 적용하여 동작 속도 정보를 "1.2배속"으로 생성할 수 있다.
- [229] 한편, 코딩 로봇(100)은, 음성 데이터가 기 설정된 제어 정보와 일치하지 않는 경우, 음성 데이터를 바탕으로 신호 세기 정보를 생성하고, 신호 세기 정보에

- 대응하여 동작 반복수 정보를 생성할 수 있다.
- [230] 구체적으로, 음성 데이터에 동작 반복수 정보가 포함되지 않은 경우, 코딩 로봇(100)은 음성 신호에 기초하여 신호 세기 정보를 생성하고, 신호 세기 정보에 대응하여 동작 반복수 정보를 생성하고, 생성된 동작 반복수 정보에 대응하는 동작 반복수 제어 명령문을 생성할 수 있다.
- [231] 예를 들어, 코딩 로봇(100)은 음성 데이터에서 동작 반복수 정보 검색되지 않으면, 음성 신호에 기초하여 신호 세기 정보를 생성하고, 신호 세기 정보에 대응하여 동작 반복수 정보를 생성하고, 생성된 동작 반복수 정보에 대응하는 동작 반복수 제어 명령문을 생성할 수 있다.
- [232] 구체적으로, 코딩 로봇(100)은 음성 신호의 시간에 따른 신호 세기 중에서 최대 신호 세기를 신호 세기 정보로 생성할 수 있다.
- [233] 이후, 코딩 로봇(100)은 기준 신호 세기 대비 신호 세기 정보가 나타내는 최대 신호 세기의 신호 세기 비율을 산출하고, 기본 동작 반복수에 신호 세기 비율을 적용하여 동작 반복수 정보로 생성할 수 있다.
- [234] 예를 들어, 코딩 로봇(100)은 최대 신호 세기가 "200DB"이고, 기준 신호 세기가 "100DB"이며, 기본 동작 반복수가 "1회"인 경우, 신호 세기 비율을 "200%"로 산출하고, 기본 동작 반복수 "1회"에 신호 세기 비율 "200%"를 적용하여 동작 반복수 정보를 "2회"로 생성할 수 있다. 이때, 동작 반복수 정보는 반올림되어 자연수로 생성될 수 있다.
- [235] 생성된 동작 속도 정보 또는 동작 반복수를 바탕으로 코딩 로봇의 동작 속도 또는 동작 반복수를 제어 명령으로 획득할 수 있다.
- [236] 한편, 본 발명의 또 다른 실시예에 따라, 제어 명령이, 코딩 로봇을 이동시키기 위한 제어 명령인 경우, 코딩 로봇(100)은, 코딩 로봇 주변의 물체를 인식할 수 있다. 이후, 코딩 로봇(100)은, 인식된 물체가 기 저장된 표지판과 일치하는 경우, 기 저장된 표지판에 대응되는 제어 정보를 획득할 수 있다. 코딩 로봇(100)은, 코딩 로봇 주변의 차선을 인식할 수 있다. 코딩 로봇(100)은, 인식된 차선을 따라 이동하되, 제어 정보에 대응되게 이동할 수 있다.
- [237] 이때, 코딩 로봇(100)은, 코딩 로봇(100)으로 입력되는 영상을 바탕으로 연속된 두개의 선분 정보를 획득하고, 획득된 선분 정보를 바탕으로 차선 영역을 획득할 수 있다.
- [238] 이후, 코딩 로봇(100)은, 획득된 선분과 코딩 로봇의 거리 정보를 획득할 수 있다. 이후, 코딩 로봇(100)은, 차선 영역 및 거리 정보를 바탕으로 코딩 로봇을 제어할 수 있다.
- [239] 이를 위해, 상술한 물체 인식 블록, 표지판 인식 블록, 차선 인식 블록, 숫자 인식 블록 및 글자 인식 블록 중 적어도 하나의 블록을 이용할 수 있음은 물론이다.
- [240] 한편, 본 발명의 구성 요소들은 하드웨어인 컴퓨터와 결합되어 실행되기 위해 프로그램(또는 애플리케이션)으로 구현되어 매체에 저장될 수 있다. 본 발명의 구성 요소들은 소프트웨어 프로그래밍 또는 소프트웨어 요소들로 실행될 수

있으며, 이와 유사하게, 실시 예는 데이터 구조, 프로세스들, 루틴들 또는 다른 프로그래밍 구성들의 조합으로 구현되는 다양한 알고리즘을 포함하여, C, C++, 자바(Java), 어셈블리(assembler) 등과 같은 프로그래밍 또는 스크립팅 언어로 구현될 수 있다. 기능적인 측면들은 하나 이상의 프로세서들에서 실행되는 알고리즘으로 구현될 수 있다.

- [241] 이상, 첨부된 도면을 참조로 하여 본 발명의 실시예를 설명하였지만, 본 발명이 속하는 기술분야의 통상의 기술자는 본 발명이 그 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로, 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며, 제한적이 아닌 것으로 이해해야만 한다.

## 청구범위

- [청구항 1] 코딩 로봇의 제어 방법에 있어서,  
 상기 코딩 로봇이, 인식 대상자의 정보를 획득하는 단계;  
 상기 코딩 로봇이, 상기 획득된 인식 대상자의 정보를 바탕으로 제어 데이터를 획득하는 단계;  
 상기 코딩 로봇이, 상기 제어 데이터가 기 설정된 제어 정보와 일치하는지 여부를 판단하는 단계;  
 상기 코딩 로봇이, 상기 제어 데이터가 기 설정된 제어 정보와 일치하는 경우, 상기 기 설정된 제어 명령에 대응되는 동작을 수행하는 단계; 및  
 상기 코딩 로봇이, 상기 제어 데이터가 기 설정된 제어 정보와 일치하지 않는 경우, 상기 제어 데이터를 분석하여, 기 설정된 제어 명령 중 어느 하나와 매칭 시키는 단계;를 포함하고,  
 상기 제어 데이터는, 상기 인식 대상자의 감정 데이터 및 음성 데이터 중 적어도 하나의 데이터인 것을 특징으로 하는 제어 방법.
- [청구항 2] 제1항에 있어서,  
 상기 제어 데이터가 상기 인식 대상자의 감정 데이터인 경우,  
 상기 제어 데이터를 획득하는 단계는,  
 상기 코딩 로봇이, 상기 인식 대상자의 얼굴 촬영 이미지를 획득하는 단계;  
 상기 코딩 로봇이, 상기 얼굴 촬영 이미지를 바탕으로 모델 변수를 획득하는 단계;  
 상기 코딩 로봇이, 상기 획득된 모델 변수를 바탕으로 상기 인식 대상자가 기 등록된 사용자의 얼굴과 일치하는지 판단하는 단계;  
 상기 인식 대상자가, 상기 기 등록된 사용자와 일치하는 경우, 상기 코딩 로봇이, 상기 얼굴 촬영 이미지를 바탕으로 상기 얼굴 촬영 이미지에 대응되는 특징값을 획득하는 단계;  
 상기 코딩 로봇이, 상기 획득된 특징값을 바탕으로 상기 인식 대상자의 눈, 코, 입을 포함하는 얼굴 부위를 판단하는 단계;  
 상기 코딩 로봇이, 상기 판단된 얼굴 부위를 바탕으로 상기 인식 대상자의 감정을 판단하는 단계; 및  
 상기 코딩 로봇이, 상기 판단된 감정을 바탕으로 상기 제어 데이터를 획득하는 단계;를 포함하는 제어 방법.
- [청구항 3] 제2항에 있어서,  
 상기 제어 명령에 대응되는 동작을 수행하는 단계는,  
 상기 코딩 로봇이, 상기 판단된 감정에 대응되는 2차원 벡터 공간상의 좌표 정보를 획득하는 단계;  
 상기 벡터 공간상의 좌표 정보에 대응되는 제어 명령을 획득하는 단계; 및

상기 제어 명령에 대응되는 동작을 수행하는 단계; 를 포함하는 제어 방법.

[청구항 4]

제2항에 있어서,  
상기 제어 명령에 대응되는 동작을 수행하는 단계는,  
상기 코딩 로봇이, 상기 인식 대상자의 얼굴 촬영 이미지를 바탕으로 상기 인식 대상자의 감정 변화 정보를 획득하는 단계;  
상기 코딩 로봇이, 상기 인식 대상자의 감정이 제1 감정에서 제2 감정으로 변경된 경우, 상기 제1 감정에 대응되는 제1 좌표 정보 및 상기 제2 감정에 대응되는 제2 좌표 정보를 획득하는 단계; 및  
상기 코딩 로봇이, 상기 제1 좌표 및 상기 제2 좌표에 대응되는 벡터값에 대응되는 이동 명령을 상기 제어 명령으로 획득하는 단계;를 포함하는 제어 방법.

[청구항 5]

제1항에 있어서,  
상기 제어 명령이, 상기 코딩 로봇을 이동시키기 위한 제어 명령인 경우,  
상기 제어 방법은,  
상기 코딩 로봇이, 상기 코딩 로봇 주변의 물체를 인식하는 단계;  
상기 코딩 로봇이, 상기 인식된 물체가 기 저장된 표지판과 일치하는 경우, 상기 기 저장된 표지판에 대응되는 제어 정보를 획득하는 단계;  
상기 코딩 로봇이, 상기 코딩 로봇 주변의 차선을 인식하는 단계; 및  
상기 코딩 로봇이, 상기 인식된 차선을 따라 이동하되, 상기 제어 정보에 대응되게 이동하는 단계; 를 포함하고,  
상기 차선을 인식하는 단계는,  
상기 코딩 로봇으로 입력되는 영상을 바탕으로 연속된 두개의 선분 정보를 획득하고, 획득된 선분 정보를 바탕으로 차선 영역을 획득하는 단계;  
상기 획득된 선분과 상기 코딩 로봇의 거리 정보를 획득하는 단계; 및  
상기 차선 영역 및 상기 거리 정보를 바탕으로 상기 코딩 로봇을 제어하는 단계; 를 포함하는 제어 방법.

[청구항 6]

제1항에 있어서,  
상기 제어 데이터가 상기 인식 대상자의 음성 데이터인 경우, 상기 제어 방법은,  
상기 코딩 로봇이, 상기 음성 데이터를 바탕으로 상기 코딩 로봇에 대한 동작 속도 제어 명령 및 동작 반복수 제어 명령을 포함하는 동작 제어 블록을 생성하는 단계;  
상기 코딩 로봇이, 상기 음성 데이터를 바탕으로 동작 속도 정보 및 동작 반복수 정보 중 적어도 하나를 획득하는 단계; 및  
상기 코딩 로봇이, 동작 속도 정보 및 동작 반복수 정보 중 적어도 하나를 바탕으로 상기 코딩 로봇의 동작 속도 및 동작 반복수 중 적어도 하나에

대응되는 제어 명령을 획득하는 단계; 를 포함하는 제어 방법.  
 [청구항 7] 제6항에 있어서,  
 상기 코딩 로봇의 동작 속도 및 동작 반복수 중 적어도 하나에 대응되는  
 제어 명령을 획득하는 단계는,  
 상기 코딩 로봇이, 상기 음성 데이터가 기 설정된 제어 정보와 일치하는  
 경우, 상기 기 설정된 제어 정보에 대응되는 동작 속도 정보 또는 동작  
 반복수 정보를 바탕으로 상기 코딩 로봇의 동작 속도 또는 동작 반복수를  
 상기 제어 명령으로 획득하는 단계;  
 상기 코딩 로봇이, 상기 음성 데이터가 기 설정된 제어 정보와 일치하지  
 않는 경우, 상기 음성 데이터를 바탕으로 음절 간격 정보를 생성하고,  
 상기 음절 간격 정보에 대응하여 상기 동작 속도 정보를 생성하는 단계;  
 상기 코딩 로봇이, 상기 음성 데이터가 기 설정된 제어 정보와 일치하지  
 않는 경우, 상기 음성 데이터를 바탕으로 신호 세기 정보를 생성하고,  
 상기 신호 세기 정보에 대응하여 상기 동작 반복수 정보를 생성하는 단계;  
 및  
 상기 생성된 동작 속도 정보 또는 동작 반복수를 바탕으로 상기 코딩  
 로봇의 동작 속도 또는 동작 반복수를 상기 제어 명령으로 획득하는  
 단계;를 포함하는 제어 방법.