

(19) 대한민국특허청(KR)  
(12) 공개특허공보(A)(11) 공개번호 10-2023-0174528  
(43) 공개일자 2023년12월28일

(51) 국제특허분류(Int. Cl.)

G06V 20/58 (2022.01) G06V 10/764 (2022.01)

G06V 10/774 (2022.01) G06V 10/94 (2022.01)

G06V 20/70 (2022.01)

(52) CPC특허분류

G06V 20/582 (2022.01)

G06V 10/764 (2023.08)

(21) 출원번호 10-2022-0075577

(22) 출원일자 2022년06월21일

심사청구일자 없음

(71) 출원인

현대모비스 주식회사

서울특별시 강남구 테헤란로 203 (역삼동)

(72) 발명자

이재영

경기도 이천시 증신로325번길 39(송정동, 이천 라온프라이빗) 103동 1101호

(74) 대리인

특허법인 플러스

전체 청구항 수 : 총 9 항

(54) 발명의 명칭 임베딩 기반 객체 클래스 분류 시스템 및 그 방법

## (57) 요약

본 발명은 종래 기술보다 적은 메모리 사용량과 연산량을 사용하여 분류 네트워크 구현이 가능하여, 복잡한 클래스 정보를 갖는 분류 네트워크를 임베딩 시스템에 적용할 수 있는 임베딩 기반 객체 클래스 분류 시스템 및 그 방법에 관한 것이다.

대표도 - 도1



(52) CPC특허분류

*G06V 10/774* (2023.08)

*G06V 10/94* (2023.08)

*G06V 20/70* (2023.08)

---

## 명세서

### 청구범위

#### 청구항 1

기저장된 분류 네트워크에 객체의 클래스 정보가 라벨링된 학습 데이터 셋을 입력하여, 학습을 수행하는 제1 학습 처리부;

상기 제1 학습 처리부의 학습 결과를 기반으로 분류 네트워크를 구성하고, 상기 학습 데이터 셋을 입력하여, 학습을 수행하는 제2 학습 처리부; 및

상기 제2 학습 처리부에 의해 최종 학습 처리된 분류 네트워크를 이용하여, 입력되는 영상 데이터에 포함되어 있는 객체의 클래스 정보를 분류하여 출력하는 추론 처리부;

를 포함하는, 임베딩 기반 객체 클래스 분류 시스템.

#### 청구항 2

제1항에 있어서,

상기 제1 학습 처리부는

다수의 컨볼루션 레이어와 다수의 풀링 레이어로 구성되어, 상기 학습 데이터 셋의 특징(feature)을 추출하는 특징 추출부;

적어도 2 개의 FC(Fully-Connected) 레이어로 구성되어, 추출한 특징의 분류를 결정하는 분류 처리부; 및

기설정된 활성화 함수 레이어로 구성되어, 결정한 분류를 출력값으로 출력하는 출력 함수부;

로 분류 네트워크를 구성하되,

기설정된 손실함수와 최적화 기법을 이용하여, 출력값을 기반으로, 상기 특징 추출부와 분류 처리부의 레이어에 대한 가중치(weight)를 업데이트 설정하는, 임베딩 기반 객체 클래스 분류 시스템.

#### 청구항 3

제2항에 있어서,

상기 제2 학습 처리부는

다수의 컨볼루션 레이어와 다수의 풀링 레이어로 구성되어, 상기 학습 데이터 셋의 특징을 추출하는 특징 추출부;

적어도 2 개의 FC 레이어로 구성되어, 추출한 특징의 분류를 결정하는 분류 처리부;

기설정된 활성화 함수 레이어로 구성되어, 결정한 분류를 출력값으로 출력하는 출력 함수부; 및

적어도 하나의 임베딩 레이어로 구성되어, 상기 학습 데이터 셋을 입력받아 기설정된 차원의 실수 파라미터로 변환하는 임베딩 처리부;

로 분류 네트워크를 구성하되,

상기 특징 추출부의 레이어는

상기 제1 학습 처리부의 특징 추출부에서 최종 업데이트 설정된 가중치를 적용하고,

기설정된 손실함수와 최적화 기법을 이용하여, 상기 분류 처리부와 임베딩 처리부의 레이어에 대한 가중치를 업데이트 설정하는, 임베딩 기반 객체 클래스 분류 시스템.

#### 청구항 4

제3항에 있어서,

상기 제2 학습 처리부의 분류 처리부는

상기 제1 학습 처리부의 분류 처리부보다 적은 차원의 레이어를 구성하는, 임베딩 기반 객체 클래스 분류 시스템.

#### 청구항 5

제3항에 있어서,

상기 추론 처리부는

클래스를 분류하고자 하는 객체가 인식된 영상 데이터가 입력되는 입력부;

상기 제2 학습 처리부에 의해 최종 학습 처리된 분류 네트워크로, 상기 입력부에 의한 영상 데이터를 입력하여, 예측되는 클래스가 출력되는 출력부;

상기 제2 학습 처리부에 의해 최종 학습 처리된 상기 임베딩 처리부의 가중치 값에 상기 데이터 출력부에 의한 출력값을 맵핑 분석하는 맵핑부; 및

상기 맵핑부에 의한 맵핑 분석 결과를 이용하여, 해당하는 객체의 최종 클래스를 분류 결정하여 출력하는 추론부;

를 포함하는, 임베딩 기반 객체 클래스 분류 시스템.

#### 청구항 6

연산 처리 수단에 의해 각 단계가 수행되는 임베딩 기반 객체 클래스 분류 시스템을 이용한 임베딩 기반 객체 클래스 분류 방법으로서,

분류 네트워크에 객체의 클래스 정보가 라벨링된 학습 데이터 셋을 입력받아 학습을 수행하는 제1 학습 단계(S100);

상기 제1 학습 단계(S100)에 의한 학습 결과를 기반으로 분류 네트워크를 구성하고, 상기 학습 데이터 셋을 입력받아 학습을 수행하는 제2 학습 단계(S200); 및

상기 제2 학습 단계(S200)에 의해 최종 학습 처리된 분류 네트워크를 이용하여, 외부로부터 입력되는 클래스를 분류하고자 하는 객체가 인식된 영상 데이터에 포함되어 있는 객체의 클래스 정보를 분류하여 출력하는 추론 처리 단계(S300);

를 포함하는, 임베딩 기반 객체 클래스 분류 방법.

#### 청구항 7

제6항에 있어서,

상기 제2 학습 단계(S200)는

상기 제1 학습 단계(S100)에 의해 최종 학습 처리된 분류 네트워크를 구성하는 다수의 컨볼루션 레이어와 다수의 풀링 레이어의 가중치를 적용하여, 분류 네트워크를 구성하고,

적어도 하나의 임베딩 레이어를 포함하여, 상기 임베딩 레이어에 상기 학습 데이터 셋을 입력하여 기설정된 차원의 실수 파라미터로 변환 출력받는, 임베딩 기반 객체 클래스 분류 방법.

## 청구항 8

제7항에 있어서,

상기 제2 학습 단계(S200)의 분류 네트워크를 구성하는 FC(Fully-Connected) 레이어는

상기 제1 학습 단계(S100)의 분류 네트워크를 구성하는 FC 레이어보다 적은 차원의 레이어로 구성하는, 임베딩 기반 객체 클래스 분류 방법.

## 청구항 9

제7항에 있어서,

상기 추론 처리 단계(S300)는

상기 제2 학습 단계(S200)에 의해 최종 학습 처리된 분류 네트워크로부터, 상기 영상 데이터에 의한 예측되는 클래스를 출력받아,

최종 학습 처리된 임베딩 레이어의 가중치 값에 출력한 예측 클래스를 맵핑 분석하여, 해당하는 객체의 최종 클래스를 결정하여 출력하는, 임베딩 기반 객체 클래스 분류 방법.

## 발명의 설명

### 기술 분야

[0001] 본 발명은 임베딩 기반 객체 클래스 분류 시스템 및 그 방법에 관한 것으로서, 보다 상세하게는, 영상 데이터에 포함되어 있는 객체 영역을 인식한 후, 해당하는 객체의 클래스를 분류하면서도, 메모리 사용량과 연산량이 제한된 임베디드 시스템에서 구현 가능하도록 설계된 임베딩 기반 객체 클래스 분류 시스템 및 그 방법에 관한 것이다.

### 배경 기술

[0003] 교통 표지판은 교통에 필요한 주의, 규제 및 지시 등을 표시하는 알림판이다. 도로 상황은 경우에 따라 변하므로, 자율 주행 차량이 도로 규칙을 지키기 위해서는, 반드시 표지판 인식이 필요하다.

[0004] 입력되는 영상 데이터 내에 포함되어 있는 표지판을 인식하기 위해서는, 첫번째 단계로, 표지판 영역을 찾은 후, 두번째 단계로, 찾은 표지판 영역에 해당하는 표지판의 분류를 수행해야 한다.

[0005] 최근 딥러닝 기술이 발전됨에 따라, 객체 인식 성능이 향상되고, AP(Application Processor) 등에 NPU(Neural Processing Unit) 등이 탑재되면서, 전방 카메라에 딥러닝 네트워크 적용이 확대되고 있다.

[0006] 이러한 딥러닝 네트워크도 교통 표지판을 인식하기 위하여, 객체 탐지 네트워크(object detection network)를 사용하여, 교통 표지판의 후보 영역(bonding box)를 찾은 후, 분류 네트워크(classification network)를 사용하여 탐지한 교통 표지판이 어느 의미에 해당하는 교통 표지판인지 분류하게 된다.

[0007] 그렇지만, 이러한 교통 표지판은 컴퓨터로 영상을 생성하기 때문에, 인식하기는 용이하지만, 너무 많은 종류가 있기 때문에, 분류 네트워크를 임베디드 시스템에 구현하는 것이 어려운 문제점으로 남아있다.

[0008] 상세하게는, 딥러닝을 지원하는 임베디드 시스템이라도 작은 캐시 메모리(cache memory) 용량과 실시간 처리에 대한 제약 조건으로 인해, 네트워크의 가중치 값과 연산량이 제한된다.

[0009] 일 예를 들자면, TI 사의 'TDA4V-MID 프로세서'는 8 MB의 캐시 메모리를 제공하지만, 다수의 네트워크를 병렬로 동작 시켜야 하므로 단일 네트워크가 사용할 수 있는 메모리 크기는 2 MB 정도이다. 특히, 표지판 인식을 위해서는, 객체 탐지 네트워크로 교통 표지판의 후보 영역을 추출하고, 분류 네트워크로 세부 분류 동작을 수행해야

하므로, 제한된 메모리에 두 네트워크가 동작하게 된다. 뿐만 아니라, 분류 네트워크는 후보 영역의 수만큼 동작이 반복되게 되고, 보통 교통 표지판의 수가 300개 이상이므로, 분류 네트워크 종단에 구성되는 두 개의 FC 레이어가 사용하는 메모리 크기는 0.7 MB(300\*300\*4B\*2)가 된다.

[0010] 이와 같이, 두 개의 레이어만으로 35%에 해당하는 메모리를 소모하게 되므로, 표지판 인식을 위한 네트워크(edge network)를 임베디드 시스템에 구현하는 것이 매우 어렵다

[0011] 또한, 두 개의 레이어에서 소모하는 연산량이 175 kFlops이므로, 다수의 후보 영역이 추출될 경우, 실시간 처리 조건을 만족할 수 없는 문제점이 있다.

[0013] 물론, 자율 주행 레벨 2에서는, 본격적인 자율 주행 제어가 아닌 반드시 운전자의 제어가 개입되기 때문에, 교통 표지판 중 극히 일부인 속도 표지판 만을 인식 및 분류하더라도, 운전자에게 자율 주행 제어를 통한 큰 도움을 제공할 수 있다.

[0014] 그렇지만, 자율 주행 레벨이 3 이상일 될 경우, 주행 과정에서의 운전자의 개입이 이루어지지 않기 때문에, 단순한 형태의 속도 표지판뿐 만 아니라, 도로 형태가 변형될 가능성이 높은 공사장 표지판 등 도로 상 위치하고 있는 교통 표지판, 대부분을 인식 및 분류해야 하지만, 이를 차량 내 임베디드 시스템에 구현하는 것이 어렵기 때문에, 실질적으로 자율 주행 레벨이 증가하는 데 하나의 문제점으로 작용되고 있음이 분명하다.

[0016] 한국 공개특허공보 제10-2020-0003349호("교통 표지판 인식 시스템 및 방법")에서는 프로세서에 대한 계산 부하를 최소화하는 기술을 이용한 교통 표지판 인식 시스템 및 방법을 제공하고 있다.

## 선행기술문헌

### 특허문헌

[0018] (특허문헌 0001) 한국 공개특허공보 제10-2020-0003349호 (공개일 2020.01.09.)

## 발명의 내용

### 해결하려는 과제

[0019] 본 발명은 상기한 바와 같은 문제점을 해결하기 위해 안출된 것으로써, 객체 영역 자체의 인식은 용이하지만, 수백 종류의 다양한 클래스를 가지고 있어 메모리 사용량과 연산량이 제한된 임베디드 시스템에서의 구현에 제약 조건이 존재하는 객체 분류 네트워크를 구현 가능하도록 설계한 임베딩 기반 객체 클래스 분류 시스템 및 그 방법을 제공함에 있다.

### 과제의 해결 수단

[0021] 상기한 바와 같은 문제점을 해결하기 위한 본 발명에 의한 임베딩 기반 객체 클래스 분류 시스템은, 기저장된 분류 네트워크에 객체의 클래스 정보가 라벨링된 학습 데이터 셋을 입력하여, 학습을 수행하는 제1 학습 처리부, 상기 제1 학습 처리부의 학습 결과를 기반으로 분류 네트워크를 구성하고, 상기 학습 데이터 셋을 입력하여, 학습을 수행하는 제2 학습 처리부 및 상기 제2 학습 처리부에 의해 최종 학습 처리된 분류 네트워크를 이용하여, 입력되는 영상 데이터에 포함되어 있는 객체의 클래스 정보를 분류하여 출력하는 추론 처리부를 포함하는 것이 바람직하다.

[0022] 더 나아가, 상기 제1 학습 처리부는 다수의 컨볼루션 레이어와 다수의 풀링 레이어로 구성되어, 상기 학습 데이터 셋의 특징(feature)을 추출하는 특징 추출부, 적어도 2 개의 FC(Fully-Connected) 레이어로 구성되어, 추출한 특징의 분류를 결정하는 분류 처리부 및 기설정된 활성화 함수 레이어로 구성되어, 결정된 분류를 출력값으로 출력하는 출력 함수부로 분류 네트워크를 구성하되, 기설정된 손실함수와 최적화 기법을 이용하여, 출력값을

기반으로, 상기 특징 추출부와 분류 처리부의 레이어에 대한 가중치(weight)를 업데이트 설정하는 것이 바람직하다.

[0023] 더 나아가, 상기 제2 학습 처리부는 다수의 컨볼루션 레이어와 다수의 풀링 레이어로 구성되어, 상기 학습 데이터 셋의 특징을 추출하는 특징 추출부, 적어도 2 개의 FC 레이어로 구성되어, 추출한 특징의 분류를 결정하는 분류 처리부, 기설정된 활성화 함수 레이어로 구성되어, 결정한 분류를 출력값으로 출력하는 출력 함수부 및 적어도 하나의 임베딩 레이어로 구성되어, 상기 학습 데이터 셋을 입력받아 기설정된 차원의 실수 파라미터로 변환하는 임베딩 처리부로 분류 네트워크를 구성하되, 상기 특징 추출부의 레이어는 상기 제1 학습 처리부의 특징 추출부에서 최종 업데이트 설정된 가중치를 적용하고, 기설정된 손실함수와 최적화 기법을 이용하여, 상기 분류 처리부와 임베딩 처리부의 레이어에 대한 가중치를 업데이트 설정하는 것이 바람직하다.

[0024] 더 나아가, 상기 제2 학습 처리부의 분류 처리부는 상기 제1 학습 처리부의 분류 처리부보다 적은 차원의 레이어를 구성하는 것이 바람직하다.

[0025] 더 나아가, 상기 추론 처리부는 클래스를 분류하고자 하는 객체가 인식된 영상 데이터가 입력되는 입력부, 상기 제2 학습 처리부에 의해 최종 학습 처리된 분류 네트워크로, 상기 입력부에 의한 영상 데이터를 입력하여, 예측되는 클래스가 출력되는 출력부, 상기 제2 학습 처리부에 의해 최종 학습 처리된 상기 임베딩 처리부의 가중치 값에 상기 데이터 출력부에 의한 출력값을 맵핑 분석하는 맵핑부 및 상기 맵핑부에 의한 맵핑 분석 결과를 이용하여, 해당하는 객체의 최종 클래스를 분류 결정하여 출력하는 추론부를 포함하는 것이 바람직하다.

[0027] 상기한 바와 같은 문제점을 해결하기 위한 연산 처리 수단에 의해 각 단계가 수행되는 임베딩 기반 객체 클래스 분류 시스템을 이용한 임베딩 기반 객체 클래스 분류 방법으로서, 분류 네트워크에 객체의 클래스 정보가 라벨링된 학습 데이터 셋을 입력받아 학습을 수행하는 제1 학습 단계(S100), 상기 제1 학습 단계(S100)에 의한 학습 결과를 기반으로 분류 네트워크를 구성하고, 상기 학습 데이터 셋을 입력받아 학습을 수행하는 제2 학습 단계(S200) 및 상기 제2 학습 단계(S200)에 의해 최종 학습 처리된 분류 네트워크를 이용하여, 외부로부터 입력되는 클래스를 분류하고자 하는 객체가 인식된 영상 데이터에 포함되어 있는 객체의 클래스 정보를 분류하여 출력하는 추론 처리 단계(S300)를 포함하는 것이 바람직하다.

[0028] 더 나아가, 상기 제2 학습 단계(S200)는 상기 제1 학습 단계(S100)에 의해 최종 학습 처리된 분류 네트워크를 구성하는 다수의 컨볼루션 레이어와 다수의 풀링 레이어의 가중치를 적용하여, 분류 네트워크를 구성하고, 적어도 하나의 임베딩 레이어를 포함하여, 상기 임베딩 레이어에 상기 학습 데이터 셋을 입력하여 기설정된 차원의 실수 파라미터로 변환 출력받는 것이 바람직하다.

[0029] 더 나아가, 상기 제2 학습 단계(S200)의 분류 네트워크를 구성하는 FC(Fully-Connected) 레이어는 상기 제1 학습 단계(S100)의 분류 네트워크를 구성하는 FC 레이어보다 적은 차원의 레이어로 구성하는 것이 바람직하다.

[0030] 더 나아가, 상기 추론 처리 단계(S300)는 상기 제2 학습 단계(S200)에 의해 최종 학습 처리된 분류 네트워크로부터, 상기 영상 데이터에 의한 예측되는 클래스를 출력받아, 최종 학습 처리된 임베딩 레이어의 가중치 값에 출력한 예측 클래스를 맵핑 분석하여, 해당하는 객체의 최종 클래스를 결정하여 출력하는 것이 바람직하다.

## 발명의 효과

[0032] 상기한 바와 같은 본 발명에 의한 임베딩 기반 객체 클래스 분류 시스템 및 그 방법에 의하면, 수많은 클래스를 가지고 있는 객체(일 예를 들자면, 교통 표지판 등)를 분류하는 네트워크의 경우, 메모리 사용량 및 연산량이 제한된 임베디드 환경에서는 구현하기 어렵기 때문에, 임베딩 레이어를 사용하여 출력되는 클래스의 차원을 축소시킴으로써, 제한된 메모리 사용량 및 연산량을 통해서도 구현 가능한 장점이 있다.

[0033] 특히, 자율 주행 레벨을 증가시키기 위한 필수 조건 중 하나인 교통 표지판 분류에 적용함으로써, 클래스 누락 없이 모든 교통 표지판에 적용할 수 있어, GPS 정보가 잘못 되었거나, 예기치 못한 도로 공사 등으로 인하여 지도 정보와 실제 도로 정보가 상이할 경우에도, 교통 표지판을 인식하여 안정적인 주행을 제공할 수 있는 장점이 있다.

[0034] 또한, 교통 표지판에만 제공되는 것이 아니라, MFC(Multi-Function Camera) 등을 통한 다양한 객체 분류 네트워크에 적용하더라도, 자원 최적화가 가능하므로, 복잡한 네트워크를 임베디드 시스템에 용이하게 적용할 수

있어, 인식 성능을 향상시킬 수 있는 장점이 있다.

### 도면의 간단한 설명

- [0036] 도 1은 본 발명의 일 실시예에 따른 임베딩 기반 객체 클래스 분류 시스템을 나타낸 구성 예시도이며,  
 도 2은 본 발명의 일 실시예에 따른 임베딩 기반 객체 클래스 분류 시스템 및 그 방법에 의한 제1 학습 처리 과정을 나타낸 네트워크 예시도이며,  
 도 3은 본 발명의 일 실시예에 따른 임베딩 기반 객체 클래스 분류 시스템 및 그 방법에 의한 제2 학습 처리 과정을 나타낸 네트워크 예시도이며,  
 도 4은 본 발명의 일 실시예에 따른 임베딩 기반 객체 클래스 분류 시스템 및 그 방법에 의한 최종 학습된 네트워크를 이용한 최종 추론 과정을 나타낸 예시도이며,  
 도 5는 본 발명의 일 실시예에 따른 임베딩 기반 객체 클래스 분류 방법을 나타낸 순서 예시도이다.

### 발명을 실시하기 위한 구체적인 내용

- [0037] 이하 첨부된 도면을 참고하여 본 발명에 의한 임베딩 기반 객체 클래스 분류 시스템 및 그 방법의 바람직한 실시예에 관하여 상세히 설명한다.
- [0038] 시스템은 필요한 기능을 수행하기 위하여 조직화되고 규칙적으로 상호 작용하는 장치, 기구 및 수단 등을 포함하는 구성 요소들의 집합을 의미한다.
- [0040] 교통 표지판은 교통에 필요한 주의, 규제 및 지시 등을 표시하는 알림판으로서, 자율 주행 차량이 도로 규칙을 지키기 위해서는 표지판 인식이 필수 조건 중 하나이다.
- [0041] 그렇지만, 교통 표지판이 수백 종류의 다양한 클래스를 가지므로, 현재 차량 내 적용된 제한된 캐시 메모리 용량과 연산량 그리고 실시간 처리를 수행하기 위해서, 특정 클래스(제한 속도 관련) 만을 선택하여 분류하도록 구현되고 있는 실정이다.
- [0042] 자율 주행 레벨 3 이상의 경우, 운전자의 개입이 이루어지지 않게 되기 때문에, 자율 주행 차량이 도로 내 존재하는 모든 종류의 교통 표지판을 인지하지 못할 경우, 유동적인 도로 규칙을 지키면서 안정적으로 주행하는 것은 불가능하다.
- [0043] 이를 일반적인 분류 네트워크로 구성할 경우, one hot encoding 방법을 사용하여, 객체의 클래스 수와 동일하게 출력 수를 구성하기 때문에, 다수의 컨볼루션 레이어와 다수의 풀링 레이어로 구성되어, 입력되는 학습 데이터의 특징을 추출하는 베이스 네트워크 이후에 구성되는, 즉, 네트워크의 종단에 구성되는 FC 레이어에서 요구되는 메모리 사용량 및 연산량이 증가하게 되어, 임베딩 시스템에서의 구현이 현실적으로 불가능하다.
- [0045] 이러한 문제점을 해소하고 효율적으로 교통 표지판을 분류하기 위하여, 본 발명의 일 실시예에 따른 임베딩 기반 객체 클래스 분류 시스템 및 그 방법은, 임베딩 기반 엣지 네트워크를 개시하고 있다.
- [0046] 간단하게는, ResNet, VGG16과 같은 분류 네트워크(Classification Network)를 라벨링된 학습 데이터 셋을 사용하여 학습시킨 후, 베이스 네트워크와 이에 대한 가중치 값을 추출하여 이를 활용하게 된다.
- [0047] 베이스 네트워크는 객체의 특징(feature)을 추출하는 학습이 완료된 상태이므로, 이를 고려하여, 추가 학습을 진행하지 않고 가중치 값을 고정하여 구성하고, 임베딩 레이어와 채널 수가 축소된 FC(Fully-Connected) 레이어에 대해서만 다시 한번 학습을 수행하게 된다.
- [0048] 이러한 임베딩 레이어의 내부 구조는 bias가 없는 FC 레이어와 동일하지만, 그 목적이 one-hot encoding된 라벨 정보를 보다 작은 차원의 실수 파라미터로 변환하는 것이기 때문에, 네트워크를 통한 출력값의 차원이 압축되므로 네트워크 종단의 FC 레이어에서 사용되는 메모리 사용량과 연산량을 줄일 수 있다.



- [0050] 다만, 본 발명의 일 실시예에 따른 임베딩 기반 객체 클래스 분류 시스템 및 그 방법은 용이한 설명을 위해 수많은 클래스를 갖고 있는 객체로 '교통 표지판'을 한정하여 앞서 설명하였고, 후술할 예정이나, 이는 본 발명의 일 실시예에 불과할 뿐, 클래스의 수가 과도하여 기본적으로 필요로 하는 메모리 사용량과 연산량으로 인해 임베딩 시스템에 구현이 어려운 모든 객체 분류에 활용할 수 있다.
- [0052] 도 1은 본 발명의 일 실시예에 따른 임베딩 기반 객체 클래스 분류 시스템의 구성도를 도시한 것이다.
- [0053] 도 1에 도시된 바와 같이, 본 발명의 일 실시예에 따른 임베딩 기반 객체 클래스 분류 시스템은, 제1 학습 처리부(100), 제2 학습 처리부(200) 및 추론 처리부(300)를 포함할 수 있다. 각 구성들은 컴퓨터를 포함하는 연산 처리 수단을 통해서 동작을 수행하는 것이 바람직하며, 상술한 바와 같이, 교통 표지판을 분류하기 위해 임베딩 시스템에 구현될 경우, 차량 내 통신 채널을 통해서 송수신을 수행하는 컴퓨터를 포함하는 ECU와 같은 연산 처리 수단을 통해서 동작을 수행하게 된다.
- [0055] 각 구성에 대해서 자세히 알아보자면,
- [0056] 제1 학습 처리부(100)는 미리 저장된 분류 네트워크(일 예를 들자면, ResNET, VGG16과 같은 분류 네트워크)에 객체의 클래스 정보가 라벨링된 학습 데이터 셋을 입력하여, 학습을 수행하게 된다.
- [0057] 제1 학습 처리부(100)는 도 1에 도시된 바와 같이, 특징 추출부(110), 분류 처리부(120) 및 출력 함수부(130)로 구성되게 된다.
- [0058] 상세하게는, 도 2에 도시된 바와 같이, 다수의 레이어로 구성된 분류 네트워크에서, 데이터베이스에 저장되어 있는 객체(교통 표지판)의 클래스 정보(교통 표지판 종류)가 라벨링된 학습 데이터 셋을 입력받아, 지도 학습을 수행하게 된다.
- [0059] 라벨링된 학습 데이터 셋의 예를 들자면, 300 개의 교통 표지판을 각각 포함하는 이미지 데이터, 각 이미지 데이터가 어떤 의미의 교통 표지를 갖는지를 나타내는 라벨 데이터를 포함하게 된다.
- [0061] 특징 추출부(110)는 'feature extraction'을 위한 구성으로, 다수의 컨볼루션 레이어(convolution layer)와 다수의 풀링 레이어(pooling layer)로 구성되며, 입력되는 학습 데이터 셋의 특징(feature)을 추출하게 된다.
- [0062] 컨볼루션 레이어는 하나 이상의 필터로 구성되며, 필터의 수는 채널의 깊이를 나타내며, 필터가 많을수록 더 많은 이미지 특성을 추출하게 된다. 이러한 필터를 통과한 이미지 픽셀 값은 색상, 선, 형태, 경계 등의 특징이 뚜렷해지며, 필터를 통과한 이미지는 특성값을 가지고 있어, 특성맵(feature map)이라 하며, 이 과정을 컨볼루션 연산이라 한다. 컨볼루션 연산을 진행하면 할수록 이미지 크기는 작아지고, 채널 수는 증가하게 된다.
- [0063] 풀링 레이어는 컨볼루션 레이어 바로 다음에 구성되며, 공간(spatial size)을 축소하는 역할을 수행하게 된다. 이 때, 공간 축소는 width, height의 크기가 축소되고, 채널의 크기는 고정되게 된다. 이를 통해서, 입력 데이터의 크기가 축소되고 학습하지 않기 때문에, 변수의 수가 줄어들어 오버피팅(overfitting) 발생하는 것을 방지하게 된다.
- [0065] 분류 처리부(120)는 'classification'을 위한 구성으로, 네트워크의 종단의 적어도 2개의 FC(Fully-Connected) 레이어로 구성되며, 특징 추출부(110)에서 각각의 학습 데이터마다 추출한 특징에 대한 클래스 분류를 결정하게 된다.
- [0066] 또한, 출력 함수부(130)는 미리 설정된 활성화 함수 레이어를 이용하여, 분류 처리부(120)에서 분류한 클래스 중 확률이 가장 높은 클래스로 최종 네트워크 출력값을 결정하여 출력하게 된다.
- [0067] 이 때, 출력 함수부(130)는 미리 설정된 활성화 함수 레이어로 softmax 함수를 설정하고 있으며, softmax 함수는 마지막 레이어에서 클래스를 분류하기 위해 구성되며, 입력 값을 0 ~ 1 사이의 값으로 정규화하여 합이 1인 확률 분포로 만들어 출력하게 된다.

- [0069] 제1 학습 처리부(100)는 미리 설정된 손실함수와 최적화 기법을 이용하여, 출력 함수부(130)에 의한 출력값을 기반으로, 특징 추출부(110)와 분류 처리부(120)를 구성하고 있는 레이어에 대한 가중치(weight)를 업데이트 설정하게 된다.
- [0070] 즉, 손실함수는 모델의 출력이 얼마나 정답(실제 값)과 가까운지 측정하기 위해 사용되며, 오차가 작을수록 손실함수 값이 작게 된다. 이를 통해서, 손실함수가 작은 방향으로 네트워크 학습을 반복 수행하게 되며, 이 때, 반복 수행에 이용되는 것이 최적화 기법이다. 최적화 기법은 손실함수를 최소화하는 가중치를 찾는 과정으로서, 현재 위치에서 손실함수의 출력값이 감소하는 방향으로 조금씩 가중치를 움직여 나가는 것이다.
- [0071] 이 때, 제1 학습 처리부(100)는 손실함수로 Cross Entropy Loss를 이용하고, 최적화 기법으로 Stochastic Gradient Decent method를 이용하여, 특징 추출부(110)와 분류 처리부(120)를 구성하고 있는 레이어에 대한 가중치의 업데이트를 수행하게 된다. 즉, 제1 학습 처리부(100)는 입력되는 어느 하나의 이미지 데이터로부터 추출한 교통 표지판 영역(후보 영역)이 학습 데이터 셋을 통해서 입력받은 300 개의 이미지 데이터 중 어떤 라벨의 이미지 데이터인지 분류하고, 분류한 라벨 결과와 실제 라벨 결과(정답 데이터) 간의 손실함수를 구하되, 손실함수가 최소화되도록 최적화 기법을 사용해서 네트워크를 구성하고 있는 레이어의 가중치 값을 업데이트하게 된다.
- [0072] 이러한 제1 학습 처리부(100)에 의한 특징 추출부(110), 분류 처리부(120) 및 출력 함수부(130)에 의한 동작은 종래의 분류 네트워크의 지도 학습과 유사하게 동작하게 된다.
- [0074] 그렇지만, 제2 학습 처리부(200)는 종래의 분류 네트워크의 지도 학습과 유사하지만, 학습 과정이 상이하다.
- [0075] 상세하게는, 제2 학습 처리부(200)는 제1 학습 처리부(100)의 학습 결과를 기반으로 분류 네트워크를 구성하고, 객체의 클래스 정보가 라벨링된 학습 데이터 셋을 입력하여, 학습을 수행하게 된다. 여기서, 학습 데이터 셋은 제1 학습 처리부(100)로 입력된 학습 데이터 셋과 동일하게 된다.
- [0076] 제2 학습 처리부(200)는 임베딩 기반의 제한된 메모리 사용량과 연산량에도 구현될 수 있도록, 제1 학습 처리부(100)에 의해 학습이 완료된 베이스 네트워크를 사용하는 것이 바람직하다.
- [0077] 이를 위해, 제2 학습 처리부(200)는 도 1에 도시된 바와 같이, 특징 추출부(210), 분류 처리부(220), 출력 함수부(230) 및 임베딩 처리부(240)를 포함하게 된다.
- [0078] 도 3에 도시된 바와 같이, 특징 추출부(210)는 'feature extraction'을 위한 구성으로, 다수의 컨볼루션 레이어와 다수의 풀링 레이어로 구성되어, 입력되는 학습 데이터 셋의 특징을 추출하게 된다.
- [0079] 컨볼루션 레이어는 하나 이상의 필터로 구성되며, 필터의 수는 채널의 깊이를 나타내며, 필터가 많을수록 더 많은 이미지 특성을 추출하게 된다. 이러한 필터를 통과한 이미지 픽셀 값은 색상, 선, 형태, 경계 등의 특징이 뚜렷해지며, 필터를 통과한 이미지는 특성값을 가지고 있어, 특성맵이라 하며, 이 과정을 컨볼루션 연산이라 한다. 컨볼루션 연산을 진행하면 할수록 이미지 크기는 작아지고, 채널 수는 증가하게 된다.
- [0080] 풀링 레이어는 컨볼루션 레이어 바로 다음에 구성되어, 공간(spatial size)을 축소하는 역할을 수행하게 된다. 이 때, 공간 축소는 width, height의 크기가 축소되고, 채널의 크기는 고정되게 된다. 이를 통해서, 입력 데이터의 크기가 축소되고 학습하지 않기 때문에, 변수의 수가 줄어들어 오버피팅 발생하는 것을 방지하게 된다.
- [0081] 단, 제2 학습 처리부(200)의 특징 추출부(210)는 제1 학습 처리부(100)의 특징 추출부(110)에서 최종 업데이트 설정된 가중치를 이용하여, 구성되는 다수의 컨볼루션 레이어와 다수의 풀링 레이어의 가중치를 설정하게 된다.
- [0082] 다시 말하자면, 제1 학습 처리부(100)의 베이스 네트워크는 교통 표지판의 특징 추출 학습이 완료된 상태이므로, 제2 학습 처리부(200)의 베이스 네트워크는 이에 대한 반복 학습 처리 없이, 구성되는 레이어들의 가중치를 제1 학습 처리부(100)에 의한 최종 업데이트 결과로 고정(fix)하여 구성하게 된다.
- [0083] 이에 따라, 제2 학습 처리부(200)는 학습되는 영역이 분류 처리부(220)와 임베딩 처리부(240)로 한정되게 된다.
- [0085] 분류 처리부(220)는 'classification'을 위한 구성으로, 네트워크의 종단의 적어도 2개의 FC 레이어로 구성되어, 특징 추출부(210)에서 각각의 학습 데이터마다 추출한 특징에 대한 클래스 분류를 결정하게 된다.

- [0086] 또다른 학습 영역인 임베딩 처리부(240)는 적어도 하나의 임베딩 레이어(embedding layer)로 구성되어, 특징 추출부(210)로 입력되는 학습 데이터 셋을 같이 입력받게 된다.
- [0087] 임베딩 처리부(240)에 의한 임베딩 레이어의 내부 구조는 bias가 없는 FC 레이어와 동일하지만, 그 목적에 있어서, one hot encoding된 학습 데이터 셋을 미리 설정된 N차원(여기서 N은 1 이상의 실수임.) 실수로 변환하게 된다.
- [0088] 라벨링된 학습 데이터 셋의 예를 300 개의 교통 표지판 관련 라벨 데이터가 one hot encoding된 데이터로 가정하고 있기 때문에, 여기에 이어서, 임베딩 처리부(240)의 임베딩 레이어는 이를 미리 설정된 차원인 3차원의 실수 파라미터로 변환하게 된다.
- [0089] 다시 말하자면, 라벨링된 학습 데이터 셋이란 300개의 라벨 데이터가 구성되되, 각 라벨 데이터는 0 또는 1의 값을 갖기 때문에, 300 차원의 데이터라고 볼 수 있다. 임베딩 레이어는 이러한 300차원의 데이터를 입력받아 이를 3차원으로 변환하여 출력하게 된다.
- [0090] 즉, 300차원의 데이터가 임베딩 레이어로 입력되면, 3차원으로 변환되어 출력되고, 이 때, 3차원으로 변환 출력이란, 3 개의 실수 파라미터를 의미한다.
- [0092] 이에 따라, 제2 학습 처리부(200)는 구성되는 분류 네트워크의 출력값이 임베딩 레이어를 통해서 출력되는 3개의 실수 파라미터와 동일하게 되도록 손실함수를 구하고, 손실함수가 최소화되도록 최적화 기법을 사용해서 네트워크를 구성하고 있는 FC 레이어와 임베딩 레이어의 가중치 값을 업데이트하게 된다.
- [0093] 이 때, 제2 학습 처리부(200)는 손실함수로 L1 Loss를 이용하고, 최적화 기법으로 Stochastic Gradient Decent method를 이용하여, 분류 처리부(220)와 임베딩 처리부(240)를 구성하고 있는 레이어에 대한 가중치의 업데이트를 수행하게 된다.
- [0095] 이와 같이, 라벨 데이터의 크기가 축소되었기 때문에, 제2 학습 처리부(200)의 분류 처리부(220)에 구성된 FC 레이어는 제1 학습 처리부(100)의 분류 처리부(120)에 구성된 FC 레이어보다 채널 수가 축소된, 다시 말하자면, 적은 차원의 레이어로 구성되게 된다.
- [0096] 이를 통해서, 임베딩 레이어를 통해서, 300차원이었던 학습 데이터 셋의 클래스를 3차원으로 압축하므로, FC 레이어에 사용되는 메모리 사용량 및 연산량을 감소시킬 수 있다.
- [0098] 출력 함수부(230)는 미리 설정된 활성화 함수 레이어를 이용하여, 분류 처리부(220)에 의해서 결정한 분류를 출력값으로 출력하게 된다.
- [0099] 상세하게는, 네트워크의 최종 출력을 미리 설정된 활성화 함수 레이어인 hyperbolic tangent 함수를 사용하여, 3차원의 실수 값을 출력하게 된다.
- [0101] 추론 처리부(300)는 제2 학습 처리부(200)에 의해 최종 학습 처리된 분류 네트워크를 이용하여, 입력되는 영상 데이터, 즉, 학습이 완료된 후 새롭게 입력되는 영상 데이터에 포함되어 있는 추출 객체의 클래스 정보를 분류하여 출력하게 된다.
- [0102] 추론 처리부(300)는 도 1에 도시된 바와 같이, 입력부(310), 출력부(320), 맵핑부(330) 및 추론부(340)를 포함하게 된다.
- [0103] 입력부(310)는 클래스를 분류하고자 하는 객체가 인식된 영상 데이터가 입력되게 된다.
- [0104] 출력부(320)는 제2 학습 처리부(200)에 의해 최종 학습 처리된 분류 네트워크로 입력부(310)에 의한 영상 데이터를 입력하여, 예측되는 클래스가 출력되게 된다.
- [0105] 맵핑부(330)는 제2 학습 처리부(200)에 의해 최종 학습 처리된 임베딩 처리부(240)의 가중치 값에 데이터 출력부에 의한 출력값을 맵핑 분석하게 된다.

- [0106] 추론부(340)는 맵핑부(230)에 의한 맵핑 분석 결과를 이용하여, 해당하는 객체의 최종 클래스를 결정하여 출력하게 된다. 이 때, 추론부(340)에 의한 출력값이 해당하는 객체의 최종 클래스 분류값에 해당한다.
- [0108] 추론 처리부(300)는 도 4에 도시된 바와 같이, 제2 학습 처리부(200)에 의해 최종 학습 처리된 분류 네트워크를 이용하되, 출력 클래스의 공간을 매우 큰 차원(일 예를 들자면, 300차원)에서 미리 설정된 적은 차원(일 예를 들자면, 3차원)으로 축소시킴으로써, 딥러닝 분류 네트워크의 메모리 사용량과 연산량을 감소시킬 수 있어, 임베디드 시스템에서의 구현이 가능하도록 하였다.
- [0109] 상세하게는, 제2 학습 처리부(200)에 의해 학습이 완료된 분류 네트워크의 출력은 3개의 실수 파라미터이므로, 임베딩 레이어의 가중치 값과 비교하여, 거리(L2)가 최소가 되는 인덱스 값을 클래스 값으로 맵핑하게 된다. 이 때, 임베딩 레이어의 가중치 값은 도 4에 도시된 바와 같이, 룩업 테이블 형태로 나타낼 수 있으며, 출력값을 기준으로 주변 인덱스 값(가중치 값) 간의 거리(L2)가 가장 최소가 되는 항목으로 클래스 분류를 수행하게 된다.
- [0110] 이 때, 제2 학습 처리부(200)에 의해 학습이 완료된 분류 네트워크의 출력은 사용된 활성화 함수 레이어로 인해, 도 4에 도시된 바와 같이, 3개의 실수 파라미터( $c_0$ ,  $c_1$ ,  $c_2$ )로 출력되게 된다.
- [0112] 이러한 본 발명의 일 실시예에 따른 임베딩 기반 객체 클래스 분류 시스템 및 그 방법의 효과를 검증하기 위하여, 300 개의 교통 표지판일 때, 즉, 클래스 수가 300개이고, 2개의 FC 레이어를 구성하였을 때의 메모리 사용량과 연산량을 비교하였으며, 그 결과는 하기의 표 1과 같다.

표 1

Item	종래의 분류 네트워크	제2 학습 처리부(200)에 의해 학습이 완료된 분류 네트워크
메모리 사용량(MB)	720,000	10,600
연산량(Flops)	180,000	2,650

- [0114]
- [0116] 표 1와 같이, 종래의 분류 네트워크는 2개의 FC 레이어 모두에서 300개의 입/출력을 사용하였으나, 본 발명의 분류 네트워크는 출력 차원을 3으로 축소하고, FC 레이어의 입출력도 50으로 사용하였다. 이에 따라, 본 발명의 일 실시예에 따른 임베딩 기반 객체 클래스 분류 시스템 및 그 방법은 출력값의 차원 자체를 1/100으로 감소시킬 수 있어, 종래 방법 대비 1.5%의 메모리 사용량 및 연산량으로도 네트워크 구현이 가능하다.
- [0117] 이를 통해서, 수많은 클래스를 갖는 분류 네트워크도 임베디드 시스템에 구현할 수 있어, 다양한 분야에서 효율적으로 활용할 수 있는 장점이 있다.
- [0119] 도 5는 본 발명의 일 실시예에 따른 임베딩 기반 객체 클래스 분류 방법의 순서도를 도시한 것이다.
- [0120] 도 5에 도시된 바와 같이, 본 발명의 일 실시예에 따른 임베딩 기반 객체 클래스 분류 방법은, 제1 학습 단계(S100), 제2 학습 단계(S200) 및 추론 처리 단계(S300)를 포함할 수 있다. 각 단계는 연산 처리 수단에 의해 동작 수행되는 임베딩 기반 객체 클래스 분류 시스템을 이용하는 것이 바람직하다.
- [0122] 각 단계에 대해서 자세히 알아보자면,
- [0123] 제1 학습 단계(S100)는 미리 저장된 분류 네트워크에 객체의 클래스 정보가 라벨링된 학습 데이터 셋을 입력하여, 학습을 수행하게 된다.
- [0124] 상세하게는, 제1 학습 단계(S100)는 다수의 레이어로 구성된 분류 네트워크에서, 데이터베이스에 저장되어 있는 객체(교통 표지판)의 클래스 정보(교통 표지판 종류)가 라벨링된 학습 데이터 셋을 입력받아, 지도 학습을 수행

하게 된다.

- [0125] 이 때, 제1 학습 단계(S100)의 분류 네트워크는 다수의 컨볼루션 레이어와 다수의 풀링 레이어를 포함하여 입력되는 학습 데이터 셋의 특징을 추출하는 구성과, 적어도 2개의 FC 레이어를 포함하여 추출한 특징에 대한 클래스 분류를 결정하는 구성과, 활성화 함수 레이어를 포함하여 분류한 클래스 중 확률이 가장 높은 클래스로 최종 네트워크 출력값을 결정하는 구성을 포함하게 된다.
- [0126] 또한, 제1 학습 단계(S100)의 분류 네트워크는 미리 설정된 손실함수와 최적화 기법을 이용하여, 출력값을 기반으로, 다수의 컨볼루션 레이어, 다수의 풀링 레이어 및 적어도 2개의 FC 레이어에 대한 가중치를 업데이트 설정하게 된다.
- [0127] 즉, 출력값(분류한 라벨 결과값)과 실제 라벨 결과(정답 데이터) 간의 손실함수를 구하되, 손실함수가 최소화되도록 최적화 기법을 사용해서 네트워크를 구성하고 있는 레이어의 가중치 값을 업데이트하게 된다.
- [0129] 제2 학습 단계(S200)는 제1 학습 단계(S100)에 의한 학습 결과를 기반으로 분류 네트워크를 구성하고, 라벨링된 학습 데이터 셋을 입력하여, 학습을 수행하게 된다.
- [0130] 상세하게는, 제2 학습 단계(S200) 역시도 다수의 레이어로 구성된 분류 네트워크에서, 데이터베이스에 저장되어 있는 객체(교통 표지판)의 클래스 정보(교통 표지판 종류)가 라벨링된 학습 데이터 셋을 입력받아, 지도 학습을 수행하게 되되, 임베딩 기반의 제한된 메모리 사용량과 연산량에도 구현될 수 있도록, 제1 학습 단계(S100)에 의해 학습이 완료된 베이스 네트워크를 그대로 사용하게 된다.
- [0131] 즉, 제2 학습 단계(S200)의 분류 네트워크는 다수의 컨볼루션 레이어와 다수의 풀링 레이어를 포함하여 입력되는 학습 데이터 셋의 특징을 추출하는 구성과, 적어도 2개의 FC 레이어를 포함하여 추출한 특징에 대한 클래스 분류를 결정하는 구성과, 활성화 함수 레이어를 포함하여 분류한 클래스 중 확률이 가장 높은 클래스로 최종 네트워크 출력값을 결정하는 구성과, 임베딩 레이어를 포함하여 학습 데이터 셋의 차원을 변환하는 구성을 포함하게 된다.
- [0132] 이 때, 제2 학습 단계(S200)의 분류 네트워크는 다수의 컨볼루션 레이어와 다수의 풀링 레이어를 포함하여 입력되는 학습 데이터 셋의 특징을 추출하는 구성으로, 제1 학습 단계(S100)에 의해 최종 업데이트 설정된 가중치를 이용하여, 구성되는 다수의 컨볼루션 레이어와 다수의 풀링 레이어의 가중치를 설정하게 된다.
- [0134] 다시 말하자면, 제1 학습 단계(S100)를 통해서, 제1 학습 단계(S100)의 분류 네트워크는 교통 표지판의 특징 추출 학습이 완료된 상태이므로, 이에 대한 베이스 네트워크 영역에 대해서는 제2 학습 단계(S200)에서의 반복 학습 처리 없이 구성되는 레이어들의 가중치를 제1 학습 단계(S100)에 의한 최종 업데이트 결과로 고정하여 구성하게 된다.
- [0135] 이를 통해서, 제2 학습 단계(S200)에서의 학습 영역은 적어도 2개의 FC 레이어를 포함하여 추출한 특징에 대한 클래스 분류를 결정하는 구성과 임베딩 레이어를 포함하여 학습 데이터 셋의 차원을 변환하는 구성으로 한정되게 된다.
- [0136] 이 때, 임베딩 레이어의 내부 구조는 bias가 없는 FC 레이어와 동일하지만, 그 목적에 있어서, one hot encoding된 학습 데이터 셋을 미리 설정된 N차원(여기서 N은 1 이상의 실수임.) 실수로 변환하게 된다.
- [0137] 라벨링된 학습 데이터 셋의 예를 300 개의 교통 표지판 관련 라벨 데이터가 one hot encoding된 데이터로 가정하고 있기 때문에, 여기에 이거서, 임베딩 처리부(240)의 임베딩 레이어는 이를 미리 설정된 차원인 3차원의 실수 파라미터로 변환하게 된다.
- [0138] 다시 말하자면, 라벨링된 학습 데이터 셋이란 300개의 라벨 데이터가 구성되되, 각 라벨 데이터는 0 또는 1의 값을 갖기 때문에, 300 차원의 데이터라고 볼 수 있다. 임베딩 레이어는 이러한 300차원의 데이터를 입력받아 이를 3차원으로 변환하여 출력하게 된다.
- [0139] 즉, 300차원의 데이터가 임베딩 레이어로 입력되면, 3차원으로 변환되어 출력되고, 이 때, 3차원으로 변환 출력이란, 3 개의 실수 파라미터를 의미한다.
- [0140] 이에 따라, 제2 학습 단계(S200)의 분류 네트워크는 네트워크의 출력값이 임베딩 레이어를 통해서 출력되는 3개



의 실수 파라미터와 동일하게 되도록 손실함수를 구하고, 손실함수가 최소화되도록 최적화 기법을 사용해서 네트워크를 구성하고 있는 FC 레이어와 임베딩 레이어의 가중치 값을 업데이트하게 된다.

[0141] 이와 같이, 라벨 데이터의 크기가 축소되었기 때문에, 제2 학습 단계(S200)의 분류 네트워크에 포함되는 FC 레이어는 제1 학습 단계(S100)의 분류 네트워크에 포함되는 FC 레이어보다 채널 수가 축소된, 다시 말하자면, 적은 차원의 레이어로 구성되게 된다.

[0142] 이를 통해서, 임베딩 레이어를 통해서, 300차원이었던 학습 데이터 셋의 클래스를 3차원으로 압축하므로, FC 레이어에 사용되는 메모리 사용량 및 연산량을 감소시킬 수 있다.

[0144] 추론 처리 단계(S300)는 제2 학습 단계(S200)에 의해 최종 학습 처리된 분류 네트워크를 이용하여, 외부로부터 입력되는 클래스를 분류하고자 하는 객체가 인식된 영상 데이터에 포함되어 있는 객체의 클래스 정보를 분류하여 출력하게 된다.

[0145] 상세하게는, 추론 처리 단계(S300)는 제2 학습 단계(S200)에 의해 최종 학습 처리된 분류 네트워크로부터, 상기 영상 데이터에 의한 예측되는 클래스를 출력받아, 최종 학습 처리된 임베딩 레이어의 가중치 값에 출력한 예측 클래스를 맵핑 분석하여, 해당하는 객체의 최종 클래스를 결정하여 출력하게 된다.

[0146] 즉, 학습이 완료된 후 새롭게 입력되는 영상 데이터에 포함되어 있는 추출 객체의 클래스 정보를 분류하여 출력하는 것으로, 제2 학습 단계(S200)에 의해 최종 학습 처리된 분류 네트워크를 이용하되, 출력 클래스의 공간을 매우 큰 차원(일 예를 들자면, 300차원)에서 미리 설정된 적은 차원(일 예를 들자면, 3차원)으로 축소시킴으로써, 딥러닝 분류 네트워크의 메모리 사용량과 연산량을 감소시킬 수 있어, 임베디드 시스템에서의 구현이 가능하도록 하였다.

[0147] 추론 처리 단계(S300)는, 제2 학습 단계(S200)에 의해 최종 학습 처리된 분류 네트워크의 출력은 3개의 실수 파라미터이므로, 임베딩 레이어의 가중치 값과 비교하여, 거리(L2)가 최소가 되는 인덱스 값을 클래스 값으로 맵핑하게 된다. 이 때, 임베딩 레이어의 가중치 값은 도 4에 도시된 바와 같이, 룩업 테이블 형태로 나타낼 수 있으며, 출력값을 기준으로 주변 인덱스 값(가중치 값) 간의 거리(L2)가 가장 최소가 되는 항목으로 클래스 분류를 수행하게 된다.

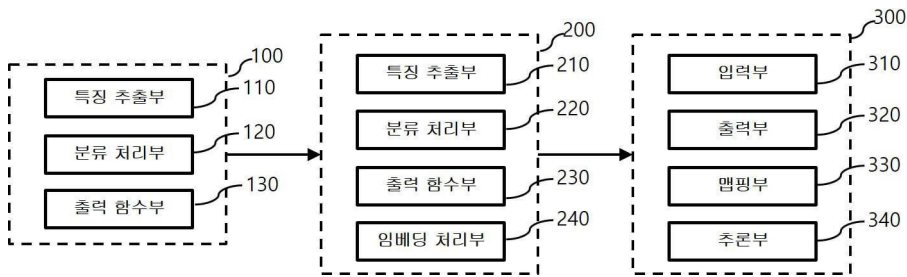
[0149] 본 발명은 상기한 실시예에 한정되지 아니하며, 적용범위가 다양함은 물론이고, 청구범위에서 청구하는 본 발명의 요지를 벗어남이 없이 다양한 변형 실시가 가능한 것은 물론이다.

## 부호의 설명

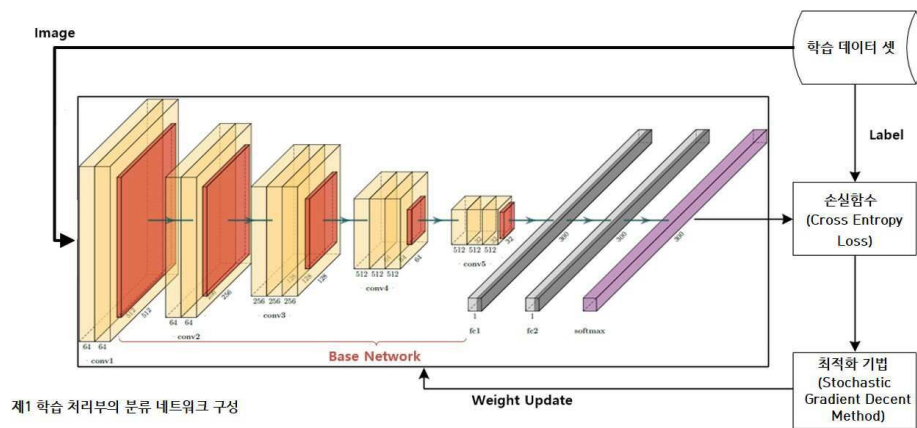
[0151] 100 : 제1 학습 처리부  
110 : 특징 추출부      120 : 분류 처리부  
130 : 출력 함수부  
200 : 제2 학습 처리부  
210 : 특징 추출부      220 : 분류 처리부  
230 : 출력 함수부      240 : 임베딩 처리부  
300 : 추론 처리부  
310 : 입력부      320 : 출력부  
330 : 맵핑부      340 : 추론부

## 도면

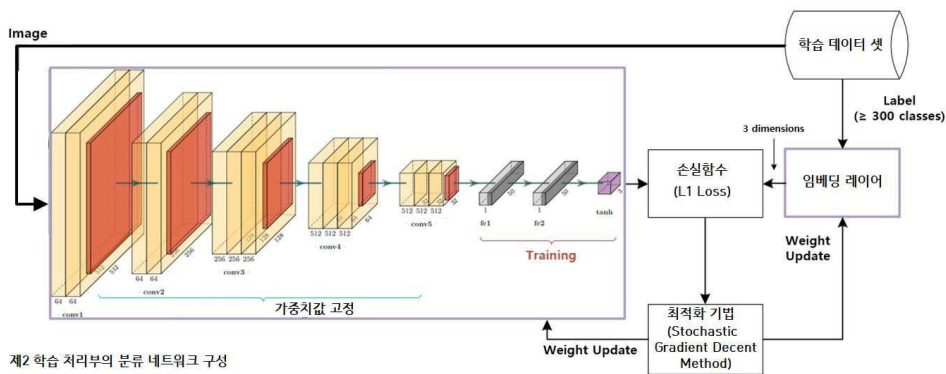
### 도면1



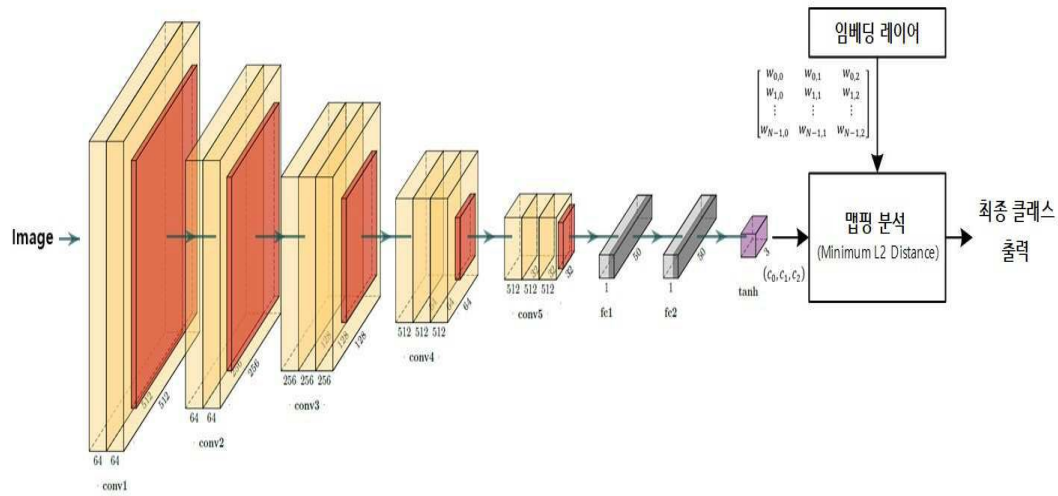
### 도면2



### 도면3



도면4



도면5

