(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2018-0092455
(43) 공개일자 2018년08월20일

(51) 국제특허분류(Int. Cl.)
G06K 9/32 (2006.01) *G06K 17/00* (2006.01)
G06K 9/34 (2006.01) *G06K 9/38* (2006.01)
(52) CPC특허분류
G06K 9/3258 (2013.01)
G06K 9/344 (2013.01)
(21) 출원번호 10-2017-0018148
(22) 출원일자 2017년02월09일
심사청구일자 2017년02월09일

(71) 출원인
한국기술교육대학교 산학협력단
충남 천안시 동남구 병천면 충절로 1600, 내 (한국기술교육대학교)
(72) 발명자
조태훈
경기도 성남시 분당구 미금로 215 청솔마을대원아파트, 806-1202
서성민
대전광역시 서구 청사로 65, 102-605
(뒷면에 계속)
(74) 대리인
정희환

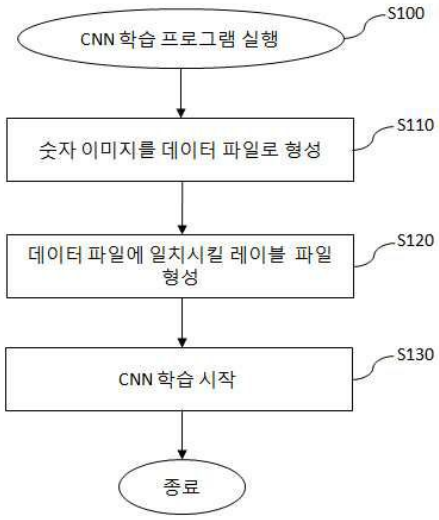
전체 청구항 수 : 총 2 항

(54) 발명의 명칭 **딥 러닝을 이용한 카드번호 인식방법**

(57) 요 약

본 발명은 딥 러닝을 이용한 카드번호 인식방법에 관한 것으로, 여러 각도에서 카드를 촬영하여 수천개 이상의 숫자 이미지를 데이터 형태로 변환하고 해당 이미지들이 어떤 숫자를 나타내는지 레이블링(labeling)을 진행하는 단계; 상기 숫자 이미지 데이터와 레이블링(labeling) 데이터를 사용하여 해당 이미지가 나타내는 숫자에 대한 CNN(Convolution Neural Network)을 이용하여 학습을 진행하는 단계; 상기 CNN 학습을 마치면 weight 값과 뉴럴 네트워크(neural network)의 구조를 포함하고 있는 파일로 생성하는 단계; 상기 CNN 학습 및 판별에 필요한 숫자를 스마트폰으로 카드를 촬영하여 서버 PC에서 최적의 이미지로 추출하여 데이터 파일로 생성하는 단계; 및 상기 PC에서 생성한 데이터 파일과 상기 CNN을 이용하여 학습된 파일을 불러 숫자를 판별하여 스마트폰의 화면에 숫자를 출력하는 단계;를 포함하는 것을 특징으로 한다.

대 표 도 - 도1



(52) CPC특허분류
G06K 9/38 (2013.01)
G06N 3/084 (2013.01)
G06K 2017/0067 (2013.01)
(72) 발명자
정대헌
대전광역시 중구 계백로1603번길 28-19
윤영선
경기도 수원시 영통구 매영로 366, 723-403

명 세 서

청구범위

청구항 1

여러 각도에서 카드를 촬영하여 수천개 이상의 숫자 이미지를 데이터 형태로 변환하고 해당 이미지들이 어떤 숫자를 나타내는지 레이블링(labeling)을 진행하는 단계;

상기 숫자 이미지 데이터와 레이블링(labeling) 데이터를 사용하여 해당 이미지가 나타내는 숫자에 대한 CNN(Convolution Neural Network)을 이용하여 학습을 진행하는 단계;

상기 CNN 학습을 마치면 weight 값과 뉴럴 네트워크(neural network)의 구조를 포함하고 있는 파일로 생성하는 단계;

상기 CNN 학습 및 판별에 필요한 숫자를 스마트폰으로 카드를 촬영하여 서버 PC에서 최적의 이미지로 추출하여 데이터 파일로 생성하는 단계; 및

상기 PC에서 생성한 데이터 파일과 상기 CNN을 이용하여 학습된 파일을 불러 숫자를 판별하여 스마트폰의 화면에 숫자를 출력하는 단계;를 포함하는 것을 특징으로 하는 딥 러닝을 이용한 카드번호 인식방법.

청구항 2

제1항에 있어서,

상기 스마트폰으로 카드를 촬영하여 서버 PC에서 최적의 이미지로 추출하는 단계는,

스마트폰에서 사진을 전송받은 카드 이미지를 PC에서 그레이(Gray) 변환 및 평활화를 거쳐 전처리 과정을 진행하는 단계;

상기 전처리 과정 이후에 카드에서 16자리의 숫자 부분의 위치만을 파악하기 위해 ROI을 지정하는 단계;

상기 ROI로 지정된 이미지 부분에서 엣지(Edge)를 추출하여 숫자들의 윤곽선을 추출하는 단계;

상기 추출된 엣지를 토대로 카드 사이에 빈공간의 위치를 찾기 위해 수직 투영을 진행하는 단계;

상기 수직 투영 이후에 16자리 숫자를 4등분하는 단계; 및

상기 4등분한 숫자를 16개로 분할하여 이미지를 28x28 크기로 변환하는 단계;를 포함하는 것을 특징으로 하는 딥 러닝을 이용한 카드번호 인식방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 카드번호 인식방법에 관한 것으로, 더욱 상세하게는 딥 러닝을 이용한 카드번호 인식방법에 관한 것이다.

배 경 기 술

[0002] 현재까지 모바일 및 웹 상에서 거래 시 신용카드를 이용해 결제를 하기 위해선 카드 번호를 사용자가 직접 입력해야 하는 상황이다.

[0003] 이러한 불편함을 해결하기 위해서 신용 카드 번호와 유효기간을 핸드폰 카메라를 통해 입력란에 자동으로 입력해 주는 기술들이 나왔다. 그에 대표적인 것은 오픈 소스 형태의 card.io가 있다. 이 어플리케이션은 스마트폰 카메라를 이용해 화면상 표시되는 특정 위치에 카드를 위치시키면 16자리의 카드 숫자를 인식하는 형태이다.

[0004] 그러나 이 기술의 문제점은 카드를 촬영하는 환경에 따라 숫자를 올바르게 인식할 수 없는 상황이 있다는 것이다. 예를 들어 조명이 밝은 곳에서는 카드 숫자 부분에 생기는 명암과 숫자에서 조명으로 인해 반사되는 현상 즉, 하이라이트 현상이 발생하게 될 경우 숫자를 올바르게 인식하지 못하는 경우가 있다.

[0005] 다른 기술들로는 삼성 페이, 안드로이드 페이와 같은 기술에서 결제 시 사용할 카드를 등록하는 과정에서 스마트폰 카메라를 이용하여 카드 숫자를 자동 추출하여 입력하는 기술을 사용하고 있다. 이러한 기술들 역시 촬영하는 환경에 따라 숫자를 올바르게 인식할 수 없는 경우가 있다.

발명의 내용

해결하려는 과제

[0006] 상기와 같은 문제점을 해결하기 위한 본 발명의 목적은 모바일 상에서 카드 결제 시 카메라로 촬영하여 정확하게 숫자를 인식할 수 있는 딥 러닝을 이용한 카드번호 인식방법을 제공하는 데 있다.

과제의 해결 수단

[0007] 상기와 같은 목적을 달성하기 위한 본 발명은 여러 각도에서 카드를 촬영하여 수천개 이상의 숫자 이미지를 데이터 형태로 변환하고 해당 이미지들이 어떤 숫자를 나타내는지 레이블링(labeling)을 진행하는 단계; 상기 숫자 이미지 데이터와 레이블링(labeling) 데이터를 사용하여 해당 이미지가 나타내는 숫자에 대한 CNN(Convolution Neural Network)을 이용하여 학습을 진행하는 단계; 상기 CNN 학습을 마치면 weight 값과 뉴럴 네트워크(neural network)의 구조를 포함하고 있는 파일로 생성하는 단계; 상기 CNN 학습 및 판별에 필요한 숫자를 스마트폰으로 카드를 촬영하여 서버 PC에서 최적의 이미지로 추출하여 데이터 파일로 생성하는 단계; 및 상기 PC에서 생성한 데이터 파일과 상기 CNN을 이용하여 학습된 파일을 불러 숫자를 판별하여 스마트폰의 화면에 숫자를 출력하는 단계;를 포함하는 딥 러닝을 이용한 카드번호 인식방법을 제공한다.

[0008] 상기 스마트폰으로 카드를 촬영하여 서버 PC에서 최적의 이미지로 추출하는 단계는, 스마트폰에서 사진을 전송받은 카드 이미지를 PC에서 그레이(Gray) 변환 및 평활화를 거쳐 전처리 과정을 진행하는 단계; 상기 전처리 과정 이후에 카드에서 16자리의 숫자 부분의 위치만을 파악하기 위해 ROI을 지정하는 단계; 상기 ROI로 지정된 이미지 부분에서 엣지(Edge)를 추출하여 숫자들의 윤곽선을 추출하는 단계; 상기 추출된 엣지를 토대로 카드 사이에 빈공간의 위치를 찾기 위해 수직 투영을 진행하는 단계; 상기 수직 투영 이후에 16자리 숫자를 4등분하는 단계; 및 상기 4등분한 숫자를 16개로 분할하여 이미지를 28x28 크기로 변환하는 단계;를 포함하는 것을 특징으로 한다.

발명의 효과

[0009] 이상과 같이 본 발명에 따르면, 모바일 상에서 결제 시 신용카드를 카메라로 촬영하여 숫자를 딥 러닝을 이용하여 정확하게 인식함으로써 전자거래를 함에 있어 간편한 결제 보조 수단으로 이용될 수 있다.

도면의 간단한 설명

[0010] 도 1은 본 발명의 일실시예에 따른 딥 러닝을 이용한 카드번호 인식방법을 나타낸 흐름도이다.

도 2는 본 발명의 일실시예에 따른 모바일 폰으로 촬영한 신용카드의 숫자 인식 방법을 흐름도이다.

도 3은 본 발명의 일실시예에 따른 모바일 폰으로 촬영한 신용카드의 숫자 인식 과정의 출력을 보여주는 화면이다.

발명을 실시하기 위한 구체적인 내용

[0011] 아래에서는 첨부한 도면을 참고로 하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다.

[0012] 그러면, 본 발명의 일실시예에 따른 딥 러닝을 이용한 카드번호 인식방법에 대하여 설명한다.

[0013] 도 1은 본 발명의 일실시예에 따른 딥 러닝을 이용한 카드번호 인식방법을 나타낸 흐름도이다.

[0014] 도 1을 참조하면, 본 발명의 딥 러닝을 이용한 카드번호 인식방법은, 먼저 주체적인 실행단계 전 CNN(Convolution Neural Network)을 이용한 신용카드의 숫자 이미지 학습 단계를 실행한다(S100). 여기서, CNN이란 딥 러닝 알고리즘의 하나로써 이미지를 입력 시 그 이미지가 어떤 이미지인지 판별하는 Classification 모델이라고 할 수 있다.

[0015] CNN은 일반적으로 3계층을 가지는데 3계층은 아래와 같이 설명할 수 있다.

[0016] 1) Convolution Layer : convolution을 이용해 나타낸 특징을 추출하는 layer로 입력된 이미지에서 의미 있는 특징을 추출한다.

[0017] 2) Pooling Layer : 이미지의 특성상 많은 픽셀이 존재하는데 이에 따라 상당한 양의 이미지 특징들이 존재하게 된다. 이러한 특징들을 줄이기 위해 subsampling한다.

[0018] 3) Feedforward Layer : convolution layer와 pooling layer에서 나온 특징들을 이용해 분류를 위해 사용된다.

[0019] 일반적인 CNN의 구조는 Convolution layer->Pooling layer->Convolution layer->Pooling layer-> ... -> Feedforward layer로 이루어져 있다. Pooling layer를 이용해 이미지를 줄이고 Convolution layer을 이용해 자질을 추출하는 것을 반복한 후 Feedforward layer를 통해 분류가 이루어진다고 할 수 있다. 층이 많을수록 분류 성능은 좋아진다.

[0020] 실시예로써, 약 100여장의 카드를 여러 각도 여러 환경에서 촬영하여 약 2600여개의 숫자를 획득하였고, 이에 한정하지 않는다. 획득한 숫자 중 0부터 9까지의 분포는 아래 표와 같다.

표 1

[0021]

0	1	2	3	4	5	6	7	8	9
249	262	243	280	232	255	274	221	243	241

[0022] 이때, 약 2600여개의 숫자 이미지를 데이터 형태로 변환하여 데이터 파일로 형성하고(S110), 해당 이미지들이 어떤 숫자를 나타내는지 labeling을 진행하여 데이터 파일에 일치시킬 레이블 파일을 형성한다(S120). 다시 말해 숫자 1에 해당하는 이미지는 1이라고 표시하여 데이터를 생성한다.

[0023] 이후 실제 이미지 데이터와 labeling 데이터를 사용해 해당 이미지가 나타내는 숫자에 대한 CNN 학습을 진행하고(S130), CNN 학습을 마치면 weight 값과 neural network의 구조를 포함하고 있는 파일을 생성한다. 이 파일의 확장자를 nnt 파일이라고 하면 추후 인식단계에서 이 파일을 이용해 카드 숫자를 분류하는데 사용한다.

[0024] 이하에서는 카드 이미지에서 CNN 학습 및 판별에 필요한 숫자를 최적의 이미지로 추출하기 위한 숫자 검출 알고리즘에 대하여 설명한다.

[0025] 도 2는 본 발명의 일실시예에 따른 스마트폰으로 촬영한 신용카드의 숫자 인식 방법을 흐름도이고, 도 3은 본 발명의 일실시예에 따른 스마트폰으로 촬영한 신용카드의 숫자 인식 과정의 출력을 보여주는 화면이다.

[0026] 도 2 및 도 3을 참조하면, 본 발명의 일실시예에 따른 스마트폰으로 촬영한 신용카드의 숫자 인식 방법은, 본 발명에서는 2가지의 실행 단계를 가진다.

[0027] 첫 번째는 스마트폰에서의 실행 단계이며, 두 번째는 서버 PC에서의 실행단계이다. 스마트폰과 서버 PC에서의 실행 단계는 병행적으로 이루어진다. 처음 스마트폰에서 어플리케이션을 실행하고(S200), PC에서 영상처리 및 인식 프로그램을 실행한다(S300).

[0028] 먼저, 스마트폰에선 신용카드 사진을 촬영한 후 TCP/IP 통신을 통해 서버 PC로 전송한다(S210).

[0029] 스마트폰에서 촬영한 카드 사진을 전송받은 PC는 카드 이미지를 Gray 변환 및 Equalize(평활화)를 거쳐 전처리 과정을 진행한다(S310).

[0030] 여기서, Gray 변환은 3채널 색상을 가진 이미지를 gray scale의 이미지로 바꾸는 것인데 이는 추후 진행할 평활화 및 edge 추출을 위함이다. 여기서 평활화란 이미지의 밝기 분포가 치우친 것을 넓은 영역에 거쳐 분포하도록 넓혀 주는 것이다. 즉, 밝기 분포를 넓히는 작업이다.

[0031] 이러한 전처리 과정 이후 카드에서 16자리의 숫자 부분의 위치만을 파악하기 위해 ROI(관심영역)을 지정한다(S320).

[0032] 그 후 ROI로 지정된 이미지 부분에서 엣지(Edge)를 추출하게 되고(S330) 이렇게 추출된 엣지를 이용해 수직 투영(vertical projection) 한다(S340).

[0033] 이때, 엣지 추출은 카드 배경을 제외하고 숫자들의 윤곽선을 추출하는 것이다. 엣지 추출 시 sobel operator를

사용하였는데 sobel은 3x3 크기의 행렬을 사용하여 연산 시 중심을 기준으로 각 방향의 앞뒤의 값을 비교해 변화량을 검출하는 알고리즘이다.

[0034] sobel은 밝기의 변화율 즉, 기울기를 검출하기 위한 1차 미분 연산자 중 하나이다. 기울기가 크다는 것은 밝기 값의 변화가 크게 일어난다는 뜻이고 영역의 경계가 존재한다는 것을 의미한다. 이미지에 sobel의 3x3 행렬(mask)을 이용해 이미지의 모든 방향에서 기울기(gradient)를 구하게 되면 이미지에 있는 윤곽선들을 얻을 수 있다.

[0035] 수직 투영을 하는 이유는 이미지의 가로를 기준으로 수직으로 픽셀을 모으게 되면 엣지가 있는 부분과 없는 부분을 구별할 수 있게 되는데 이는 엣지가 없는 부분은 카드 숫자들 사이의 빈 공간을 구별할 수 있기 때문이다.

[0036] 수직 투영 이후 16자리를 4등분 하는데 이는 projection의 결과에서 16자리 숫자 중 1, 5, 9, 13번째 숫자의 시작 부분 엣지와 4, 8, 12, 16번째 숫자의 끝 부분 엣지 위치를 판별하여 4 등분을 진행한다. 이 과정을 통해 16자리 숫자를 4개씩 쪼갠 이미지가 4개 나오게 되고, 이 4개 이미지를 각 이미지의 가로 길이를 4로 나누어 다시 4등분 하여 총 16개 숫자의 엣지 형태 이미지를 생성한다.

[0037] 위의 과정으로 생성된 16개의 이미지는 대부분 양옆의 여백이 없는 상태의 이미지인데 이는 학습과정에서 오류를 발생할 수 있기 때문에 이미지 양옆에 여백을 넣기 위한 작업을 진행한다. 여백을 넣기 위해서 16개의 엣지 이미지를 각각 수직 투영하여 엣지가 시작되는 위치와 끝나는 위치를 판별한다.

[0038] 이 후 4개의 숫자로 이루어진 4개의 이미지와 비교를 진행한다. 즉, 첫 번째 숫자는 첫 등분의 숫자에 있기 때문에 첫 등분의 이미지와 비교하고 5번째 숫자는 2번째 등분에 존재하는 숫자이기 때문에 2번째 등분의 이미지와 비교를 진행한다.

[0039] 이때, 비교를 진행하는데 있어 구현한 메칭(matching) 방법을 이용하는데 이 방법은 비교할 이미지를 비교 대상의 이미지에서 좌에서 우로 한 픽셀씩 이동하면서 정확하게 맞는 위치를 찾아내는 것이다. 비교 대상에서 정확한 위치가 판별되었으면 이 전에 확인하였던 각 숫자의 엣지 시작점에서 좌로 끝점에서 우로 5픽셀씩 추가하여 다시 한 번 분할한다. 이와 같이 분할된 각 이미지들은 양 옆에 여백을 가지는 이미지들도 다시 생성된다.

[0040] 최종 분할까지 진행한 후 이미지를 28x28 크기로 변환하고 변환된 이미지를 파일로 저장한다(S350). 이렇게 저장되는 이미지는 16개가 되며 이 16개의 데이터를 하나로 모아 학습에 필요한 형태의 데이터 파일로 생성한다(S360).

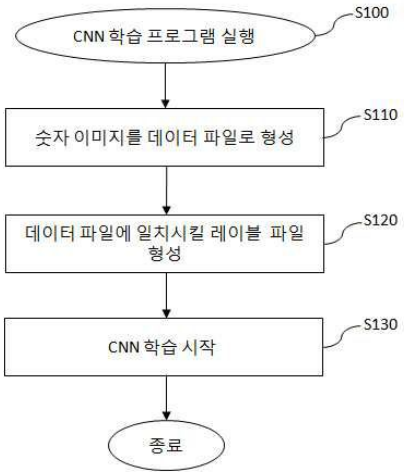
[0041] 위의 과정을 통해 생성된 하나의 데이터 파일과 이전에 생성하였던 학습된 nnt형식의 파일을 불러온 후 숫자를 판별하는데(S370) 숫자를 판별하는데 있어서 앞에 설명한 CNN을 이용하였다. 즉, 최종적으로 만들어진 하나의 파일에는 16개 숫자의 형태가 저장되어 있으며 각 숫자의 특징을 추출하여 이미 학습된 데이터 파일에서 해당 특징을 가지는 것이 어느 숫자 인가를 판별하는 것이다.

[0042] 이처럼 PC에서 판별된 16개의 숫자들은 사용자의 스마트폰으로 전송이 되며 스마트폰 화면에서 판별한 16개의 숫자를 출력한다(S220).

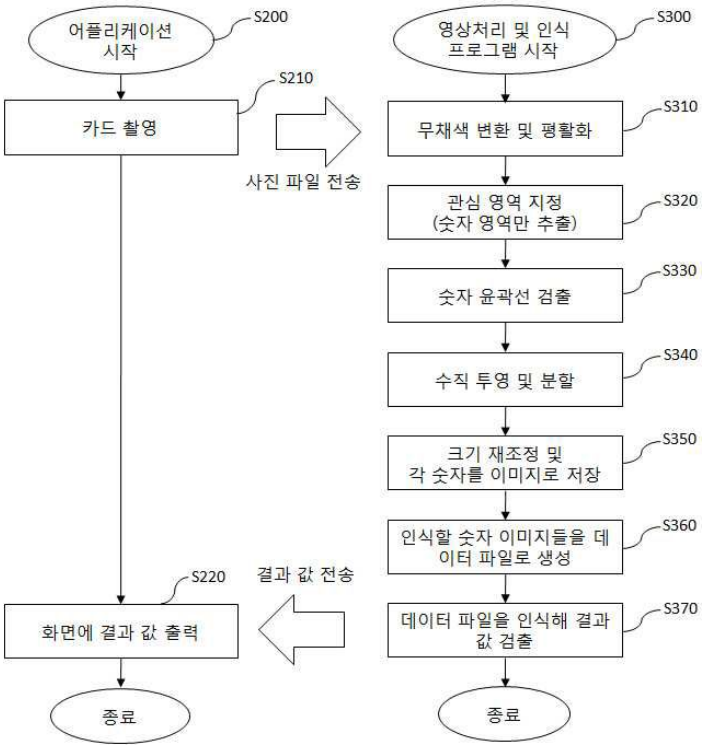
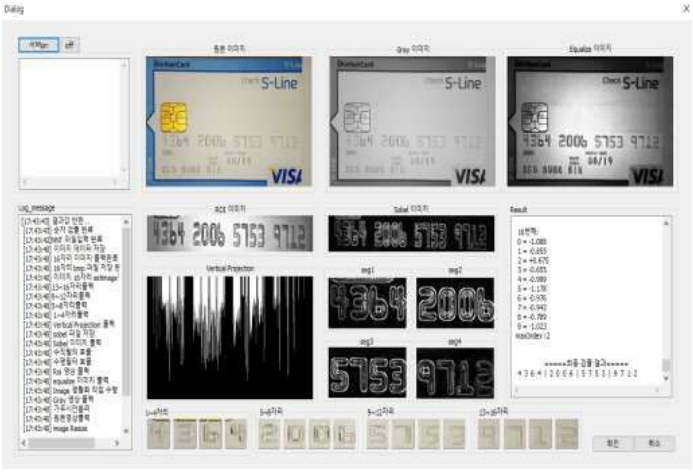
[0043] 이상에서 본 발명의 실시예에 대하여 상세하게 설명하였지만 본 발명의 권리범위는 이에 한정되는 것은 아니고 다음의 청구범위에서 정의하고 있는 본 발명의 기본 개념을 이용한 당업자의 여러 변형 및 개량 형태 또한 본 발명의 권리범위에 속하는 것이다.

도면

도면1


CNN 학습 프로그램 실행
S100
숫자 이미지를 데이터 파일로 형성
S110
데이터 파일에 일치시킬 레이블 파일 형성
S120
CNN 학습 시작
S130
종료

도면2


어플리케이션 시작
S200
카드 촬영
S210
사진 파일 전송
영상처리 및 인식 프로그램 시작
S300
무채색 변환 및 평활화
S310
관심 영역 지정 (숫자 영역만 추출)
S320
숫자 윤곽선 검출
S330
수직 투영 및 분할
S340
크기 재조정 및 각 숫자를 이미지로 저장
S350
인식할 숫자 이미지들을 데이터 파일로 생성
S360
데이터 파일을 인식해 결과 값 검출
S370
결과 값 전송
S220
화면에 결과 값 출력
종료
종료

도면3