



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2023-0130270
(43) 공개일자 2023년09월12일

(51) 국제특허분류(Int. Cl.)
G06V 10/764 (2022.01) G06V 10/44 (2022.01)
(52) CPC특허분류
G06V 10/764 (2023.08)
G06V 10/449 (2023.08)
(21) 출원번호 10-2022-0027206
(22) 출원일자 2022년03월03일
심사청구일자 2022년03월03일

(71) 출원인
조선대학교산학협력단
광주광역시 동구 필문대로 309 (서석동)
(72) 발명자
권구락
광주광역시 서구 화운로 24 힐스테이트 303동 303호
(74) 대리인
특허법인주원

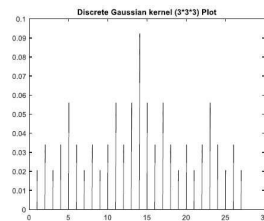
전체 청구항 수 : 총 10 항

(54) 발명의 명칭 심층신경망을 위한 가우스 활성화 매개 변수 레이어를 이용한 이미지 분류 또는 인식방법

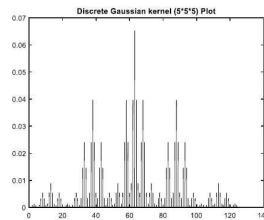
(57) 요약

본 발명은 심층신경망을 위한 가우스 활성화 매개 변수 레이어를 이용한 이미지 분류 또는 인식방법에 관한 것이다.

대표도 - 도1



(a)



(b)

(52) CPC특허분류

G06T 2207/10088 (2013.01)

G06V 2201/03 (2022.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	1345333992
과제번호	2021R1I1A3050703
부처명	교육부
과제관리(전문)기관명	한국연구재단
연구사업명	이공학학술연구기반구축(R&D)
연구과제명	치매케어를 위한 ATN(V), MRI-PET, 커넥툼 기반 진단 지원 및 진행 예측 기술
기 여 율	1/1
과제수행기관명	조선대학교산학협력단
연구기간	2021.06.01 ~ 2022.02.28

명세서

청구범위

청구항 1

- 1) 이미지를 입력하는 단계;
- 2) 심층신경망(deep neural network) 계층을 형성하는 단계;
- 3) 가우시안 활성화 파라미터(GAP)로 특징 벡터를 정규화 단계;
- 4) 비선형 함수를 이용하여 수정하는 단계;
- 5) 최대 풀링(Max pooling)하는 단계;
- 6) 상기 2) 내지 5)를 1회 이상 반복하는 단계;
- 7) 완전 연결된 계층(FCL)을 형성하는 단계;
- 8) 드롭아웃(Dropout) 단계;
- 9) 상기 7) 및 8)
- 10) 분류 결과를 출력하는 단계;를 포함하는 이미지 분류 또는 인식방법

청구항 2

제1항에 있어서, 상기 이미지는 2차원 또는 3차원 이미지인 것을 특징으로 하는 이미지 분류 또는 인식방법

청구항 3

제1항에 있어서, 상기 이미지는 동물 사진, Caltech-102 이미지, CIFAR-10 이미지 및 자기공명영상이미지(MRI)로 이루어진 군에서 선택된 하나 이상인 것을 특징으로 하는 이미지 분류 또는 인식방법

청구항 4

제1항에 있어서, 상기 심층신경망은 컨볼루션 신경망인 것을 특징으로 하는 이미지 분류 또는 인식방법

청구항 5

제1항에 있어서, 상기 3)은 언샤프닝(un-sharpening) 프로세스를 포함하는 것을 특징으로 하는, 이미지 분류 또는 인식방법

청구항 6

제5항에 있어서, 상기 언샤프닝(un-sharpening) 프로세스는 i) 입력 이미지 볼륨과 가우스 커널의 상관 연산; ii) 마스크 커널을 생성하기 위해 흐릿한 버전으로 원본 입력 이미지 볼륨을 추출; 및 iii) 원래 입력에 마스크된 버전 추가를 포함하는 것을 특징으로 하는, 이미지 분류 또는 인식방법

청구항 7

제1항에 있어서, 상기 비선형 함수는 정류 선형 유닛(ReLU) 및 누출 정류 선형 유닛(Leaky ReLU)으로 이루어진 군에서 선택된 하나인 것을 특징으로 하는, 이미지 분류 또는 인식방법

청구항 8

제1항에 있어서, 상기 6)은 2회 내지 4회 수행되는 것을 특징으로 하는, 이미지 분류 또는 인식방법

청구항 9

제1항에 있어서, 상기 7)은 3회 수행되는 것을 특징으로 하는, 이미지 분류 또는 인식방법

청구항 10

제1항에 있어서, 상기 출력은 하기 수학적 식 5를 통해 이루어지는 것인, 이미지 분류 또는 인식방법

[수학적 식 5]

$$Z = \alpha.X + \beta.Xmask + \gamma$$

(상기 Z는 출력; X는 입력; Xmask=X-Xg; Xg는 특징 벡터 X_n 을 가중 평균 방식으로 변환한 선명한 버전의 이미지를 나타내고, 학습 가능한 매개변수 α , β 및 γ 는 각각 X, X_{mask} 및 오프셋의 크기를 조정한다.)

발명의 설명

기술 분야

[0001] 본 발명은 심층신경망을 위한 가우스 활성화 매개 변수 레이어를 이용한 이미지 분류 또는 인식방법을 제공하는 방법에 관한 것이다.

배경 기술

[0003] 딥 뉴럴 네트워크(Deep neural network, DNN)는 1989년 필기 인식을 위해 부상한 CNN(Convolutional Neural Network)인 LeNet-5의 성공 이후 머신 러닝(machine learning, ML) 및 딥 러닝 분야의 다크호스였다[1],[2]. DNN 사용의 큰 성공은 AlexNet[3], GoogleNet[4], ResNet[5], 객체 인식(R-CNN[6],[7]), 장면 분할(SegNet[8],[9]) 및 기타 인간 지각을 위한 지루한 작업을 사용하는 ImageNet에서와 같이 패턴 인식, 이미지 분류를 위한 특징의 정확한 추출에 기여하는 더 많은 학습 가능한 모델 매개변수를 수용할 수 있는 능력 때문이다. 이미지 분류 또는 인식에서 일반적으로 사용되는 DNN은 CNN이며, 컨볼루션 필터는 클래스 식별을 위해 가장자리, 색상, 모서리, 선과 같은 기본 수준 특징에서 질감, 패턴, 모양 등과 같은 상위 수준 특징에 이르기까지 주요 특징 감지기로 사용된다[10]-[12],[31]. 따라서 CNN은 기본적으로 이미지 특징 추출기이다. CNN에서 컨볼루션 필터의 가중치는 훈련의 핵심 매개변수이며 특정 필터의 작동 방식을 결정한다. 컨볼루션 필터 외에도 많은 다른 학습 가능한 레이어가 역전파를 통해 학습하는 동안 가중치 업데이트에 참여하므로 클래스 레이블 속성이 있는 최종 다운 샘플링된 특징을 생성하기 위해 모두 함께 작동한다. 최적화를 위한 훈련 알고리즘 또는 목적 함수 및 초기화 기술은 교차 엔트로피 손실을 최소화하여 훈련을 중지할 수 있는 핵심 구성 요소이다. 즉, 가중치가 더 이상 변경될 수 없고 기술적으로 모델/네트워크 수렴을 이룬다고 한다. 최적의 수렴을 달성하는 것이 주요 목표이지만 가장 낮은 최소값에 도달하는 것만으로는 테스트(보이지 않는) 샘플 분류에서 높은 성공률을 보장할 수 없다. 즉, '테스트 오류'는 낮고 '훈련 오류'는 높을 수 있다. ML에서 이러한 친숙한 문제를 과적합 또는 일반화 오류라고 하며 이는 모든 DNN 아키텍처에서 가장 심각한 문제이다. 이 일반화 오류를 줄이기 위해 드롭아웃, 조기 중지, 무작위 샘플링 등과 함께 많은 정규화 기술이 도입되었다. 그러나 여러 벤치마크 데이터 세트의 최첨단 알고리즘으로 수많은 ML 알고리즘이 아직 완전히 이해되지 않고 많은 작업에서 블랙박스로 작동하고 있기 때문에 많은 표준 DNN이 일반화하지 못하는 사실을 통해 실현할 수 있다[13],[36],[37].

[0004] 게다가, 스크래치 훈련 중인 모든 NN은 매개변수 초기화의 임의성에 영향을 받는다. 이는 훈련 시간 동안 계층의 입력 노드 분포에 약간의 불일치를 가져오고 공변량 이동으로 간주되는 포화 비선형성으로 네트워크를 훈련시키는 것을 극도로 어렵게 만든다[14]. 특히 내부 공변량 이동으로 간주되는 이 현상은 수렴과 일반성에 대한 일종의 저주이기도 하다. 이를 극복하기 위해 이 공변량 이동의 영향을 최소화하기 위해 배치 정규화(batch normalization, BN)가 도입되었다. 또한 훈련 과정의 속도를 높이고 그라디언트를 폭발시키지 않고 더 높은 학습률에 대처할 수 있다. BN은 계층별 미백 기술 이미지, 즉 정규화 및 역상관을 위한 평균 0 및 단위 분산을 사용하며 활성화당 2개의 추가 매개변수만 스케일링용이고 다른 하나는 이동용이다. 훈련 시간을 줄이고 네트워크의 해석 능력을 보존하는 데 도움이 된다[15]. 컨볼루션 가중치가 너무 빨리 업데이트되어 가중치가 더 빠르게 수렴 방향으로 이동하는 경향이 있는 반면 유효성 검사 정확도는 여전히 훨씬 뒤떨어지기 때문일 수 있다. BN은 주로 빠른 훈련과 내부 공변량 이동을 줄이기 위해 설계되었으며, 주장된 정규화 기술은 드롭아웃 사용을 제거한다고 한다. 이 때문에 BN은 dropout이 동일하게 하므로 일반화 오류를 줄인다고 주장한다. 그러나 동시에 기

본 아키텍처에서 BN을 사용할 때 눈에 띄는 과적합 문제가 관찰되었다. BN과 함께 최근에 분류 목적으로 사용되는 다른 많은 활성화 함수는 최적의 모델 피팅과 과적합 위험이 적은 학습 가능한 매개변수로 설계되었다[16]. DNN에서 미니배치 평균을 사용하지 않고 정규화 계층을 설계하기 위해 최근에 수행된 유사한 작업에는 필터 응답 정규화(filter response normalization, FRN)[38], 그룹 정규화(group normalization, GN)[39] 및 계층 정규화(layer normalization, LN)[40]가 있다. 이 모든 방법은 배율 계산을 위한 미니 배치 종속성을 피하기 위해 배치 차원에서 작동하지 않고 활성화 맵 채널 통계만 사용한다. 그러나 그들 중 누구도 본 발명에서 제안한 방법으로 레이어의 선명 효과를 위해 가우스 필터 또는 이미지 필터와 같은 필터링 기능을 사용하지 않는다. 따라서 본 발명은 과적합 및 공변량 이동의 이러한 문제를 해결하기 위해 활성화 매개변수와 함께 정규화 계층을 설계하는 데 관심이 있었다. 실험을 위해 인코더 기반 네트워크의 4개 레이어를 각 인코더에 정규화 레이어가 있는 기본 아키텍처로 사용하고 있다. 또한 기본 모델과 아키텍처 면에서 유사한 로컬 응답 정규화[3]가 있는 AlexNet도 분류 결과를 비교하기 위해 모든 데이터 세트를 스캐치 훈련하는 데 사용된다. 제안하는 방법은 CNN normalization layer와 matrix equivalency를 통해 수학적으로 설명되었다. 제안된 레이어에서 활성화당 3개의 매개변수를 사용한다. 하나는 원래 특징의 크기를 조정하고 다른 하나는 마스크된 특징의 크기를 조정하며 마지막 매개변수는 출력을 이동하기 위한 오프셋 또는 바이어스로 사용한다.

[0005] 본 발명은 심층 신경망과 통합하기 위해 새로운 가우시안 필터 기반 정규화 레이어를 제안한다. 이 레이어를 가우시안 활성화 매개변수(Gaussian activated parametric, GAP) 레이어라고 한다. 또한 Minibatch Averaged scaling을 수행하는 대신 계층 내 정규화를 위해 동일한 채널 평균 스케일링을 제안한다.

[0006] 또한, 본 발명은 특징 추출, 히스토그램 분석, 상관 테스트를 통한 내부 공분산 문제 및 과적합을 연구하기 위해 최종 분류 결과의 비교와 함께 기본 아키텍처에서 BN 대신 제안된 GAP 계층에 대해 비교 분석을 수행하였다.

[0007] CNN의 초기 레이어에 있는 컨볼루션 필터 가중치는 네트워크가 완전히 훈련되면 일부 이미지 필터를 모방하는 경향이 있는 것으로 관찰되었다. Krizhevsky et al.[3]에서 훈련된 convolutional weights/parameters는 대부분 edge detector와 color filter였다. 에지 필터는 수평, 수직 및 대각선 에지를 감지할 수 있다. 게다가 이러한 필터는 본질적으로 병진 불변이므로 모든 입력 이미지에 대해 공간적으로 작동한다. Kaiming et al.[16]은 첫 번째 컨볼루션 레이어의 필터가 edge 또는 texture detector와 같은 대부분 Gabor와 유사한 필터였으며 전체 교육 후 결과에 따르면 필터의 긍정적인 응답과 부정적인 응답이 모두 존중되는 것으로 나타났다. 기본 이미지 필터와 부정적인 응답에 관한 이러한 이전 결과는 훈련 중에 똑같이 중요했기 때문에 조기 특징 감지 및 빠른 모방을 위한 레이어를 설계하는 데 관심이 있었다. 이를 위해 다른 필터를 수동으로 초기화하는 대신 가장자리, 색상, 얼룩 등과 같은 단일 특징뿐만 아니라 전체 결과를 매끄럽게 만드는 필터를 찾는 데 관심이 있었다. 본 발명에서는 Sobel, Prewitt, Robert 등[17]-[20]의 edge detection filter의 사용은 특징 분산 없이 단방향으로 작동하며 분류 결과도 좋지 않았다. 따라서 학습된 특징을 향상하고 가장자리, 얼룩, 색상, 모서리 등과 같은 모든 유형의 특징 속성에 대해 미미하고 영리하게 작업할 수 있는 범용 필터가 필요하다. 이를 위해 가우시안 필터를 선택했다. 가우시안 필터는 정규화된 입력과 함께 평균 평활화 함수로 사용할 수 있다. 그러나 일괄 정규화와 함께 가우스 평활화를 사용하는 것은 적절하지 않아 특징을 정규화하기 위해 필터/채널 평균 정규화를 사용한다. 이 정규화에서는 미니 배치 속성이 포함되지 않고 대신 모든 필터 가중치(즉, 모든 채널의 컨볼루션 가중치)의 평균 특징이 실시예에서 나중에 자세히 설명하는 것처럼 각 배치 차원에 대해 채널별로 정규화된다. 따라서 얻은 평균 특징 벡터는 미니 배치 속성을 혼합하지 않고 채널 이미지에 대해서만 작동한다. 이렇게 하면 노이즈 주입이 감소하므로 단일 이미지 속성만 요약된다. 제안된 레이어를 특징 향상기로 사용하기 때문에 활성화된 이미지 출력에 대해 작동하기 위해 컨볼루션 레이어를 따른다.

[0008] NN에서 임의성의 효과를 이해하기 위한 실험[21]에서 가우스 함수를 사용하여 입력에 노이즈를 추가하고 소스 이미지에 임의의 픽셀을 추가하여 신호에 불균일성을 가져왔다. 이것은 레이블과 훈련 픽셀의 관계를 파괴한다. 실험을 더 무작위로 만들기 위해 이미지 픽셀을 다시 섞고 가우스 분포에서 다시 샘플링했다. 그러나 NN은 이에 크게 영향을 받지 않았고 테스트 세트에 적합할 수 있었다. 훈련 중에 사용된 확률적 경사 하강 알고리즘[22] 덕분이다. 본 발명은 가우스 함수를 노이즈 생성기로 사용하지 않고 언샤프닝(un-sharpening) 프로세스를 위한 마스크 생성기로 사용했으며 채널 정규화[41] 작업은 그렇게 하도록 동기를 부여했다. 본 발명은 각 계층의 입력과 출력 사이의 상관 속성을 파괴하지 않고 좋은 훈련 수렴을 얻는 것을 목표로 했다. 전체 이미지에 대해 가우시안 평활화를 수행하는 대신 가우시안 커널을 사용하여 원본 이미지 커널과 일치하는 평균과 분산을 생성했다. 커널 크기는 이전 컨볼루션 필터 커널의 크기에 따라 결정되므로 평활화 과정에서 필터 속성이 섞이지 않는다.

[0009] 본 발명은 특히 CNN을 위한 심층 신경망을 위한 새로운 가우스 활성화 매개변수(GAP) 레이어를 제안한다. 이 레

이어는 가우스 필터 기반 언샤프닝 기술을 사용하여 특징 벡터를 정규화한다. 본 발명은 정규화를 위해 제안된 레이어를 사용하여 배치 정규화의 결과와 거의 유사한 결과를 제공하며 일부 경우에는 더 높은 훈련 시간을 희생시키면서 결과가 약간 더 낫다는 것을 보여준다. 제안된 레이어의 결과를 보여주기 위해 인코더 기반 네트워크의 4개 레이어를 분류를 위한 기본 아키텍처로 사용하고 있다. 여기서 처음 두 개의 정규화 레이어는 GAP 레이어 또는 BN 레이어이고 나머지는 BN 레이어이다. 실험 결과는 ADNI MRI(3D CNN 정확도: 93.58% vs. 91.89%) 및 CIFAR-10(2D CNN 정확도: 75.21% vs. 75.11%)과 같은 대용량 데이터 세트에서 정규화 계층으로 두 개의 GAP 레이어를 분류하는 것이 더 좋고, 5마리 동물 데이터 세트(2D CNN 정확도: 62.92% 대 58.48%)와 같은 더 작은 데이터 세트에서 첫 번째 BN 레이어를 GAP로 교체하면 결과가 더 좋다. 따라서 더 작은 데이터 세트에는 단일 GAP 계층 정규화를 사용하고 더 큰 데이터 세트 교육에는 두 개의 GAP 정규화 계층을 사용하는 것이 좋습니다. 또한 제안된 방법은 스크래치 훈련에서 교차 채널 정규화 기반 AlexNet 네트워크보다 더 나은 결과를 생성했다.

선행기술문헌

특허문헌

- [0011] (특허문헌 0001) 1. 한국공개특허 제2021-0067592호
(특허문헌 0002) 2. 한국공개특허 제2021-0082617호

발명의 내용

해결하려는 과제

- [0012] 본 발명은 심층 CNN의 초기 및 중간 특징 레이어를 정규화하고 활성화하여 맞춤형 레이어가 특징을 보다 잘 구별하고 레이어가 대상 도메인 분류를 위해 원활하게 조정되도록 하는 것을 목적으로 한다.
- [0013] 보다 구체적으로, 본 발명은 심층신경망을 위한 가우스 활성화 매개 변수 레이어를 이용한 이미지 분류 또는 인식방법을 제공하는 것을 목적으로 한다.

과제의 해결 수단

- [0015] 청구항 확정 후 작성 예정

발명의 효과

- [0017] 본 발명에 따른 심층신경망을 위한 가우스 활성화 매개 변수 레이어를 이용한 이미지 분류 또는 인식방법은 훈련시간의 단축과 정확한 패턴 인식 및 이미지 분류를 통해 우수한 출력 결과를 제공하는 효과가 있다.

도면의 간단한 설명

- [0019] 도 1은 선형 플롯에서 가우스 3D 커널의 시각화 (a) 27개의 가중치 행렬이 있는 3*3*3 커널 (b) 125개 가중치 행렬이 있는 5*5*5 커널을 나타낸 것이다. 각 해당 봉우리에 대해 더 높은 중심 값을 확인할 수 있다.
- 도 2는 (a) 5마리 동물 데이터 세트 (b) Caltech-102 (c) CIFAR-10 (d) 3D MRI_baseline에 대한 동일한 훈련 환경에 대해 서로 다른 정규화 체계를 사용한 훈련 및 검증 정확도 곡선을 나타낸 것이다. 모든 그래프에서 훈련 정확도는 100% BN으로 더 빨리 수렴(100%)에 도달하는 반면 검증 곡선은 더 오랜 시간 동안 훈련 곡선을 따르지 않는다. 이는 가중치 업데이트 프로세스가 더 빨리 끝나고 더 잘 일반화할 수 없음을 의미한다. 50% BN과 75% BN의 경우 훈련 곡선은 최근에 수렴을 달성하고 검증도 마찬가지로이다. 이것은 네트워크가 (동일한 학습 속도로) 가중치를 천천히 업데이트하는 데 도움이 되므로 과적합을 줄일 수 있다(모든 경우에 훈련 및 검증 정확도의 큰 차이로 인해 과적합의 명백한 사례가 있지만 더 나은 아키텍처가 결과를 변경할 수 있음). 그러나 본

발명에서는 기본 아키텍처를 사용하여 제안된 아이디어를 테스트한다. 여기에서 GAP 레이어가 있는 CNN은 100% 정확도에 천천히 도달하는 경향이 있어 유효성 검사 정확도가 더 오랜 시간 동안 다루어지는 반면 BN 레이어를 사용하면 훈련 정확도가 빠르게 증가하여 훈련 초기 단계의 훈련 및 검증 계층 사이에서 더 높은 간격을 유발한다(공동으로 제안된 레이어에 대해 GAP라고 명명했다). 따라서 과적합 문제는 배치 정규화에서 아직 완전히 해결되지 않았다. 낮은 학습률과 높은 학습률 하락 요인에도 불구하고 문제가 여전히 우세하다. 이와 관련하여 본 발명은 훈련 수렴이 제안된 GAP 계층으로 더 잘 일반화되기를 바란다. 본 발명의 결과는 테스트 결과가 대부분의 경우 동일하거나 약간 개선되었으며 특징 추출도 향상되었음을 시사한다.

도 3은 비교를 위한 입력 및 출력 히스토그램과 함께 제안된 레이어의 도식적 표현을 나타낸 것이다. 입력 신호는 에지 감지 프로세스를 보여주기 위해 램프 신호로 표시된다. 그러나 본 발명에서 레이어에 대한 입력 X 는 이전 컨볼루션 레이어에서 활성화된 이미지 행렬이다. 입력은 정규화 장치를 통과하여 특성 값의 범위가 좁은 X 의 크기가 조정되고 이동된 버전을 생성한다. 나중에 가우시안 평활화 기능은 특징 벡터 X_g 을 가중 평균 방식으로 변환하여 선명한 버전의 이미지, 즉 X_g 를 생성한다. X 와 X_g 의 차이는 마스킹 벡터 X_{mask} 를 생성한다. 이 벡터는 Z 를 생성하기 위해 다시 원래 X 와 함께 추가된다. 학습 가능한 매개변수 α , β 및 γ 는 각각 X , X_{mask} 및 오프셋의 크기를 조정한다.

도 4는 (a) 첫 번째 컨볼루션 레이어 시각화, (b) (a) 시각화 다음에 오는 첫 번째 BN 레이어, (c) 첫 번째 컨볼루션 레이어 시각화, (d) (c) 시각화 다음에 오는 첫 번째 GAP 계층을 나타낸 것이다. 배치와 GAP의 출력 차이를 각각의 컨볼루션 레이어에서 확인할 수 있다. 배치 속성의 삽입으로 인해 배치 정규화에서 색상이 크게 변경되지만 필터 색상의 급격한 변화 없이 가우스 출력은 동일하게 유지된다. 색상은 유사한 색상과 약간 혼합되어 있으므로 여기에서 스무딩 프로세스가 수행된다.

도 5는 동일한 조건으로 (a) b1b2b3b4 기본 아키텍처 사용 (b) g1b2b3b4 기본 아키텍처 사용 (c) g1g2b3b4 기본 아키텍처 사용 (d) AlexNet 아키텍처를 사용하여 CIFAR-10 스크래치 훈련된 네트워크에서 Deep dream[32]을 사용하여 생성된 10개의 클래스(비행기, 자동차, 새, 고양이, 사슴, 개, 개구리, 말, 배, 트럭)의 훈련된 네트워크에 의해 학습된 클래스별 특성을 나타낸 것이다. 좀 더 자세히 보려면 확대할 수 있다. 코드 구현은 부록을 참조할 수 있다.

도 6은 표 2에 나타낸 바와 같이, 검증 정확도, 테스트 정확도, 최소 95% CI 오류, 최대 95% CI 오류와 함께 비교를 위해 다른 데이터 세트에 대한 분류 결과를 나타낸 것이다. 검증 정확도와 테스트 정확도는 편견을 피하기 위해 동일한 교육 및 테스트 조건으로 동일한 세트에서 계산되었다.

도 7은 테스트 단계에서 5마리 동물 데이터 세트의 샘플 이미지에 대한 계층별 필터 응답 시각화 및 히스토그램 플롯을 나타낸 것이다. 7.1(a) b1b2b3b4의 b1층에 입력된 모든 필터의 히스토그램, 7.1(b) 층 b1에 있는 32개 필터의 평균 필터 응답, 7.1(c) 레이어 b1에서 출력된 모든 특징의 히스토그램, 7.2(a) g1g2b3b4의 레이어 g1에 대한 모든 필터 입력의 히스토그램, 7.2(b) 레이어 g1에 있는 32개 필터의 평균 필터 응답, 7.2(c) 레이어 g1에서 출력된 모든 특징의 히스토그램, 7.3(a) b1b2b3b4의 레이어 b2에 대한 모든 필터 입력의 히스토그램, 7.3(b) 층 b2에 있는 64개 필터의 평균 필터 응답, 7.3(c) 레이어 b2에서 출력된 모든 특징의 히스토그램, 7.4(a) g1g2b3b4의 레이어 g2에 대한 모든 필터 입력의 히스토그램, 7.4(b) 층 g2에 있는 64개 필터의 평균 필터 응답, 7.4(c) 레이어 g2에서 출력된 모든 피쳐의 히스토그램, 7.1(c) 및 7.3(c)에서 배치 정규화 레이어 히스토그램의 출력은 피쳐 분포가 입력(예: 7.1(a) 및 7.3(a))과 완전히 다르다는 것을 보여준다. 각각 가중치 값의 빈도는 입력에서 평균 범위에 더 많았지만 출력에서는 나중에 평균 주변 가중치가 감소했다. 분포의 이러한 급격한 변화는 각각 상관 계수 90.46 및 84.73과 함께 평균 응답 플롯 7.1(b) 및 7.3(b)에도 표시된다. 그러나 GAP 정규화를 사용하는 동안 출력 픽셀/가중치 분포가 완전히 변경되지는 않는다. GAP 레이어 7.2(c) 및 7.4(c)를 사용한 출력 히스토그램은 각각 7.2(a) 및 7.4(a)의 입력 패턴을 따른다. 또한 평균 플롯 응답은 입력 필터 평균으로 매우 높은 수렴을 보여주고 상관 값은 각각 94.6과 91.3이다. 코드 구현은 부록을 참조할 수 있다.

도 8은 5마리 동물 데이터 세트의 모든 테스트 이미지에 대한 정규화 계층의 입력과 출력 간의 상관 값 플롯을 나타낸 것이다. 여기 레이어 2에서 BN(b1) 레이어는 모든 테스트 세트에 대해 약 90%의 상관 값을 생성하는 반면 GAP 레이어(g1)는 b1보다 약간 더 높은 상관 값을 갖는다. 두 번째 정규화 계층, 즉 계층 6에서 BN 계층(b2)은 입력과 모든 테스트 세트의 넓은 범위(94%에서 22%까지)에 대해 매우 낮은 상관 출력을 생성하지만, g2의 출력은 입력과 크게 상관 관계가 없으므로 입력과 약 90% 상관 관계의 범위에 있다. 입력 X 는 선행 컨볼루션 계층의 출력이고 출력 Z 는 정규화 계층의 출력이다. 레이어 상관 값이 매우 낮으면 레이어가 해당 특징을 역상관

했음을 의미한다. 그러나 완벽한 상관관계도 쓸모가 없다.

도 9는 다른 아키텍처 9.1(a) b1b2b3b4의 첫 번째 FCL 특징 9.1(b) b1b2b3b4 훈련된 네트워크의 두 번째 FCL 특징 9.1(c) b1b2b3b4 훈련된 네트워크의 세 번째 FCL 특징의 MRI_baseline 296 테스트 대상에 대한 훈련된 네트워크의 T-SNE 투영을 나타낸 것이다. 9.2(a) g1b2b3b4 훈련된 네트워크의 첫 번째 FCL 특징, 9.2(b) g1b2b3b4 훈련된 네트워크의 두 번째 FCL 특징, 9.2(c) g1b2b3b4 훈련된 네트워크의 세 번째 FCL 특징, 9.3(a) g1g2b3b4 훈련된 네트워크의 첫 번째 FCL 특징, 9.3(b) g1g2b3b4 훈련된 네트워크의 두 번째 FCL 특징, 9.3(c) g1g2b3b4 훈련된 네트워크의 세 번째 FCL 특징을 나타낸 것이다. 첫 번째 열에서 첫 번째 FCL 계층의 활성화된 특징 1728개, 즉 fc_1 은 모든 테스트 세트에 대해 x 및 y 차원에서 2개의 주성분(principal component, PC)으로 투영된다. 중간 열 864에서 두 번째 FCL의 활성화된 특징, 즉 fc_2 레이어는 모든 테스트 세트에 대해 x 및 y 차원에서 2개의 PC에 투영된다. 마지막 열에서 세 번째 FCL 계층의 클래스별 활성화 특징, 즉 fc_3 은 모든 테스트 세트에 대해 x 및 y 차원의 2개의 주성분으로 투영된다. 착색된 점의 분포는 1차 FCL 층에서 뚜렷하게 분리되지 않고 이후 2차 및 3차 FCL 층에서 분리되기 시작한다. 서로 다른 색상으로 겹친 영역이 오류 영역이다. 9.4는 (a) b1b2b3b4 네트워크 분류 (b) g1b2b3b4 네트워크 분류 (c) g1g2b3b4 네트워크 분류를 사용하여 MRI_baseline 테스트 세트 분류에 대해 표시된 단일 그래프의 모든 3개 클래스에 대한 ROC 곡선을 나타낸 것이다.

도 10은 자연 이미지에 대한 다양한 시각화 알고리즘을 사용한 특징 감지 히트맵 비교 결과를 나타낸 것이다. 테스트 이미지에 열지도를 연속적으로 생성하는 데 사용되는 기술은 LIME[33], Occlusion[34], Grad-Cam[35]이다. 전반적으로 AlexNet[3]은 분류에 영향을 미치는 영역인 히트맵 영역이 좁고, 유사하게 g1g2b3b4와 g1b2b3b4의 히트맵 영역은 BN만을 사용한 것(b1b2b3b4)보다 더 넓고 정확하다. GAP 정규화를 사용하여 수행된 더 나은 특징 감지 프로세스를 의미한다.

도 11은 데이터세트(아키텍처)에 대한 정오분류표를 나타낸 것이다.

발명을 실시하기 위한 구체적인 내용

이하, 실시예를 통하여 본 발명을 더욱 상세히 설명하기로 한다. 이들 실시예는 단지 본 발명을 예시하기 위한 것이므로, 본 발명의 범위가 이들 실시예에 제한되는 것으로 해석되지 않는다.

<실시예 1> Gaussian Filter and un-sharpening process

가우시안 필터[23]는 공간 가중 평균 필터로, 모든 이미지-픽셀 분포에 대한 점 확산 함수로 작동한다. 그것은 픽셀의 정규 분포를 생성하는 중앙 픽셀에 더 높은 가중치를 갖는 불균일한 저역 통과 필터로 작동한다. 그리고 회전 대칭 특성은 이미지 형태학적 작업에 방향성 편향성을 제공한다. 가우시안 필터의 이 속성은 평균 값을 사용하여 이미지 속성에 대한 평활화 결과를 생성하면서 가장자리와 밝기를 약간 보존하는 데 도움이 된다. 그러나 입력 표준 편차(standard deviation, SD)(σ)는 이미지를 흐리게 하는 데 약간의 영향을 미친다. T. Lindeberg[24]는 최적의 이산화된 가우시안 커널을 유도하고 가보 함수가 가우스 도함수와 매우 유사하게 보일 수 있음을 입증했다. CNN 특징 행렬에서 작동하려면 커널의 이산 근사치가 필요하다. 이것은 i , j 및 k 가 행렬 행, 열 및 깊이를 나타내고 σ^2 는 표준 가우스 출력에 대해 1과 동일한 입력 분산인 수학적 1에서와 같이 수행된다. $3 \times 3 \times 3$ 및 $5 \times 5 \times 5$ 행렬의 출력은 도 1에 나와 있다.

수학식 1

$$h(i, j, k, \sigma) = \frac{1}{(\sqrt{2\pi}\sigma)^3} e^{-\frac{(i^2+j^2+k^2)}{2\sigma^2}}$$

- [0027] $1/\sqrt{2\pi}$ 라는 용어는 지수 함수에 대한 적분이 1이 아니라는 사실에서 비롯된 정규화 상수이다. 가우시안 커널을 사용한 컨볼루션은 선형 연산[23]이므로 시스템에 비선형성을 가져오려면 수정을 위해 Leaky ReLU와 같은 일부 비선형 함수를 추가해야 한다. 3D 커널은 2D 및 3D CNN 모두에 대한 GAP 레이어 설계에 사용되며, 2D CNN에서 필터는 모든 컨볼루션 기반 이미지의 평균으로 작동하는 반면 3D CNN에서는 평균 값이 전체 볼륨 자체에서 작동한다. 그러나 두 경우 모두 학습 계수 매개변수는 최종 출력에 대한 채널 필터의 평균 효과를 확인하여 별도로 업데이트 된다.
- [0028] 언샤프닝(Un-sharpening) 프로세스는 다음의 3단계를 포함한다. i) 이미지의 흐림, 즉 입력 이미지 볼륨과 가우스 커널의 상관 연산; ii) 마스킹 커널을 생성하기 위해 흐릿한 버전으로 원본 입력 이미지 볼륨을 추출; iii) 원래 입력에 마스크된 버전 추가. 도 3은 레이어 내부에 관련된 작업을 자세히 보여준다.
- [0030] <실시예 2> Proposed Gaussian Activated Parametric (GAP) Normalization Layer
- [0031] A. Architecture and Training
- [0032] 제안된 가우시안 활성화 매개변수(g1) 레이어에 대한 입력 역할을 하는 기본 아키텍처(표 1 참조)에서와 같이 첫 번째 컨볼루션 레이어의 출력을 고려한다. 이 입력은 $X = [227, 227, 32, 64] = [\text{image_row}, \text{image_col}, \text{channel_size}, \text{minibatch_size}]$ 와 같은 크기의 2D CNN용 4D 배열(3D CNN용 5D 배열)로 나타낼 수 있다. 여기에서 입력 X 는 32개의 필터 출력에서 각각 크기가 227×227 인 2D 이미지, 즉 동일한 이미지에 대해 32개의 서로 다른 활성화된 이미지가 있는 단일 이미지의 $227 \times 227 \times 32$ 볼륨을 포함하고 마지막 차원인 4차원은 훈련용 미니 배치 번호를 나타낸다. 각 미니배치는 여러 클래스에 대해 서로 다른 이미지를 나타내므로 총 64개의 서로 다른 입력 이미지가 있으며 첫 번째 GAP 계층에서 순방향 전파를 위한 일괄 입력으로 105.53M 픽셀/가중치를 사용한다. 마찬가지로 2차 레이어의 경우 풀링 후 출력 크기가 113×113 으로 줄어들어 2차 컨볼루션 레이어의 출력은 $113 \times 113 \times 64 \times 64$ (2차 컨볼루션의 필터 64개)이므로 총 52.3M 개의 가중치 입력 두 번째 GAP 레이어의 경우에 해당한다. 더 명확하게 하기 위해 입력 X 또는 첫 번째 GAP 입력은 수식 2에서 다음 블록 행렬로 나타낼 수 있다. 여기서 X_b^n 는 훈련의 b번째 미니 배치에서 n번째 필터의 활성화된 이미지를 나타낸다.
- [0033] 표 1은 기본 아키텍처는 분류를 위해 BN에 대해 제안된 방법을 테스트하는 데 사용된다. 2D 및 3D 아키텍처는 활성화 크기에 따라 다르다. 참고: 여기에서 'g1g2b3b4' 아키텍처는 1차 정규화 GAP(g1), 2차 GAP(g2) 및 3차, 4차 BN을 각각 b3 및 b4로 나타낸다. 마찬가지로 b1b2b3b4는 모든 BN을 의미한다. 하이퍼파라미터 및 활성화 기능의 선택은 이전 작업[12]을 기반으로 한다.

표 1

S. N	DESCRIPTION (2D BASE ARCHITECTURE)	2D ACTIVATION SIZE	2D LEARNABLE PARAMETERS	DESCRIPTION (3D BASE ARCHITECTURE)	3D ACTIVATION SIZE	3D LEARNABLE parameters
1	Input Image	227×227×3		64×64×64 images with 'zero center' normalization	64×64×64	
Encoder-1	Convolution	32 1×3 convolutions with stride [1 1] and padding 'same'	Weights- 3×3×32 Bias- 1×1×32	64 3×3×1 convolutions with stride [1 1 1] and padding 'same'	64×64×64×64	Weights- 3×3×3×64 Bias- 1×1×1×64 Alpha- 1×1×1×64 Beta- 1×1×1×64 Gamma- 1×1×1×64 OR Offset- 1×1×1×64 Scale- 1×1×1×64
	Batch Normalization (b1) or GAP Normalization (g1)	Normalization or activated Normalization	Alpha- 1×1×32 Beta- 1×1×32 Gamma- 1×1×32 OR Offset- 1×1×32 Scale- 1×1×32	Normalization or activated Normalization	64×64×64×64	
	Leaky ReLU	Leaky ReLU with scale 0.01		Leaky ReLU with scale 0.01	64×64×64×64	
	Max Pooling	2×2 max pooling with stride [2 2] and padding [0 0 0 0]		2×2×2 max pooling with stride [1 1 1] and padding [0 0 0 0 0 0 0 0]	63×63×63×64	
Encoder-2	Convolution	64 5×5 convolutions with stride [1 1] and padding 'same'	Weights- 5×5×32×64 Bias- 1×1×64	64 5×5×5×64 convolutions with stride [1 1 1] and padding 'same'	63×63×63×64	Weights- 5×5×5×64×64 Bias- 1×1×1×64 Alpha- 1×1×1×64 Beta- 1×1×1×64 Gamma- 1×1×1×64 OR Offset- 1×1×1×64 Scale- 1×1×1×64
	Batch Normalization (b2) or GAP Normalization (g2)	Normalization or activated Normalization	Alpha- 1×1×64 Beta- 1×1×64 Gamma- 1×1×64 OR Offset- 1×1×64 Scale- 1×1×64	Normalization or activated Normalization	63×63×63×64	
	Leaky ReLU	Leaky ReLU with scale 0.01		Leaky ReLU with scale 0.01	63×63×63×64	
	Max Pooling	2×2 max pooling with stride [2 2] and padding [0 0 0 0]		2×2×2 max pooling with stride [2 2 2] and padding [0 0 0 0 0 0 0 0]	31×31×31×64	
Encoder-3	Convolution	64 7×7 convolutions with stride [1 1] and padding 'same'	Weights- 7×7×64×64 Bias- 1×1×64	64 7×7×7×64 convolutions with stride [1 1 1] and padding 'same'	31×31×31×64	Weights- 7×7×7×64×64 Bias- 1×1×1×64 Offset- 1×1×1×64 Scale- 1×1×1×64
	Batch Normalization (b3)	Batch normalization	Offset- 1×1×64 Scale- 1×1×64	Batch normalization with 64 channels	31×31×31×64	
	Leaky ReLU	Leaky ReLU with scale 0.01		Leaky ReLU with scale 0.01	31×31×31×64	
	Max Pooling	2×2 max pooling with stride [2 2] and padding [0 0 0 0]		2×2×2 max pooling with stride [3 3 3] and padding [0 0 0 0 0 0 0 0]	10×10×10×64	
Encoder-4	Convolution	128 9×9 convolutions with stride [1 1] and padding 'same'	Weights- 9×9×64×64 Bias- 1×1×64	64 9×9×9×64 convolutions with stride [1 1 1] and padding 'same'	10×10×10×64	Weights- 9×9×9×64×64 Bias- 1×1×1×64 Offset- 1×1×1×64 Scale- 1×1×1×64
	Batch Normalization (b4)	Batch normalization	Offset- 1×1×64 Scale- 1×1×64	Batch normalization with 64 channels	10×10×10×64	
	Leaky ReLU	Leaky ReLU with scale 0.01		Leaky ReLU with scale 0.01	10×10×10×64	
	Max Pooling	2×2 max pooling with stride [2 2] and padding [0 0 0 0]		2×2×2 max pooling with stride [4 4 4] and padding [0 0 0 0 0 0 0 0]	3×3×3×64	
18	Fully Connected	1152 fully connected layer	Weights- 1152×25088 Bias- 1152×1	1728 fully connected layer	1×1×1×1728	Weights- 1728×1728 Bias- 1728×1
19	Dropout	50% dropout		50% dropout	1×1×1×1728	
20	Fully Connected	576 fully connected layer	Weights- 576×1152 Bias- 576×1	864 fully connected layer	1×1×1×864	Weights- 1728×864 Bias- 864×1
21	Dropout	50% dropout		50% dropout	1×1×1×864	
22	Fully Connected	N fully connected layer where 'N' is the number of trained class	Weights- N×576 Bias- N×1	3 fully connected layer	1×1×3	Weights- 3×64 Bias- 3×1
23	SoftMax	SoftMax		SoftMax	1×1×3	
24	Output	Cross-entropy		Cross-entropy with 'ADP and 2 other classes	1×1×3	

수학식 2

$$X = \begin{bmatrix} X_1^1 & \cdots & X_1^{32} \\ \vdots & \ddots & \vdots \\ X_{64}^1 & \cdots & X_{64}^{32} \end{bmatrix}$$

[0039] 여기서 평균값 X_m 은 3차원, 즉 필터 또는 채널의 수를 기반으로 계산되며(3D CNN의 경우 4차원에서 계산됨) 행렬로 나타낼 수 있습니다.

수학식 3

[0041]

$$\text{i.e., } X_m = \begin{bmatrix} X_1^1 & \dots & X_1^{32} \\ \vdots & \ddots & \vdots \\ X_{64}^1 & \dots & X_{64}^{32} \end{bmatrix} = \begin{bmatrix} X_1' \\ \vdots \\ X_{64}' \end{bmatrix}$$

[0043] 여기에서 오른쪽의 열 행렬은 서로 다른 훈련 샘플에서 가져온 각 배치의 평균값을 포함한다. 이제 $X_m = [227 \ 227 \ 1 \ 64]$ 에 대한 차원은 컨볼루션 필터로 '다르게' 활성화된 '같은 이미지'의 평균값을 제공한다. 모든 채널의 경험적 평균 값이므로 훈련 중에 업데이트된 각 배치에 대해 64개의 서로 다른 이미지를 포함하고 훈련 후에 사용되지 않으므로 경험적 평균이다. 유사하게, 표준 편차는 $X_s = [227 \ 227 \ 1 \ 64]$ 에 대해서도 계산된다. 이제 본 발명에서는 X_m 과 X_s 를 $X_f = (X - X_m)/X_s$ 로 사용하여 X 를 다시 중심에 놓고 다시 스케일링한다. 그러면 평균 중심 및 $1/X_s$ 스케일링된 출력 X_f 가 생성된다. 행렬의 모든 산술 연산은 요소별로 동일한 차원 연산이다. 따라서

$X_f = [227 \ 227 \ 32 \ 64]$ 크기이다. 여기서 수학식 3에서 X_1' 은 첫 번째 훈련 이미지 $X_1^1, X_1^2, \dots, X_1^{32}$ 등의

활성화된 출력의 평균이다. 따라서 분산은 명목상 매우 작고, 즉, 일반적으로 $\text{var}(X_b') \approx \text{var}(X_b'')$ 이다. BN에서 훈련이 완료되면 각 BN 레이어는 훈련된 네트워크에 저장된 활성화당 'n'개의 훈련된 평균과 분산을 가지며 나중에 예측하는 동안 입력을 정규화하는 데 사용된다. 그러나 본 발명의 방법에서 레이어는 학습된 평균과 SD를 대신 저장하지 않고 예측 중에 컨볼루션 커널의 가중치가 아래로 전달되므로 각 이미지에 대한 경험적 평균이 각 채널 출력에서 계산되므로 사전 학습된 평균과 SD가 필요하지 않다.

[0044] 채널 정규화[41]는 CNN의 각 채널을 모든 훈련 예제에 대해 독립적으로 표준화하고 결과 입력을 (학습 가능한) 스칼라로 확장 및 이동한다. 단일 훈련 샘플에 대한 인스턴스 정규화[42] 및 BN과 유사하다. 본 발명의 경우 X_f 는 원래 미니배치 입력의 정규화되지 않은 버전이며, 이는 입력 X 의 이동 및 크기 조정된 버전이다. α , β 는 각각 입력 및 비선형화 버전에 대해 별도의 채널 표준화 요소로 작동한다. 그러나 사전 훈련된 α , β 및 γ 값은 각 채널을 개별적으로 스케일링한다. X_f 는 이제 가우시안 커널과 공간적으로 상관된다. 즉, 수학식 4에서와 같이 첫 번째 GAP 레이어에 대한 이전 컨볼루션 필터(여기서는 $3 \times 3 \times 3$) 크기와 동일한 크기의 커널로 필터링이 수행된다. 여기서 h_{www} 는 $w=1$ 에서 3 또는 5로 i , j 및 k 에 대해 수학식 (1)로 계산된 3D 커널에 대한 가우시안 필터 가중치를 나타낸다. 두 번째 GAP 정규화, 즉 $g1g2b3b4$ 아키텍처의 $g2$ 레이어의 경우 사용된 가우스 커널 첫 번째 정규화 계층과 다른 두 번째 정규화 계층의 2차 필터 응답을 얻기 위해 서로 이산 가우시안 커널의 컨볼루션을 통해 얻어진다. 활성화 맵(3차/채널 차원)의 특징 정규화를 수용하기 위해 모든 GAP 레이어에서 3D 커널을 사용했으며 필터링 커널은 전체 X_f 벡터에서 이동한다. 수학식 4에서 'o' 기호는 영상 처리에서 공간 필터링 연산과 유사한 상관 연산을 나타낸다. 수학적으로 상관 프로세스는 곱셈 프로세스 전에 신호가 반전되지 않는다는 점을 제외하고 시간 영역에서 컨볼루션과 동일하다. 여기에서 아이디어는 모든 32×64 이미지가 포함되도록 복제된 방식으로 모든 이미지에 필터링을 수행하는 것이다. 그러나 블러 또는 스무딩 효과는 첫 번째 배치의 $X_1^1, X_1^2, \dots, X_1^{32}$ 등의 각 이미지에 대해 다르게 작동한다. 획득된 가우시안 이미지는 수학식 4와 같이

행렬로 표현될 수 있으며, 여기서 각각의 Xg_b'' 는 가우시안 필터 이후의 이미지를 나타낸다. 적용된 가우시안

커널은 선형 상관을 수행하기 위해 이전 컨볼루션 커널과 동일한 크기이다. 따라서 정규화와 활성화를 단일 단위로 결합하려는 초기 목표는 실행 불가능해 보였으므로 분류를 추가로 지원하기 위해 추가 비선형(활성화) 함수를 사용해야 한다.

수학식 4

$$Xg = \begin{bmatrix} h_{11w} & \cdots & h_{1ww} \\ \vdots & \ddots & \vdots \\ h_{ww1} & \cdots & h_{www} \end{bmatrix} o \begin{bmatrix} Xf_1^1 & \cdots & Xf_1^{32} \\ \vdots & \ddots & \vdots \\ Xf_{64}^1 & \cdots & Xf_{64}^{32} \end{bmatrix}$$

$$= \begin{bmatrix} Xg_1^1 & \cdots & Xg_1^{32} \\ \vdots & \ddots & \vdots \\ Xg_{64}^1 & \cdots & Xg_{64}^{32} \end{bmatrix}$$

그리고 마스크는 배치 속성을 상속하지 않고 각 채널 출력에 대해 원래 버전에서 정규화된 가우시안 신호를 빼는 $X_{mask}=X-Xg$ 로 얻는다. 따라서 마지막으로 레이어의 출력을 다음과 같이 전달한다.

수학식 5

$$Z = \alpha.X + \beta.X_{mask} + \gamma$$

여기서 α , β 및 γ 는 각각 크기가 $[1 \ 1 \ N]$ 인 학습 가능한 매개변수로, 각 'N' 필터에 대해 고유한 값을 가지며 출력 값을 최적화하는 데 도움이 된다. α , β 및 γ 는 초기에 0과 1 사이에서 선택되고 가중치 업데이트의 역전과 동안 기울기 출력을 제어하기 위해 스케일링 계수로 작용한다. 그리고 수학식 5의 두 번째 매개변수 항, 즉 β 는 처음에는 1보다 작으나 역전과 동안 가중치가 업데이트되면 값은 $\beta>1$ 이 되는 경향이 있으며 이 경우 높은 부스팅 필터로 작용하여 연삭 마스크(unsharp masking)의 기여를 강조한다. $\beta<1$ 일 때 샤프닝 마스크가 더 강조된다(도 3 참조). 다른 학습 가능한 매개변수 α 및 γ 도 역전과 동안 업데이트되며, α 는 출력을 $(\beta.X_{mask})$ 와 동일하게 만들기 위한 스케일링 계수로 작동하고 γ 는 레이어 기울기 손실에 영향을 주지 않는 바이어스로 작동한다. 지원 코드와 함께 MATLAB 코드 구현은 부록에 포함되어 있다.

SGD(Stochastic Gradient Descent)는 표준 오류[22]와 같이 전체 학습 세트 오류를 한 번에 사용하지 않고 미니 배치를 사용하여 계산된 학습 가능한 매개변수(가중치, 편향, 오프셋, 계수) 값을 업데이트하는 학습 알고리즘이다[22],[25]. 여기서 손실은 수학식 6에서와 같이 각 반복에서 손실의 음의 기울기로 네트워크의 매개변수를 업데이트하여 미니 배치 세트에서 계산된다.

수학식 6

$$w_l^{t+1} = w_l^t - \gamma^t \cdot \alpha_l \frac{dL}{dw_l^t} + r (w_l^t - w_l^{t-1})$$

[0055]

[0057] 여기서 가중치 또는 편향 또는 오프셋 w_l^{t+1} 업데이트는 각 반복 't+1'에서 레이어 'l'에서 이전 반복 w_l^t 의 가중치를 사용한다. α_l 은 레이어 l의 매개변수에 대한 학습률 하이퍼파라미터이며, 실험에서 값 >0, 초기에 0.001로 유지된다. 본 발명은 또한 모멘텀과 함께 SGD를 사용하고 있기 때문에 매개변수 가중치 업데이트의 진동을 줄이고 최적 값을 향한 가장 가파른 하강 경로를 찾는다. 이를 위해 운동량 비율로 알려진 하이퍼 매개변수 r 을 0.95로 설정한다. 또한, 음의 항은 매 에포크 후에 역전파(back-propagation)에서와 같이 업데이트된 레이어의 손실 함수의 기울기를 나타낸다.

[0058] 역전파는 CNN에서 훈련 가능한 모든 매개변수(w.r.t)에 대한 손실의 도함수를 계산한다. 입력 X, 즉 $\frac{dl}{dX}$ 및 기타 매개변수 α , β 및 γ 에 대한 계층 기울기 손실 w.r.t는 $k=X_{mask}/X$ 로 업데이트된다.

[0059] 수학식 5로부터 본 발명은 수학식 7을 도출한다.

수학식 7

$$\frac{dl}{dX} = \alpha \cdot \frac{dl}{dz} + \beta \cdot k \cdot \frac{dl}{dz}$$

[0061]

[0063] 본 발명의 실험에서, 'k' 값이 사용될 때, 그라디언트 소실 문제를 일으키는 작은 그라디언트 값을 생성한다는 것을 발견했다. 따라서 k는 1로 선택되었다. 따라서 32개의 활성화와 64개의 미니 배치 크기를 가진 GAP 레이어에 대해 그라디언트 손실을 정의한다. 다음과 같이:

수학식 8

$$\frac{dl}{d\alpha} = \sum_{b=1}^{64} \sum_{n=1}^{32} X_n^b \cdot \frac{dl}{dz}$$

[0065]

수학식 9

$$\frac{dl}{d\beta} = \sum_{b=1}^{64} \sum_{n=1}^{32} Xmask_n^b \cdot \frac{dl}{dz}$$

[0067]

수학식 10

$$\frac{dl}{d\gamma} = \sum_{b=1}^{64} \sum_{n=1}^{32} \frac{dl}{dz}$$

[0069]

[0071] 미니배치에서의 훈련은 수렴 시간을 달성하는 데 상당한 영향을 미친다. 즉, 미니배치 크기가 클수록 Epoch당 반복 횟수를 줄여 훈련을 더 빠르게 하는 반면 훈련 정확도에도 영향을 미친다. 본 발명은 또한 미니 배치로 네트워크를 훈련하고 있다. 즉, 엔트로피 손실은 미니 배치 입력을 기반으로 계산되지만 사용된 정규화 계층이 GAP인 경우 정규화 프로세스는 배치 정규화가 아니다. 정규화에서는 일반적으로 입력이 0 평균 및 단위 분산으로 선형 변환되는 미백 프로세스가 선호된다. 즉, 입력은 내부 공변량 이동의 악영향을 제거하는 것으로 간주되는 고정 분포를 갖는다. 활성화 프로세스는 활성화 함수가 가중치 출력을 결정하는 필터링 프로세스와 같다. 유사하게 본 발명의 방법에서 가우스 필터는 정규화된 마스크를 생성하기 위해 학습된 매개변수와 함께 활성화 함수로 작동한다. 이 마스크는 원래 입력과 함께 추가 추출된 특징($\beta \cdot Xmask$)으로 작동하므로 원래 신호가 필터 평균 응답으로 약간 부스트된다. 따라서 나중에 원래 입력에 선형으로 추가할 때 마스크에 따라 값을 추가한다. 마스크된 값만 사용하면 전체 입력 이미지 속성을 잃게 된다. 따라서 마스크는 입력의 선형 속성을 잃지 않고 계산된 분산을 가져오기 위해 원래 입력에 추가된다.

[0072] 초기 학습률, 학습 중단률 등과 같은 소수의 하이퍼파라미터는 학습 시간 및 학습률에 영향을 미치지만 장기적으로 결과는 크게 다르지 않으며 최근에 달성된 수렴은 테스트 결과에만 약간 영향을 미친다. 완전 연결 계층(Fully connected layer, FCL)은 입력에서 출력까지 모든 매개변수가 연결된 단일 계층 피드포워드 네트워크 역할을 한다. 이러한 특성 때문에 FCL은 네트워크에서 과적합을 유발하는 것으로 비난받고 있으며, 그 사이에 dropout과 같은 potential regularization 기술이 사용된다[26],[30]. 따라서 매개변수의 수를 줄이고 궁극적으로 과적합을 위해 기본 아키텍처에서 FCL의 수를 4에서 3으로 줄였다[12].

[0074] B. Classification Performance and Discussion

[0075] 본 발명은 처음 2개의 인코더 부분에 사용된 서로 다른 정규화 레이어를 사용하여 기본 아키텍처를 사용하여 여러 분류 실험을 수행했다. 본 발명의 주요 목표는 분류 결과를 비교하는 것이었다. 처음 두 개의 BN 레이어는 GAP 레이어로 대체되었다. 이를 위해 벤치마크 2D 데이터 세트가 사용되었다. 사용된 데이터셋은 CIFAR-10[10], Caltech-102[27], 5마리 동물, 2D 분류 목적을 위한 OASIS[28]의 MRI 영상이다. 이 중 5마리 동물과 MRI 이미지는 비공개로 준비되어 공개 리포지토리에 제공되는 반면 나머지는 이미 공개적으로 제공된다. 5마리 동물 데이터셋은 호랑이, 사자, 개, 고양이, 여우 등 5가지 동물 클래스당 약 700개의 이미지로 구성된다. OASIS MRI는 참가자의 임상치매등급(CDR) 수준에 따라 분류된 4개 등급의 총 5220개의 영상으로 구성된다. 참가자에 대한 자세한 내용은 2D CNN [29]에 대한 이전 작업에 설명되어 있다. 3D CNN 성능을 테스트하기 위해 ADNI 데이터베이스(<http://adni.loni.usc.edu/>)에서 얻은 3D-MRI 볼륨을 사용했다. 부피가 큰 훈련 샘플과 작은 훈련 샘플의 결과를 비교하기 위해 3D CNN에 대해 2가지 유형의 데이터 세트를 준비했다. MRI_baseline은 988개의 MRI 샘플이 있는 더 큰 데이터 세트이고 MRI_small은 187개의 MRI 샘플이 있는 데이터 세트이다. 이 MRI는 AD(알츠하이머병), MCI(경도 인지 장애) 및 NC(정상 대조군)의 3가지 클래스 중 하나에 속한다. 이러한 의료 데이터베이스

에 대한 자세한 내용은 이전 작업[12]에서 찾을 수 있다.

[0076]

표 2는 실험에 대한 세부 조건과 얻은 결과를 나타낸다. 전반적으로, 정규화를 위해 제안된 계층을 사용하면 배치 정규화와 거의 유사한 결과를 제공하며 일부 경우에는 훈련 시간이 높을수록 결과가 약간 더 좋다. 실험 결과는 100% 배치 정규화를 사용하여 전체 테스트 정확도 93.58% 대 91.89%로 MRI_baseline과 같은 더 큰 데이터 세트에서 정규화 계층으로 이중 GAP 레이어로 분류하는 것이 더 좋다. 마찬가지로 CIFAR-10 테스트 정확도는 첫 번째 및 두 번째 인코더에서 GAP 정규화 레이어를 BN 레이어로 교체하여 75.11%에서 75.21%로 향상되었다. 정규화 및 활성화를 위해 제안된 방법의 사용을 지원하는 5마리 동물 데이터 세트(2D CNN 정확도: 62.92% 대 58.48%)와 같은 더 작은 데이터 세트에서 첫 번째 BN 레이어를 GAP로 교체하면 결과가 더 좋다.

[0077]

표 2는 표 1과 같이 동일한 기본 아키텍처에서 다양한 정규화 기술을 사용한 자세한 실험 결과를 나타낸 것이다. 굵은 숫자는 최고의 성능 결과를 나타낸다. 훈련 환경은 자연 이미지의 모든 데이터 세트의 경우 및 의료 MRI의 경우 유사했다. AlexNet 모델은 MATLAB Deep Learning Tool을 사용하여 얻었다. 동일한 데이터 세트의 경우 교육, 검증 및 테스트 자료도 동일하므로 결과는 어떠한 경우에도 편향될 수 없다. 정확도는 예측 중에 올바르게 분류된 샘플의 %를 나타내는 반면 평균 테스트 재현율 및 정밀도는 클래스별 재현율 및 정밀도의 평균을 취하여 계산된다. 95% CI 오류는 95% 신뢰 점수의 오류를 나타내며 95% 이상의 점수를 가진 오류는 최소 오류 값에 대해서만 계산되고 95% 미만의 점수가 있는 오류는 최대 오류 값에 대해서만 계산된다.

표 2

Natural Dataset	ARCHITECTURE	EPOCH	# OF TRAINING IMAGE		# OF VALIDATION IMAGE	# OF TESTING IMAGE		TRAINING TIME (MIN)		VALIDATION ACCURACY		TESTING ACCURACY		AVERAGE TEST PRECISION		95% CI ERROR (MIN)		95% CI ERROR (MAX)	
			NG	IMAGE		NG	IMAGE	NG	TIME	ACCURACY	ON RACY	NG	ACCURACY	PRECISION	RECALL	ERR	ERR	CI ERR	CI ERR
Caltech-102	g1b2b3b4		133	100	69.6	66.59	0.5714	0.5078	31.24	35.57									
	g1g2b3b4		345	100	68.27	66.54	0.5531	0.5108	31.3	35.62									
	b1b2b3b4	100	91	100	69.1	66.54	0.5178	0.5178	31.33	35.62									
5-animals	Weightless AlexNet (WA)		84	100	60.36	59.69	0.4317	0.426	38.16	42.65									
	g1b2b3b4		42	100	60.19	62.92	0.6424	0.6329	33.32	40.85									
	g1g2b3b4		120	100	59.1	58.16	0.6037	0.5801	37.99	45.69									
CIFAR-10	b1b2b3b4	100	30	100	57.1	58.48	0.5941	0.5899	37.68	45.37									
	WA		24	90	48.09	42.47	0.4151	0.421	53.67	61.38									
	g1b2b3b4		851	100	74.55	74.68	0.7465	0.7461	24.54	26.24									
OASIS MRI (2D MRI)	g1g2b3b4	100	2300	100	75.3	75.21	0.7502	0.7521	23.94	25.64									
	b1b2b3b4		390	100	74.68	75.11	0.7565	0.7513	24.02	25.72									
	WA		360	100	75.14	74.76	0.7492	0.7476	24.39	26.09									
ADNI MRI small (3D CNN)	g1b2b3b4	50	180	100	99.67	99.62	0.9966	0.9967	0.08	0.69									
	g1g2b3b4		41	100	99.71	99.81	0.9976	0.9987	0.01	0.51									
	WA		13	100	98.66	99.75	0.9951	0.995	0.2	0.95									
ADNI MRI baseline (3D CNN)	g1b2b3b4	100	196	100	71.05	80.1	0.8247	0.7895	9.43	30.57									
	g1g2b3b4		551	100	73.36	81.82	0.8197	0.8092	7.99	27.38									
	b1b2b3b4		81	100	73.68	83.64	0.8349	0.8288	6.59	26.14									
MRI baseline (3D CNN)	g1b2b3b4	100	670	100	94.92	92.23	0.9438	0.906	4.72	10.82									
	g1g2b3b4		1780	100	94.42	93.58	0.9346	0.9256	3.63	9.21									
	b1b2b3b4		244	100	94.92	91.89	0.9134	0.9079	5.11	11.22									

[0079]

[0081] C. Feature Visualization and Analysis

[0083] D. Correlation and Generalization

[0084] 상관 관계는 두 신호 간의 유사성을 측정하는 것이다. 두 확률 변수의 상관 계수는 N 스칼라 관측값을 Pearson 상관 계수 r로 사용하여 입력 특성 행렬(X)과 출력(Z) 간의 선형 종속성을 측정한다.

수학식 11

$$r(X, Y) = \frac{1}{N-1} \sum_{i=1}^N \left(\frac{X_i - \mu_X}{\sigma_X} \right) \left(\frac{Y_i - \mu_Y}{\sigma_Y} \right) \\ = \frac{\text{cov}(X, Y)}{\sigma_X \sigma_Y}$$

[0086]

[0088] 상관 계수는 두 벡터 행렬 간의 공분산 'cov'도 측정한다. BN이 해당 레이어의 내부 공분산 이동을 줄이기 위해 명시되어 있으므로 Pearson 상관 계수(r)를 사용하여 이 레이어에서 출력과 입력 간의 공분산을 측정했다. 여기서 본 발명의 경우 X는 선행 컨볼루션 레이어에서 생성된 3D 특징 행렬이고 Z는 X와 같은 차원을 가진 정규화 레이어의 출력이다. 그리고 'r'의 값은 단일 이미지에 대한 모든 필터에 대한 단일 값이므로 활성화 레이어의 평균 특성을 연관시킨다. (필터 방식 응답의 경우 그림 7.1(b)에서 7.4(b)까지의 평균 응답 플롯을 플로팅했다.) 완벽한 상관 관계는 입력과 출력 간에 분산 없이 동일한 결과를 가져오므로 레이어가 무용지물이 되지만 매우 낮은 상관도 위험하다. 레이어 입력과 출력 사이의 변동성과 이동이 매우 높아 레이어가 사라지는 그라디언트처럼 사고를 의심하게 만들기 때문이다. 따라서 높은 상관관계가 좋은 것인지 아닌지는 여전히 불분명하며, 본 발명의 경우에는 BN 결과보다 높은 상관값을 기대했지만 실제로 밝혀졌다.

[0089] 과적합은 훈련된 네트워크 내에서 테스트 성능과 훈련 성능의 차이를 가져오는 ML의 주요 역경 중 하나이다. 이는 훈련 세트의 오차가 매우 적지만 네트워크를 통해 보이지 않는 유사한 데이터가 예측될 때 오차가 클 때 확인할 수 있다. 이는 네트워크가 '잘 암기(memorized well)'되었지만 '잘 학습되지 않음(not learned well)'을 의미한다. 과적합에 대한 네트워크의 운명을 결정하는 것은 주로 매개변수의 수이다. 네트워크의 매개변수 수가 훈련 세트의 총 포인트 수보다 훨씬 작으면 과적합 가능성이 거의 또는 전혀 없다는 점을 염두에 두어야 한다. 이는 단순히 훈련 네트워크의 매개변수를 증가시키면 과적합 일반화 오류를 확인하기 위해 표준 테스트 오류 (STE) = 1 - 정확도에 대해 95% 이상의 신뢰 점수에 대해 'N' 테스트 샘플의 예측 오류 범위를 계산했다. 이 오차 한계 테스트는 Wald 테스트라고도 하며 표 2와 같이 95% 신뢰 구간 범위에서 최소 및 최대 오차를 나타내며 정확도 그래프와 함께 도 6에 표시된다. 또한 테스트 이미지에 대한 T-SNE 투영을 플롯하고 도 9와 같이 오류 분포를 찾았다.

[0091] <CONCLUSION>

[0092] 결론을 내리기 위해 2D 및 3D 아키텍처 CNN에서 정규화 계층으로 GAP 계층에 대한 제안된 아이디어를 실험했다. 본 발명은 GAP 레이어의 사용이 더 큰 3D 데이터 세트와 더 가벼운 2D 데이터 세트의 경우 유사하거나 약간 더 나은 결과를 생성한다는 것을 보여준다. 훈련 및 검증 그래프를 통한 과적합 현상, 평균 응답 플롯 및 상관 플롯을 통한 정규화 계층 공분산, TSNE 및 히스토그램 플롯을 통한 특징 표현에 대해 연구했다. 요약하면 다음과 같이 결과를 나열했다.

- [0094] 1. b1b2b3b4 아키텍처를 사용하면 훈련 정확도가 더 빨리 높아져 중간 훈련 동안 검증 정확도가 여전히 너무 낮아 훈련과 검증 정확도 사이의 격차가 더 커 과적합 상태를 나타낸다. 따라서 GAP 레이어를 사용하여 훈련 중 가중치의 더 빠른 수렴을 지연했다(도 2 참조).
- [0096] 2. 초기 레이어에서 컨볼루션 필터의 가중치는 BN 동안 컨볼루션에서 정규화 프로세스로 갑자기 변경되는 것으로 보인다(도 4 참조). 이는 입력 이미지의 특징 속성이 정규화 후 입력과 매우 상관관계가 없음을 시사할 수 있다. 결과적으로 BN의 입력과 출력 사이의 상관 계수는 상당히 낮다. 그러나 GAP를 사용하면 필터 가중치가 약간만 변경되고 입출력 상관 값이 높아져(도 7 및 8 참조) 이미지 속성의 왜곡이 적음을 나타낸다.
- [0098] 3. BN의 스케일링에 사용되는 미니배치 평균값으로 인해 정규화 레이어의 출력은 미니배치 평균으로 스케일링된다. 즉, 이미지의 미니배치 속성이 혼합되어 특징 값이 더 많이 압축된다(도 7의 히스토그램 플롯 참조). 이것은 상기 2.에서 논의된 바와 같이 BN 출력의 더 높은 변동성을 가져올 수 있다. 반면에 GAP 정규화에서 스케일링 평균 계수는 동일한 이미지의 활성화된 채널에서 계산된다. 즉, 미니 배치 = 1과 동일하므로 이미지 속성 동일한 이미지의 이미지는 특징 속성을 손상시키지 않고 혼합될 뿐이다.
- [0100] 4. 스케일링 평균과 분산은 상기 3.과 같이 각 입력 이미지에 대해 경험적으로 계산되므로 테스트 단계에서 BN과 같이 훈련된 평균과 분산 값을 전달할 필요가 없다. 또한 배치 통계 추정이 정확하지 않기 때문에 배치 크기가 작을수록 BN의 오류가 더 높습니다.
- [0102] 5. 의사결정을 위한 활성화 영역은 BN 기반 b1b2b3b4 및 AlexNet보다 g1g2b3b4 또는 g1b2b3b4를 사용하는 대부분의 경우 더 넓고 정확하다(도 10 참조).
- [0104] 6. 대부분의 실험에서 결과가 약간 더 우수하고(표 2) T-SNE 투영을 통해 시각화할 수도 있다(도 9 참조).
- [0106] <부록: Appendix>
- [0107] MATLAB 구현 코드는 C1, C2, C3으로 표시된다. 성능의 혼동 매트릭스(표 2 참조)도 제공된다. 준비된 데이터 세트는 다음에서 다운로드할 수 있다.
- [0108] https://drive.google.com/drive/folders/1G1fsK2VxaHkvtJJf_vpiB3rMpiqCcdkB2?usp=sharing

[0110] C1: 배치 정규화 계층의 대안으로 컨볼루션 계층 다음에 사용되는 GAP 계층에 대한 MATLAB 코드 구현.

```

encoder_1 = [
    convolution2dLayer([3 3], 32, 'Padding',
        'same', 'WeightsInitializer', 'glorot');
    test_gauss(32,'g1',3); % batchNormalization
    Layer for BN
    leakyReluLayer('Name', 'relu_s1');
    maxPooling2dLayer(poolSize,'Stride',2);
]
The defination for test_gauss is as below:
classdef test_gauss < nnet.layer.Layer
    properties(Learnable)
        Alpha
        Beta
        Gamma
    end
    properties
        hsize
    end
    methods
        function layer =
            test_gaus(numChannels,name,hsize)
            layer.Name = name;
            layer.hsize = hsize;
            layer.Description = "Scale factor" +
                numChannels + "channels";
            layer.Alpha = rand([1 1 numChannels]); % 3D
            rand([1 1 1 numChannels]);
            layer.Beta = rand([1 1 numChannels]);
            layer.Gamma = rand([1 1 numChannels]);
        end
        function Z = predict(layer, X)
            X_mean = double(mean(X,3)); %
            X(w1,w2,fiternumber,batchsize)
            X_std = std(X,0,3); % For 3D 4th dimension is
            used for mean and std
            X_F = (X-X_mean)/X_std;
            h_size = layer.hsize;
            h = fspecial3('gaussian',h_size);
            h_f = [h]; % h_f = convn(h,h,'same')for 2nd gap
            layer
            X_test = imfilter(X_F,h_f,'replicate');
            Z = layer.Alpha.*X + layer.Beta.*(X-X_test) +
            layer.Gamma;

```

[0111]

```

end
function [dLdX, dLdAlpha,dLdBeta,dLdGamma] =
backward(layer, X, ~, dLdZ, ~)
    h_size = layer.hsize;
    h = fspecial3('gaussian',h_size);
    X_mean = double(mean(X,3)); %X(w1,w2,
fiternumber, batchsize)
    X_std = std(X,0,3);
    X_F = (X-X_mean)/X_std;
    h_f = [h];
    X_test = imfilter(X_F,h_f,'replicate');
    dLdX = (layer.Alpha.*dLdZ + layer.Beta.*dLdZ);
    dLdAlpha = X.* dLdZ;
    dLdAlpha = sum(dLdAlpha,[1 2]); % for 3D

```

[0113]

```

    dLdBeta = sum(dLdBeta,[1 2 3]);
    dLdAlpha = sum(dLdAlpha,4); % for 3D
    dLdAlpha = sum(dLdAlpha,5);
    dLdBeta = (X-X_test).* dLdZ;
    dLdBeta = sum(dLdBeta,[1 2]);
    dLdBeta = sum(dLdBeta,4);
    dLdGamma = dLdZ;
    dLdGamma = sum(dLdGamma,[1 2]);
    dLdGamma = sum(dLdGamma,4);
end
end
end

```

[0114]

[0116] C2: 도 7 및 8에서와 같이 기능 시각화 및 평균 응답을 위한 코드 스니펫.

[0118] 도 7.1(b) ~ 7.4(b)와 같이 기능을 시각화하고 제안된 레이어의 평균 응답을 얻으려면 여기서 X와 Z는 각각 예측 기능에 대한 입력과 출력을 나타낸다.

```

...
Load net,X,Z % Need to save net, X and Z layer as.mat
file previously.
feature = gather(X);
x(:) = gather(mean(X,[1 2]));
x = x';
z(:) = gather(mean(Z,[1 2]));
z = z';
plot(x)
hold on
plot(z)
hold off
Dsize = size(X,4); %for 3D CNN size(X,5)
for i = 1:Dsize
A = gather(X(:,:,:,i)); %for 3D CNN 'i' in 5th
dimension i.e minibatch representing test samples
B = gather(Z(:,:,:,i));
R (:,i) = corrcoef(A,B);
plot(R)
end
...

```

[0120] ...

[0122] C3: 도 5와 같이 Deep dream을 사용하여 특징 맵을 생성하기 위한 코드 스니펫 구현

```

...
load net; % Load Pretrained Network
layer = 22; % Last FC layer of the network
trained_net.Layers(end).ClassNames(channels)
% Generate detailed images that strongly activate these
classes.
I = deepDreamImage(trained_net,layer,channels,...
'Verbose', false, ...
'NumIterations',50);
figure
montage(I)
name = net.Layers(layer).Name;
title(['Layer', name, 'Features'])

```

[0123] ...

[0125] 도 11을 참조할 수 있다.

[0127] <References>

- [0128] 1. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, et al., "Backpropagation applied to handwritten zip code recognition", Neural Comput., vol. 1, no. 4, pp. 541-551, Dec. 1989.
- [0129] 2. Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, et al., "Handwritten digit recognition with a back-propagation network", Proc. Adv. Neural Inf. Process. Syst., pp. 396-404, 1990.
- [0130] 3. A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks", Proc. Adv. Neural Inf. Process. Syst., vol. 25, pp. 1097-1105, 2012.
- [0131] 4. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, et al., "Going deeper with

convolutions", Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 1-9, Jun. 2015.

- [0132] 5. K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 770-778, Jun. 2016.
- [0133] 6. R. Girshick, "Fast R-CNN", Proc. IEEE Int. Conf. Comput. Vis. (ICCV), pp. 1440-1448, Dec. 2015.
- [0134] 7. S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks", IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137-1149, Jun. 2017.
- [0135] 8. V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation", IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 12, pp. 2481-2495, Dec. 2017.
- [0136] 9. A. Kendall, V. Badrinarayanan and R. Cipolla, "Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding", Proc. Brit. Mach. Vis. Conf. (BMVC), pp. 57.1-57.12, 2017.
- [0137] 10. A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images", pp. 1-60, 2009.
- [0138] 11. A. Mordvintsev, C. Olah and M. Tyka, Inceptionism: Going Deeper Into Neural Networks, Feb. 2015, [online] Available: <https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html>.
- [0139] 12. B. Khagi and G.-R. Kwon, "3D CNN design for the classification of Alzheimer's disease using brain MRI and PET", IEEE Access, vol. 8, pp. 217830-217847, 2020.
- [0140] 13. C. Gulcehre and Y. Bengio, "Knowledge matters: Importance of prior information for optimization", J. Mach. Learn. Res., vol. 17, pp. 1-32, Jan. 2016.
- [0141] 14. S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", Proc. 32nd Int. Conf. Mach. Learn. (ICML), vol. 1, pp. 448-456, Jul. 2015.
- [0142] 15. S. Wiesler and H. Ney, "A convergence analysis of log-linear training", Proc. Adv. Neural Inf. Process. Syst., vol. 24, pp. 657-665, 2011.
- [0143] 16. K. He, X. Zhang, S. Ren and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification", Proc. IEEE Int. Conf. Comput. Vis. (ICCV), pp. 1026-1034, Dec. 2015.
- [0144] 17. J. M. Prewitt, "Object enhancement and extraction", Picture Process. Psychopictorics, vol. 10, no. 1, pp. 15-19, 1970.
- [0145] 18. R. M. Lewitt and S. Matej, "Overview of methods for image reconstruction from projections in emission computed tomography", Proc. IEEE, vol. 91, no. 10, pp. 1588-1611, Oct. 2003.
- [0146] 19. C. Rafael, Gonzalez Richard E. Woods Digital Image Processing, Upper Saddle River, NJ, USA:Prentice-Hall, 2002.
- [0147] 20. I. Sobel, "History and definition of the Sobel operator", Proc. Retrieved World Wide Web, vol. 1505, 2014.
- [0148] 21. C. Zhang, S. Bengio, M. Hardt, B. Recht and O. Vinyals, "Understanding deep learning (still) requires rethinking generalization", Commun. ACM, vol. 64, no. 3, pp. 107-115, Mar. 2021.
- [0149] 22. H. Robbins and S. Monro, "A stochastic approximation method", Ann. Math. Statist., vol. 22, pp. 400-407, 1951.
- [0150] 23. T. Gaussian, "The Gaussian kernel", Front-End Vision and Multi-Scale Image Analysis, pp. 37-51, 2003.

- [0151] 24. T. Lindeberg, "Scale-space for discrete signals", IEEE Trans. Pattern Anal. Mach. Intell., vol. 12, no. 3, pp. 234-254, Mar. 1990.
- [0152] 25. K. P. Murphy, Machine Learning: A Probabilistic Perspective, Cambridge, MA, USA:MIT Press, 2012.
- [0153] 26. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting", J. Mach. Learn. Res., vol. 15, no. 1, pp. 1929-1958, 2014.
- [0154] 27. L. Fei-Fei, R. Fergus and P. Perona, "Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories", Proc. Conf. Comput. Vis. Pattern Recognit. Workshop, pp. 178, Jun. 2004.
- [0155] 28. D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris and R. L. Buckner, "Open access series of imaging studies (OASIS): Cross-sectional MRI data in young middle aged nondemented and demented older adults", J. Cognit. Neurosci., vol. 19, no. 9, pp. 1498-1507, Sep. 2007.
- [0156] 29. B. Khagi, G. Kwon and R. Lama, "Comparative analysis of Alzheimer' s disease classification by CDR level using CNN feature selection and machine-learning techniques", Int. J. Imag. Syst. Technol., vol. 29, no. 3, pp. 297-310, Sep. 2019.
- [0157] 30. J. T. Springenberg, A. Dosovitskiy, T. Brox and M. Riedmiller, "Striving for simplicity: The all convolutional net", Proc. 3rd Int. Conf. Learn. Represent. (ICLR), pp. 1-14, 2015.
- [0158] 31. R. Girshick, J. Donahue, T. Darrell and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation", IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 1, pp. 142-158, Jan. 2016.
- [0159] 32. J. Yosinski, J. Clune, A. Nguyen, T. Fuchs and H. Lipson, "Understanding neural networks through deep visualization", arXiv:1506.06579, 2015, [online] Available: <http://arxiv.org/abs/1506.06579>.
- [0160] 33. M. T. Ribeiro, S. Singh and C. Guestrin, " 'Why should i trust you?' explaining the predictions of any classifier", Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, pp. 1135-1144, 2016.
- [0161] 34. M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks", Proc. Eur. Conf. Comput. Vis., vol. 8689, pp. 818-833, 2014.
- [0162] 35. R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Grad-CAM: Visual explanations from deep networks Via gradient-based localization", Proc. IEEE Int. Conf. Comput. Vis. (ICCV), pp. 618-626, Oct. 2017.
- [0163] 36. A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures", IEEE Access, vol. 7, pp. 53040-53065, 2019.
- [0164] 37. L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, et al., "Review of deep learning: Concepts CNN architectures challenges applications future directions", J. Big Data, vol. 8, no. 1, pp. 1-74, Dec. 2021.
- [0165] 38. S. Singh and S. Krishnan, "Filter response normalization layer: Eliminating batch dependence in the training of deep neural networks", Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 11234-11243, Jun. 2020.
- [0166] 39.Y. Wu and K. He, "Group normalization", Int. J. Comput. Vis., vol. 128, no. 3, pp. 742-755, Mar. 2020.
- [0167] 40.J. Lei Ba, J. Ryan Kiros and G. E. Hinton, "Layer normalization", arXiv:1607.06450, 2016, [online] Available: <http://arxiv.org/abs/1607.06450>.
- [0168] 41.Z. Dai and R. Heckel, "Channel normalization in convolutional neural networks avoids vanishing

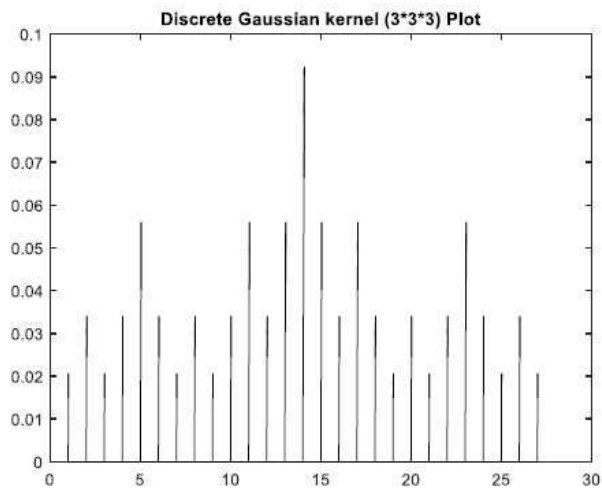
gradients", arXiv:1907.09539, 2019, [online] Available: <https://arxiv.org/abs/1907.09539>.

[0169] 42.D. Ulyanov, A. Vedaldi and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization", arXiv:1607.08022, 2016, [online] Available: <http://arxiv.org/abs/1607.08022>.

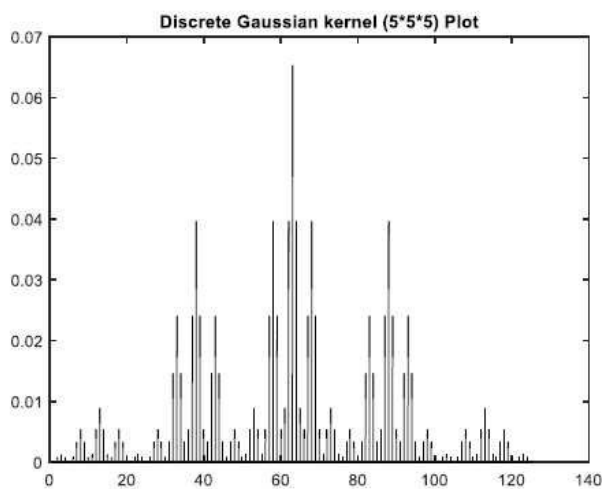
[0171] 이상의 설명은 본 발명을 예시적으로 설명한 것에 불과한 것으로, 본 발명에 속하는 기술분야에서 통상의 지식을 가지는 자라면 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 다양한 변형이 가능할 것이다. 따라서, 본 명세서에 개시된 실시예들은 본 발명을 한정하기 위한 것이 아니라 설명하기 위한 것이고, 이러한 실시예에 의하여 본 발명의 사상과 범위가 한정되는 것은 아니다. 본 발명의 보호범위는 아래의 청구범위에 의하여 해석되어야 하며, 그와 동등한 범위 내에 모든 기술은 본 발명의 권리범위에 포함하는 것으로 해석되어야 할 것이다.

도면

도면1

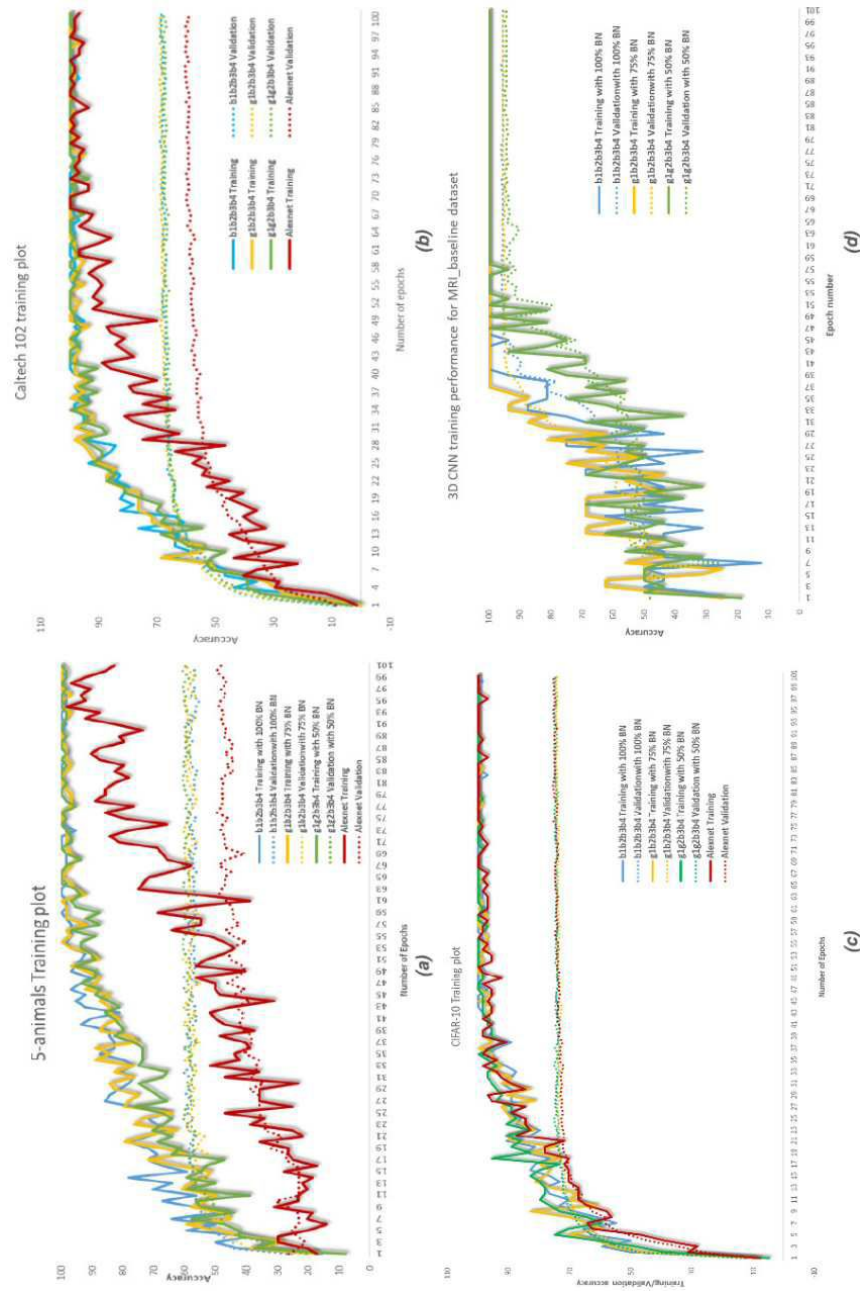


(a)

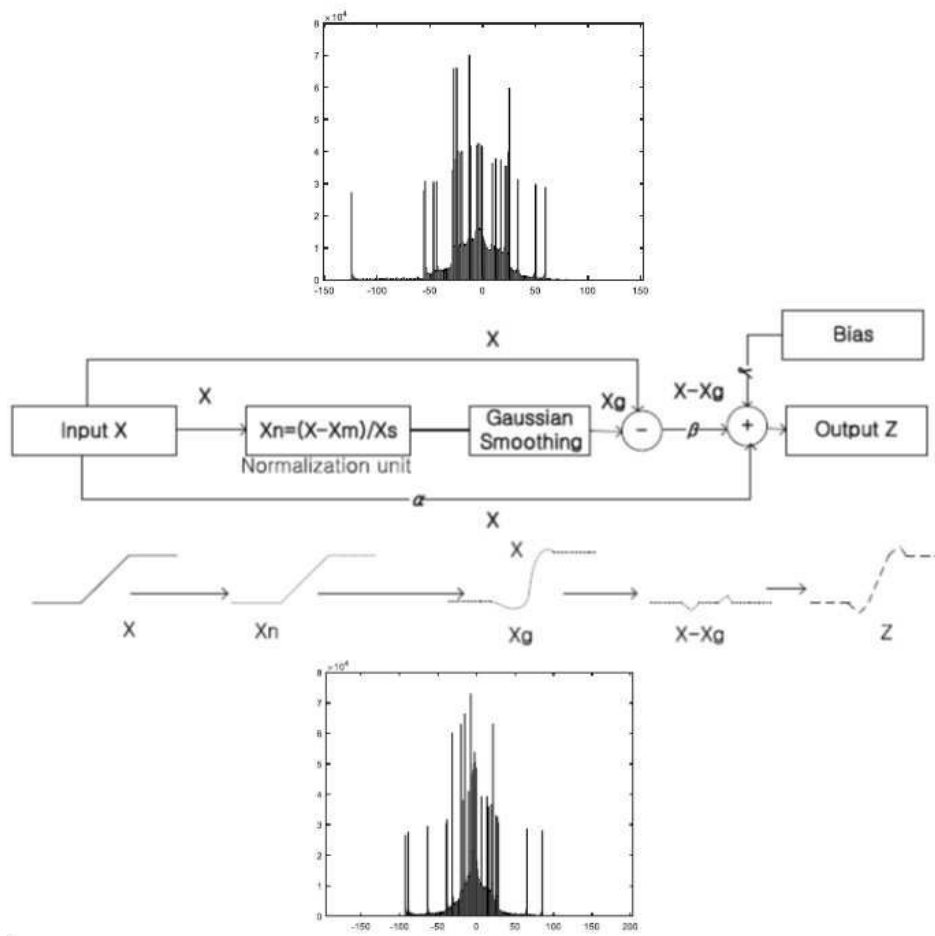


(b)

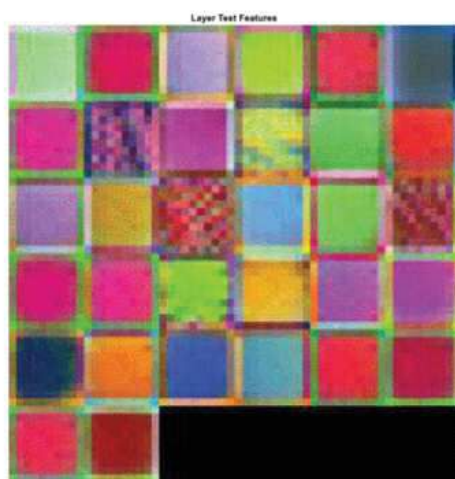
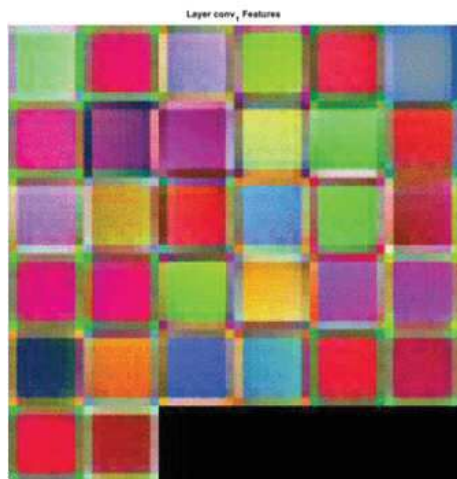
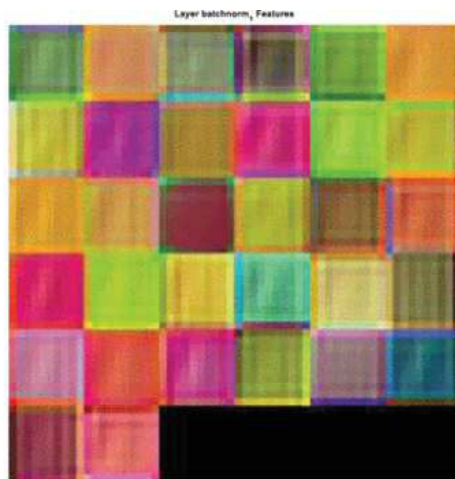
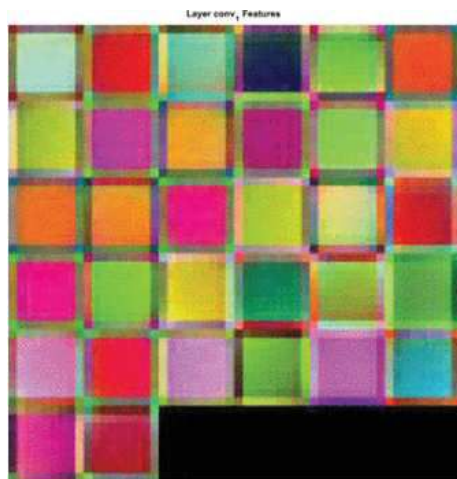
도면2



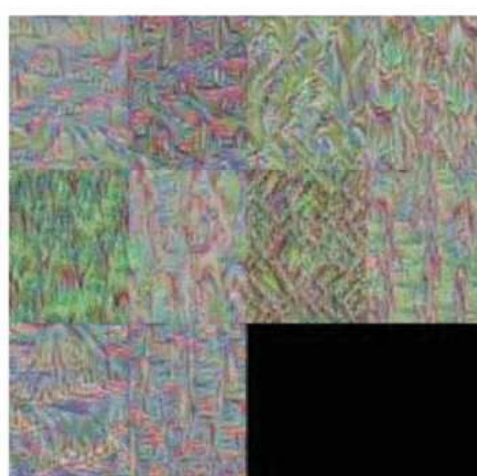
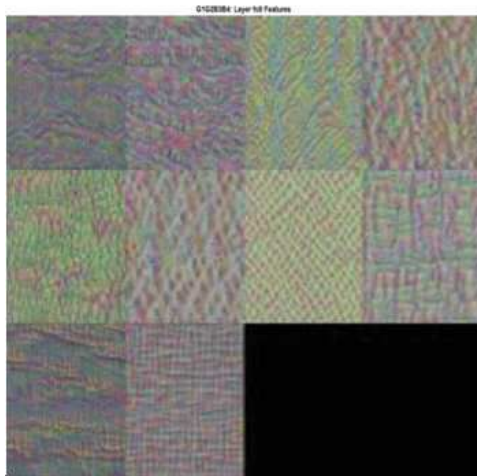
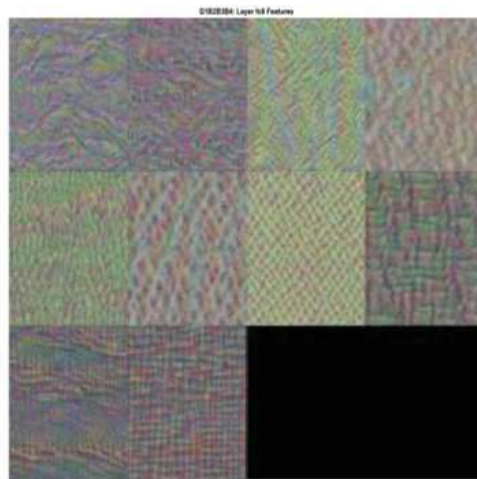
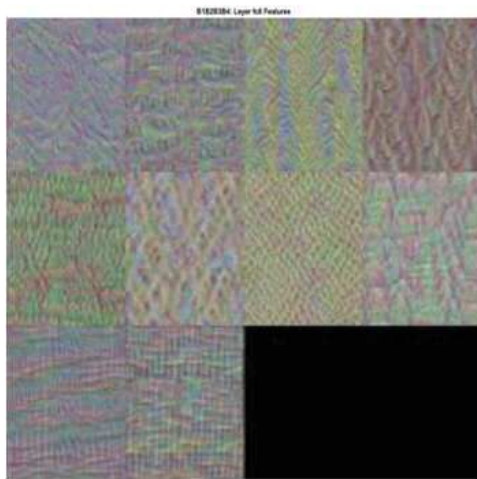
도면3



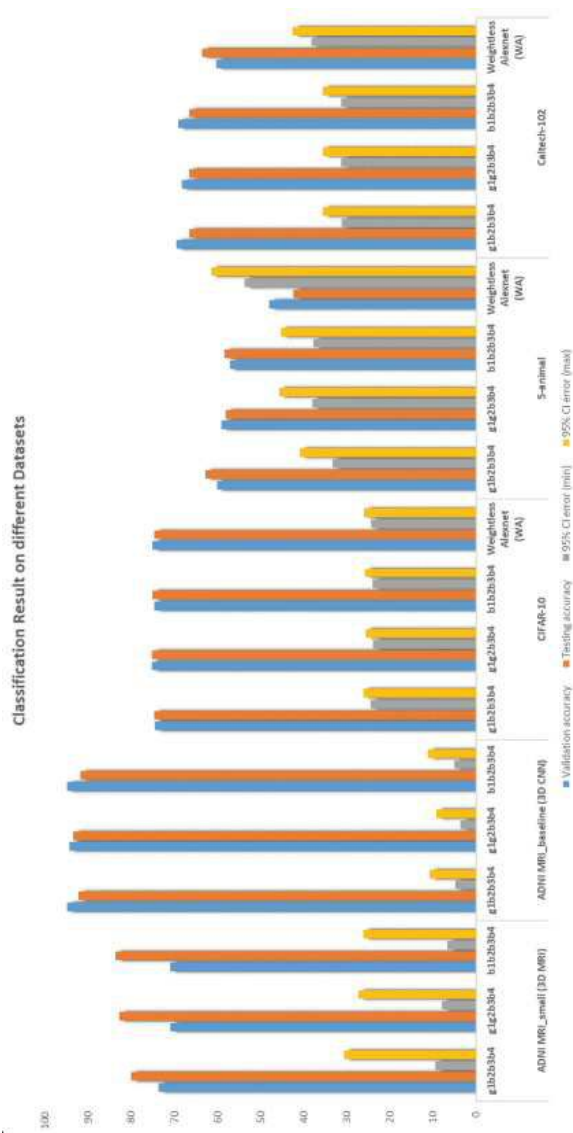
도면4



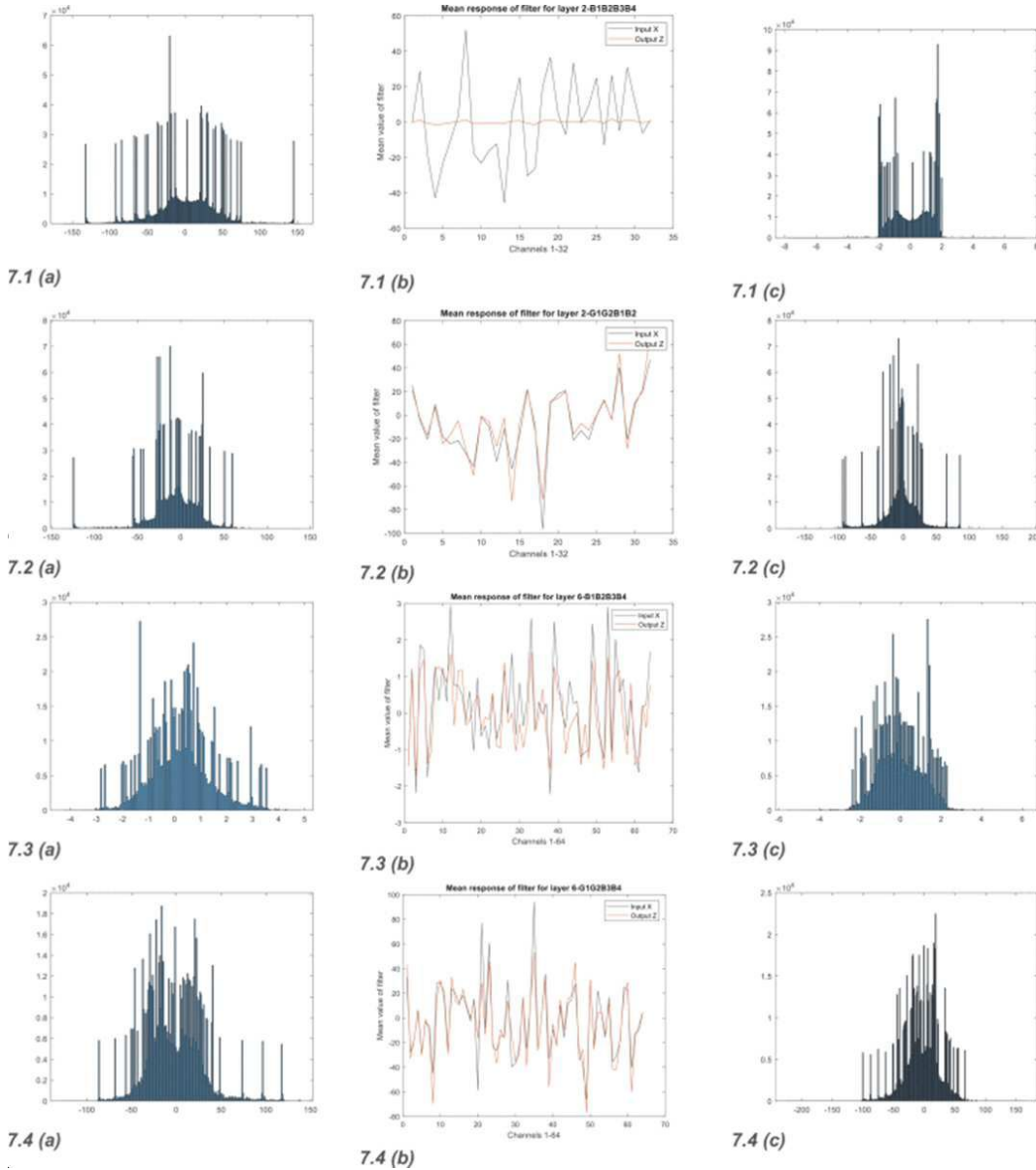
도면5



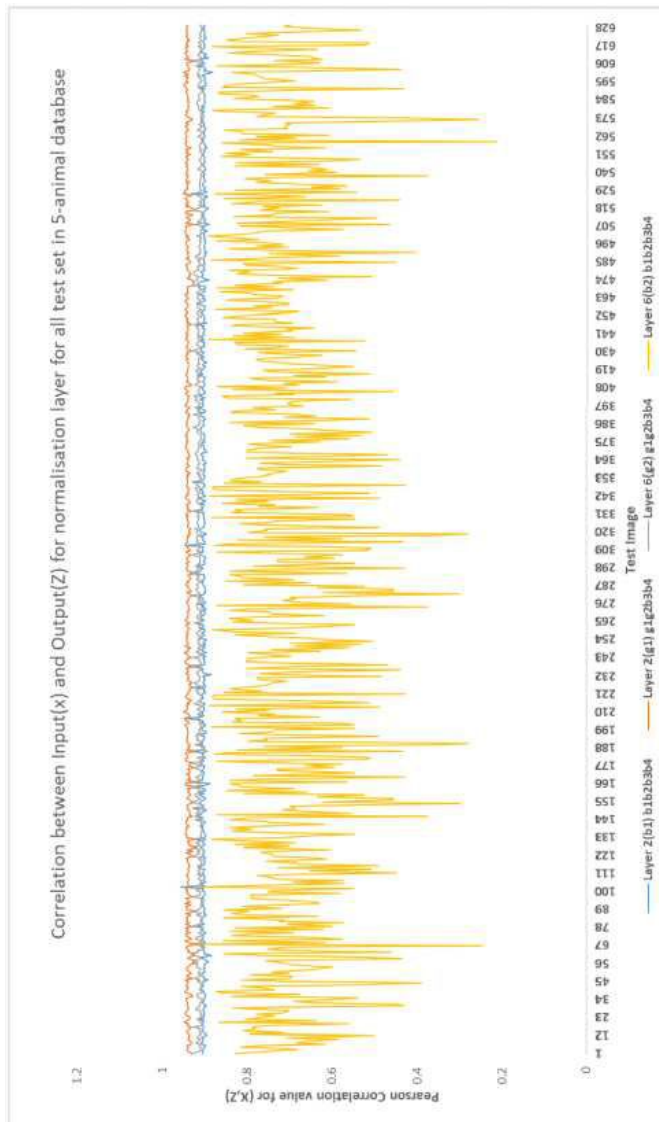
도면6



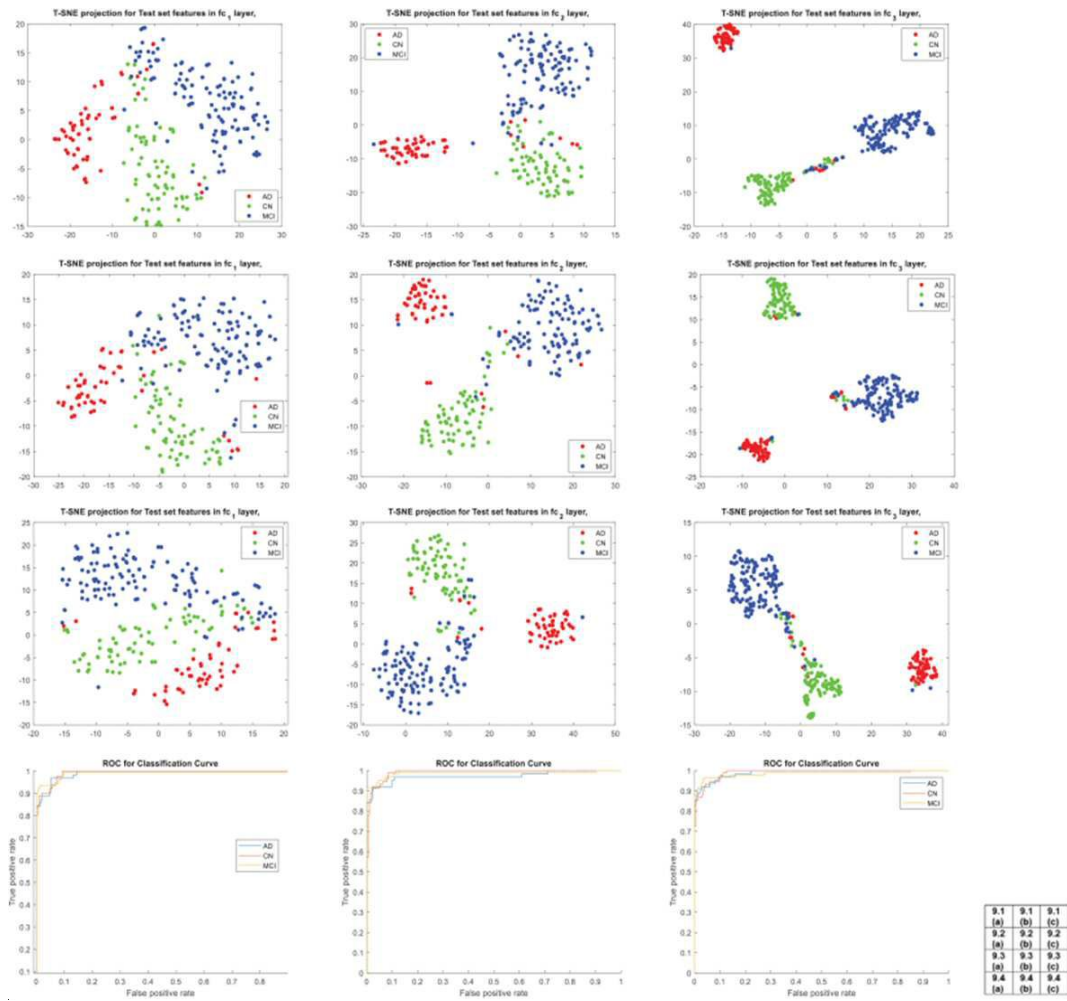
도면7



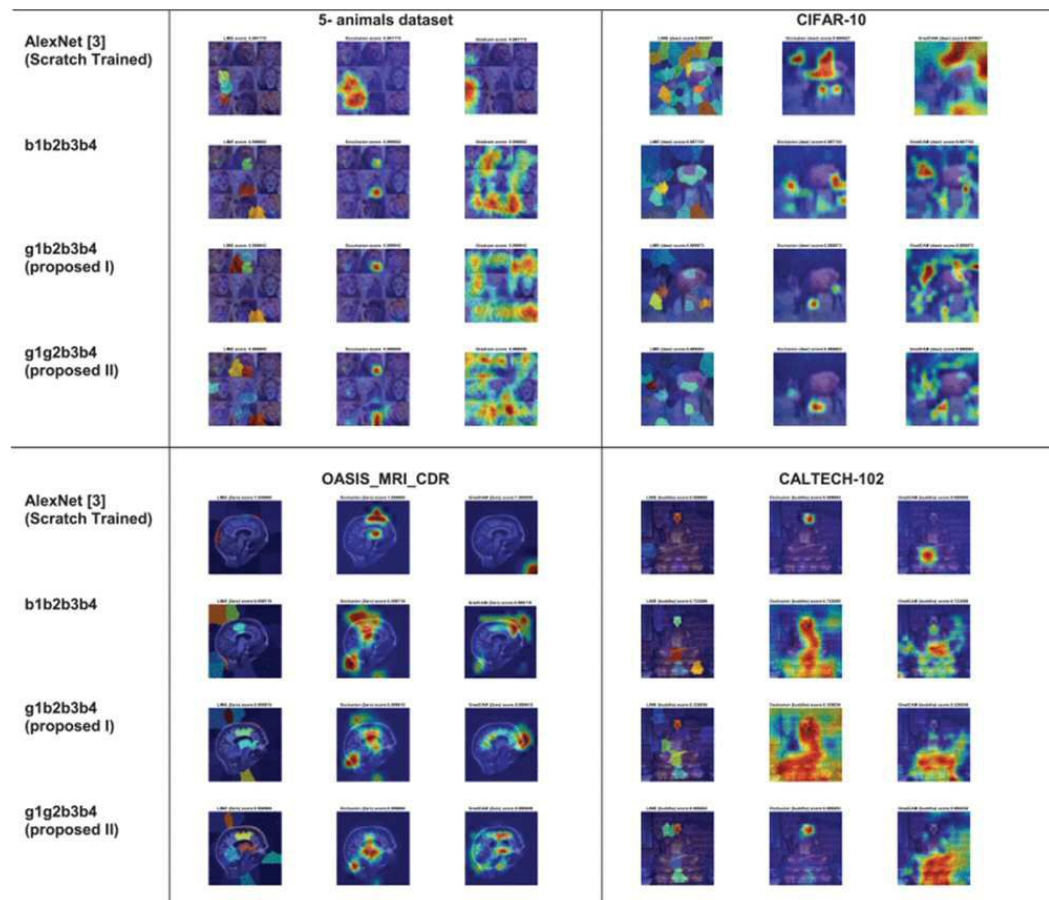
도면8



도면9



도면10



도면11

