

FFT 적용을 통한 Convolution 연산속도 향상에 관한 연구

임수창[†], 김종찬^{††}

A Study on the Optimization of Convolution Operation Speed through FFT Algorithm

Su-Chang Lim[†], Jong-Chan Kim^{††}

ABSTRACT

Convolution neural networks (CNNs) show notable performance in image processing and are used as representative core models. CNNs extract and learn features from large amounts of train dataset. In general, it has a structure in which a convolution layer and a fully connected layer are stacked. The core of CNN is the convolution layer. The size of the kernel used for feature extraction and the number that affect the depth of the feature map determine the amount of weight parameters of the CNN that can be learned. These parameters are the main causes of increasing the computational complexity and memory usage of the entire neural network. The most computationally expensive components in CNNs are fully connected and spatial convolution computations. In this paper, we propose a Fourier Convolution Neural Network that performs the operation of the convolution layer in the Fourier domain. We work on modifying and improving the amount of computation by applying the fast fourier transform method. Using the MNIST dataset, the performance was similar to that of the general CNN in terms of accuracy. In terms of operation speed, 7.2% faster operation speed was achieved. An average of 19% faster speed was achieved in experiments using 1024x1024 images and various sizes of kernels.

Key words: Computer Vision, Convolution Neural Networks, Deep Learning, Fast Fourier Transform, Convolution

1. 서 론

딥러닝(Deep Learning)은 컴퓨터 성능향상과 여러 이론과 방법론이 등장하면서 비약적인 발전을 이루었다. 객체 인식, 분할, 추적 등 컴퓨터 비전 분야에 적용하는 연구가 진행되었다[1-3]. 특히, 컨볼루션 신경망(Convolution Neural Networks, CNN)은 이미지 처리에 우수한 성능을 보여주어 대표적인 핵심

모델로 사용된다. CNN은 1980년대에 Y. LeCun에 의해 개발되었다[4]. 수기로 작성된 우편 번호, 핀 코드 등을 인식하는 우편 작업에 사용되었다. 모든 딥러닝 모델에 있어 가장 중요한 것은 네트워크 훈련에 많은 양의 데이터와 많은 컴퓨팅 리소스가 필요하다는 것이다. 이러한 제약사항은 CNN의 주요 단점이었다. 당시 제한적인 데이터만 얻을 수 있었던 상황에서 CNN은 우편 부문에 제한적으로 사용했다.

* Corresponding Author : Jong-Chan Kim, Address: (57922) Jungang-ro 255, Suncheon, Jeonnam, Korea, TEL : +82-61-750-3628, FAX : +82-61-750-3620, E-mail : seaghost@sunchon.ac.kr

Receipt date : Oct. 28, 2021, Revision date : Nov. 6, 2021
Approval date : Nov. 9, 2021

[†] Dept. of Computer Engineering, Suncheon National University (suchanglim@sunchon.ac.kr)

^{††} Dept. of Computer Engineering, Suncheon National University

* This work was supported by a Research promotion program of SCNU

2012년 Alex Krizhevsky은 ILSVRC(ImageNet Large Scale Visual Recognition Challenge)에서 CNN기반 AlexNet을 발표하였다[5,6]. 이 네트워크는 기존 고전적인 기계학습 알고리즘보다 우수한 분류 성능을 달성하였다. 수백만 장의 이미지로 구성된 ImageNet 데이터셋과 과거에 비해 비약적으로 발전한 컴퓨팅 성능으로 인하여 CNN이 다시 주목받기 시작하였다.

CNN은 대량의 데이터에서 특징을 추출하고 학습한다. 네트워크는 일반적으로 N개층 이상의 컨볼루션 레이어와 풀리 커넥티드 레이어가 적층된 구조로 이루어져 있다. CNN의 성능은 다수의 레이어가 깊고(Deeper), 넓게(Wide) 적층된 형태의 구조로 이루어질 때 향상된다고 알려져 있다[7,8]. 네트워크를 학습하기 위한 데이터의 개수도 거대해지고 있으며, 더 좋은 결과를 얻기 위해 네트워크를 깊게 만들기 위한 연구들이 진행되고 있다. 이들 연구는 다양한 사이즈로 커널을 제작한 컨볼루션 레이어를 적층시켜 일련의 모듈을 만들고, 이 모듈을 수 백층 이상 결합하여 네트워크로 만든다. CNN의 핵심은 컨볼루션 레이어다. 특징 추출에 사용되는 커널의 크기, 특징맵(Feature Map) 깊이를 결정하는 개수가 학습 가능한 CNN의 가중치 파라미터의 양을 결정한다. 이러한 파라미터는 신경망 전체의 계산 복잡도와 메모리 사용량을 증가 시키는 주요 원인이다. CNN에서 완전연결(Fully Connected) 연산과 공간 컨볼루션 연산이 가장 많은 계산 비용을 요구한다. 본 논문에서는 공간 컨볼루션의 연산량 감소를 줄이기 위한 연구를 진행하였다. 기존 컨볼루션 연산은 컨볼루션 커널과 이미지 데이터 매트릭스에서 슬라이딩 윈도우 접근 방식 또는 행렬곱으로 수행 된다[9]. 이 컨볼루션 연산은 계산 비용이 많이 들기 때문에 커널사이즈 또는 입력 이미지로 대형 이미지를 요구하는 객체 유사도 비교, 이미지 매칭과 같이 커다란 커널을 사용하는 컴퓨터 비전 작업에 적합하지 않다. 일반적으로 3×3 크기의 커널을 주로 사용한다. AlexNet이 CNN 구조를 도입하였을 때 11×11, 5×5크기의 커널을 사용하였다. 이는 훈련 시간이 오래 걸리고 계산 비용이 많이 드는 결과를 보여주었다. 작은 크기의 커널을 주로 사용하는 이유는 커다란 커널을 한번 사용하는 것 보다 더 작은 컨볼루션 레이어를 중첩하는게 분류 정확도에 좋은 영향을 주었기 때문이다. 하지만, 커

다란 커널은 이미지 픽셀 주변 관계성을 요구하는 작업에 효과적으로 영향을 미친다. 따라서 커다란 커널을 사용하는 컨볼루션 연산이 최적화되어 개선된다면 커다란 커널을 요구하는 컴퓨터비전 작업에 적용할 수 있을 것으로 판단된다. 본 논문은 컨볼루션 레이어의 연산과정을 푸리에 도메인에서 수행되는 푸리에 컨볼루션 신경망(Fourier Convolution Neural Network, FCNN)을 제안한다. 컨볼루션 연산과 동일하지만 FFT 방법을 적용하여 연산량 최적화를 진행할 수 있도록 수정 및 개선하는 작업을 진행한다. 제안하는 접근 방법은 특히 더 큰 이미지의 컨텍스트에서 복잡성을 줄이고 결과적으로 네트워크 효율성의 상당한 증가를 제공할 수 있다. 논문의 구성은 다음과 같다. 2장에서 계산비용을 줄이기 위한 관련 연구를 기술했다. 3장에서 제안한 FFT기반 컨볼루션 알고리즘을 서술하였고, 4장에서 제안한 알고리즘의 실험 및 결과를 기술 하였으며, 마지막 5장에서는 결론을 맺는다.

2. 관련연구

CNN은 일반적으로 계산 비용과 메모리 사용을 크게 요구하는 컨볼루션 계층과 완전 연결 계층으로 구성된다. 입력되는 이미지는 컨볼루션 커널과 연산을 통해 데이터의 특징이 추출된다. 커널을 구성하는 가중치 파라미터의 개수가 증가할수록 요구되는 연산량과 메모리가 많아지며 이를 만족하기 위한 하드웨어 스펙 또한 증가한다. CNN을 최적화하기 위한 다양한 연구방법들이 제시되었다. CNN 압축을 위해 사전학습된 네트워크의 가중치 파라미터에 압축 알고리즘을 적용하는 연구가 진행되었다. CNN의 계산 비용을 줄이는 연구는 크게 가중치를 압축 표현하는 신경망 압축 기법과 신경망 구조를 경량화 하는 최적화 방법 등으로 분류할 수 있다. 인공신경망 압축 기법은 사전학습(Pre-training)된 CNN의 컨볼루션 레이어의 커널과 완전 연결 레이어 파라미터의 크기를 줄이는데 중점을 두고 있다. 대표적인 인공신경망 압축 기법으로는 파라미터 연결을 끊는 가지치기(Pruning)기법이 존재한다[10]. 가지치기는 특징 추출에 미미한 영향을 지닌 파라미터는 성능에 미치는 영향이 적다고 판단하여 파라미터를 0으로 바꾸거나 연결을 끊는다. CNN의 경량화 기법은 모델 성능을

개선할 수 있는 네트워크 전체 구조를 개선하는 연구와 컨볼루션 레이어 및 완전연결 레이어 구조를 수정하여 연산량을 줄이는 연구 등이 있다[11,12]. 신경망의 중복성을 제거하며 신경망의 최적화 알고리즘을 제시하였다[13]. 특이값 분해(Singular Value Decomposition)을 완전 연결 계층의 가중치 파라미터에 적용한 연구도 진행되었다[14]. 예측 정확도에 큰 변동없이 신경망을 압축하는 결과를 보여주었다. 기존 행렬 인수 분해 방법보다 더 효과적인 벡터 양자화를 CNN에 적용하여 네트워크를 16-24배로 압축하였다[15]. Vanhoucke et al.은 기존 32비트 부동소수점으로 저장되는 파라미터를 8비트 정수형태로 바꿔 저장함으로써 모델의 용량을 줄였다[16]. Chen et al.은 네트워크 모델의 물리적 용량을 줄이기 위하여 중복 가중치를 단일 매개변수로 공유하는 방식을 적용하였다[17]. 해시 함수를 가중치 파라미터에 적용하여 연결된 가중치를 해시 버킷으로 무작위 그룹화하였다. 폴리 커넥티드 레이어를 전역 평균 풀링으로 변경하여 네트워크의 매개변수를 물리적으로 줄이는 연구가 진행되었다. S Srinivas et al.은 완전 연결 계층의 가중치 파라미터에 ReLU 활성화 함수를 변형한 알고리즘을 추가하였다[18]. 파라미터간 연산 결과로 1과 0의 이진값을 출력한다. 0으로 출력된 파라미터를 제거하는 네트워크 압축 알고리즘 방법을 제안하였다. 다른 압축 알고리즘으로 행렬 분해(Matrix Decomposition)기법이 존재한다. 가중치 파라미터 행렬을 2개 또는 N개 이상의 행렬로 분해하여 가중치 파라미터의 개수를 줄인다. 이러한 압축 기법들을 CNN 모델 등에 적용함으로써 기존의 모델 대비 모델의 크기를 줄일 수 있을 뿐만 아니라 실제 추론 과정에서의 연산량 또한 줄일 수 있다.

3. 제안하는 FFT 기반 컨볼루션 알고리즘

3.1 컨볼루션 알고리즘(Convolution Algorithm)

컨볼루션 알고리즘은 신호처리, 이미지 처리와 같은 분야에서 자주 사용된다. 딥러닝에선 컨볼루션 알고리즘을 사용하기 때문에 CNN으로 사용된다. 일반적인 컨볼루션 알고리즘은 식 1처럼 표현된다[19].

$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau)g(t-\tau)d\tau \quad (1)$$

식 1에서 g는 커널 또는 필터로 불린다. 모든 위치

에 대하여 f와 역(Inverse) g 사이의 교차 영역을 계산한다. 해당 교차 영역은 특정 위치의 컨볼루션 값이다. 반면에 상호 상관(Cross-Correlation)은 두 함수의 슬라이딩 내적(Inner Product)로 알려져 있다. 상호 상관에서 커널은 반전되지 않는다. 컨볼루션의 목적은 입력 이미지에서 유의미한 특징을 추출하는 것이다. 특징이 담고 있는 정보는 커널 형태에 의존한다. 다양한 형태의 커널은 입력 이미지에서 수평, 수직, 대각선, 가장자리와 같은 정보로 이루어진 특징을 추출한다. 각각의 커널은 가중치(weight) 값을 가지고 있고 대량의 학습 데이터를 사용하여 데이터 셋에 존재하는 패턴과 유사한 형태로 학습된다. 추출된 모든 특징을 결합하여 새로운 특징맵으로 조합하고 이를 다음 레이어에서 입력으로 사용한다. 컨볼루션은 특징맵 전체 영역에 동일한 커널을 사용하여 연산 되므로 가중치가 공유(sharing)되는 효과를 얻는다. 동시에 영역 연산이 진행되므로 맵을 구성하는 공간(spatial) 관계성을 분석한다. 가중치 공유와 공간 관계성 분석을 통해 커널 하나로 다양한 위치에서 특징을 추출하는 결과를 얻는다. 컨볼루션이 지닌 특성은 다양한 컴퓨터 비전 작업에서 효율적으로 작용한다. 특정 구성 요소(context)가 다른 구성 요소와 특정 공간적으로 관계가 있는 개체를 식별하는 작업에 특화된다.

3.2 FFT 기반 컨볼루션

이미지 처리에서 FFT는 이미지에 적용되어 주파수 영역에서 이미지를 나타내는 실수 및 허수 성분으로 분해된다. 2D 이미지의 FFT는 식 2와 식 3으로 표현한다.

$$F(x,y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-i \times 2 \times \pi \times (x \frac{m}{M} + y \frac{n}{N})} \quad (2)$$

$$f(x,y) = \frac{1}{M \cdot N} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} F(m,n) e^{i \times 2 \times \pi \times (x \frac{m}{M} + y \frac{n}{N})} \quad (3)$$

식에서 f(m,n)은 이미지 평면의 픽셀 좌표(m,n)을 나타낸다. F(x,y)는 푸리에 주파수 평면에서 이미지를 나타내는 함수다.

Fig. 1은 기본 컨볼루션 연산과정을 보여준다. 4×4 크기의 특징맵과 3×3 크기의 커널을 사용한다. 컨볼루션 연산은 전처리 과정으로 슬라이딩 윈도우 방식을 사용한다. 커널 위치에 대응되는 특징맵의 요소값

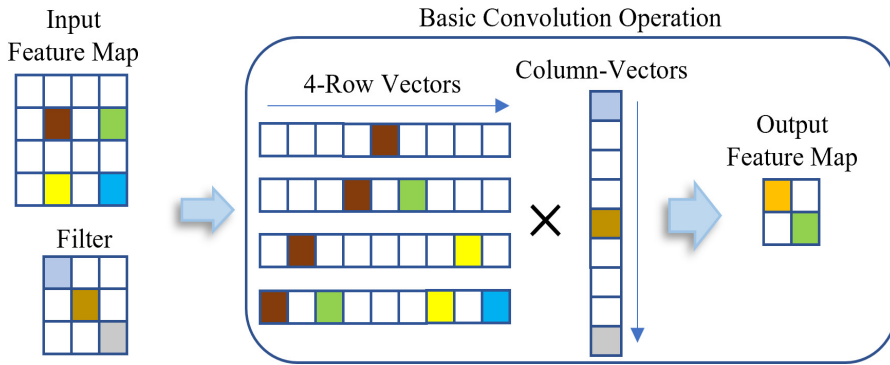


Fig. 1. Process of Basic Convolution Operation.

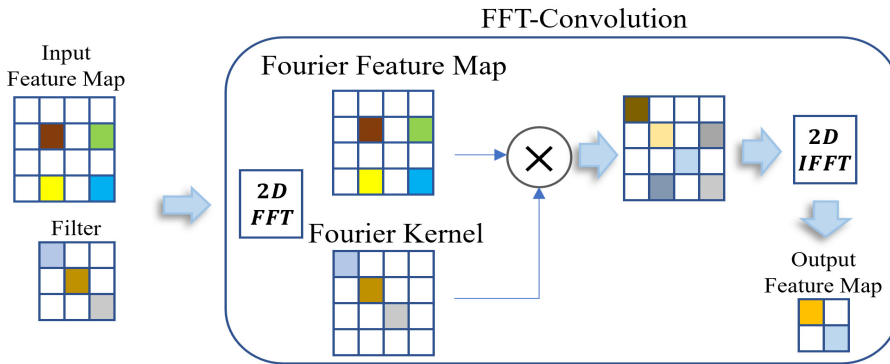


Fig. 2. Process of FFT-Convolution Operation.

을 추출한다.

Fig. 1과 같이 행벡터와 열벡터 각각을 곱셈하여 계산한다. 특징맵에서 행벡터를 추출하는 전처리 과정과 곱셈 작업이 진행되는 과정에서 많은 연산량이 요구된다. 입력 특징맵의 크기가 255×255, 커널의 크기가 3×3 일 때, 요구되는 Row-Vector의 개수는 63,001개다. 각 Row-Vector는 커널이 위치하는 요소를 포함하기 때문에 전처리에 요구되는 시간 복잡도는 증가한다. 특징맵 사이즈와 커널 사이즈 설정에 따라 이와 같은 현상이 더욱 심화된다.

Fig. 2는 FFT-Convolution을 보여준다. Fig. 1과 다른점은 특징 맵과 필터에 2D FFT(Fast Fourier Transform)를 적용하여 주파수 도메인으로 변환하는 것이다.

Fig 3과 같이 커널 사이즈를 특징맵 크기로 매칭시켜주는 작업을 진행한다. 이때 필터 영역은 왼쪽 상단 모서리에 위치한다. 나머지 영역은 0-패딩(zero-padding)을 적용한다. 0-패딩은 여백을 0으로

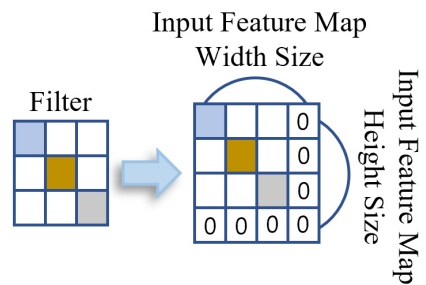


Fig. 3. Padded Convolution Filter.

채우는 것이다. 0으로 채워줌으로써 FFT 계산에 영향을 미치지 않으며 주파수 영역의 해상도를 증가시킬 수 있다. 주파수 도메인으로 변환된 입력 특징맵과 커널을 아다마르 곱셈을 사용하여 연산한다. 이는 공간 도메인에서 컨볼루션과 같은 효과를 얻는다. 마지막으로 결과물에 Inverse FFT를 적용하여 공간도메인으로 변환한다. 단순히 특징맵 사이즈 비교를 통해, 여백 부분을 0으로 채우는 작업이 전처리 과정에

Table 1. Specification of Experiment Environment.

		Version
H/W	O/S	Windows 10
	CPU	Ryzen 5600X
	RAM	32G
	GPU	NVIDIA GeForce RTX 2070
S/W	Deep learning Framework	Pytorch 1.8

서 이루어지고, 특징맵을 구하는 과정에서 한번만 곱셈이 진행되므로 연산 속도에서 표준 컨볼루션 보다 이점을 많이 얻을 수 있다.

4. 실험 결과

4.1 실험 환경

실험에 사용한 하드웨어 사양은 다음과 같다. CPU는 AMD Ryzen 5 5600X 시리즈, 그래픽카드는 2,304개의 CUDA cores로 구성된 NVIDIA GeForce RTX 2070 시리즈를 사용하였다. RAM은 DDR4 32GB, 데이터셋의 빠른 I/O를 보장 받기 위하여 SSD를 사용하였다. Table 1은 실험에 사용한 하드웨어 상세 스펙이다.

사용된 라이브러리는 CUDA(Compute Unified Device Architecture) Toolkit 11.1 버전, cuDNN(Cuda Deep Neural Network Library)은 8.0버전을 사용하였다. 구현 언어는 Python 3.8 버전, 네트워크 설계에 사용한 딥러닝 프레임워크는 PyTorch 1.8 버전을 사용하였다. 핵심 FFT 알고리즘은 Pytorch에 내장된 Cudnn 기반 FFT 모듈을 사용하였다.

4.2 실험결과

본 논문에서는 각각 동일한 종류의 파라미터와 수치를 사용하여 네트워크를 학습하였다. 배치 사이즈는 64, 최적화 함수(Optimizer)는 Adam, 학습률(Learning Rate)은 0.001을 설정했다. 사용된 네트워크는 2층의 컨볼루션 레이어, 2층의 최대 풀링 레이어, 2층의 완전 연결 레이어를 적용하였다. 활성화 함수로는 ReLU를 사용하였다. 첫 번째 실험은 MNIST 데이터셋을 사용하여 제안하는 Fourier CNN과 기본 CNN을 비교하였다.

Fig. 4는 학습데이터를 사용한 실험결과를 보여준다. 주황색 선은 네트워크 분류 정확도(Accuracy), 파란색 선(Line)은 학습 손실(Loss) 값을 보여준다. 정확도와 손실값에 있어서 두 네트워크는 눈에 주목할만한 성능차이가 발생하지 않았다. x축의 숫자 1~20은 학습 단계의 epoch를 의미한다. 두 네트워크 모두 5 epoch 까지 정확도가 상승하다가 이후 epoch부터 99%에 수렴하는 결과를 보여주었다. 손실값 또한 점진적으로 감소하는 트렌드를 보여주었다. 이 결과를 통해 제안하는 Fourier CNN이 정확도 손실에 영향을 미치지 않는다는 사실을 알 수 있다. 테스트셋의 결과에서도 Fourier CNN은 98.89%, Basic CNN은 98.85%를 보여주었다. 하지만 속도 측면에서 차이가 발생하였다. 5 Epoch 학습에 Basic CNN은 49.94초가 걸렸고, FCNN은 46.32초를 달성하며 약 7.2% 빠른 연산 속도를 달성하였다.

추가적으로 큰 커널을 사용하여 실험을 진행하였다. Fig. 5는 1024×1024 입력 이미지와 7계층의 컨볼루션 레이어로 구성된 네트워크를 사용한 실험 결과를 보여준다. 각 컨볼루션 레이어의 커널 크기는 실

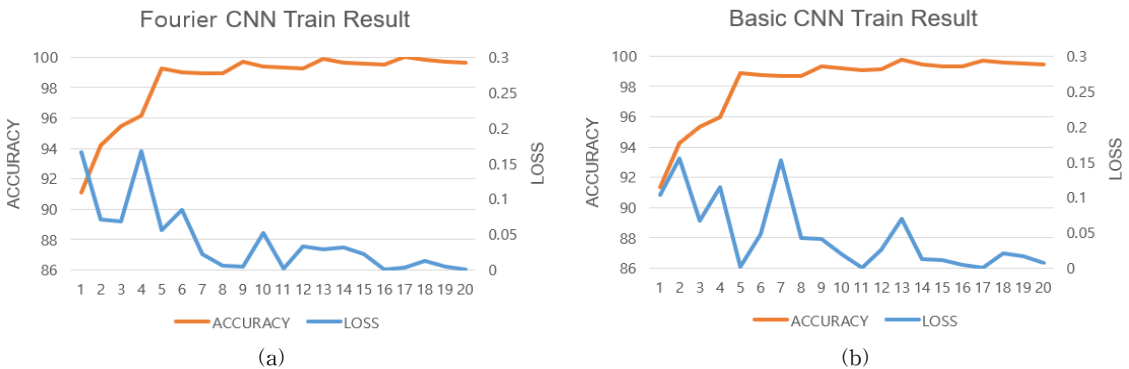


Fig. 4. Result of MNIST Dataset. (a) Result of Fourier CNN and (b) result of basic CNN.

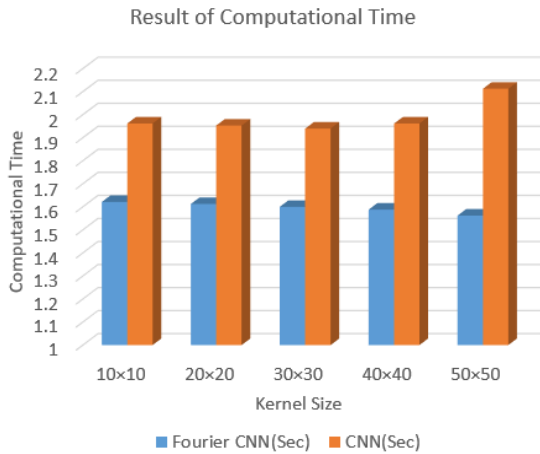


Fig. 5. Result of Computational Time Using Different Size of Kernel.

험마다 10, 20, 30, 40, 50으로 10씩 증가하여 실험하였다. Fourier CNN의 경우 10×10 크기의 커널부터 연산에 각각 1.6215, 1.6124, 1.5997, 1.5881, 1.5623초가 소요되었고, CNN의 경우 1.9619, 1.9527, 1.9397, 1.9617, 2.1126초를 달성하였다. Fourier CNN은 커널 크기가 커질수록 연산시간이 점진적으로 감소하는 추세를 보여주었다. CNN은 10×10 ~ 40×40 크기 구간에서는 약 1.9초로 동일한 연산시간이 소모되었고 50×50 크기에서 2.1초를 보여주었다. Fourier CNN은 CNN과 비교했을 때 연산시간에서 평균 19% 빠른 속도를 달성하였다.

5. 결 론

본 논문에서 제안한 Fourier CNN은 큰 사이즈의 커널을 사용했을 때 연산 속도를 향상하기 위해 고안되었다. 일반 컨볼루션 연산의 수학적 특징은 Fourier 연산과 상당 부분 유사하다. Fourier 기반 컨볼루션은 일반 컨볼루션과 비교했을 때 커널 크기가 커질수록 필요한 전처리 과정과 연산 과정이 단순화 되어 연산 속도에 이점을 얻었다. MNIST 데이터셋을 사용하여 네트워크를 학습하였다. 제안하는 방법과 기본 CNN은 각각 98% 정확도를 보여주며 정확도 측면에서 큰 차이를 보여주지 않았다. 반면 연산 속도에서는 약 7%의 차이를 보여주며 더 빠른 연산속도를 달성하였다. 1024×1024 크기의 커다란 입력 이미지를 사용한 실험에서 Fourier CNN이 평균적으로 19% 빠른 속도를 보여주었다. 제안하는 방법은 커다

란 커널 또는 패치를 사용하는 특수한 작업에 적합할 것으로 판단된다. 특히 이미지 안에서 특정 객체를 찾는 이미지 매칭 작업과, 두 이미지 사이에서의 유사도를 비교하는 컴퓨터 비전 분야에 효과적으로 적용할 수 있을 것으로 판단된다. 추후, 푸리에 연산에 소요되는 변환 비용을 감소하기 위한 연구를 진행하고자 한다. 네트워크 학습 또는 테스트 단계 초기에만 주파수 변환이 이루어지고, 이후 모든 과정은 주파수 도메인에서 학습 가능한 전주기 푸리에 연산 컨볼루션 알고리즘을 연구하고자 한다.

REFERENCE

- [1] S.W. Park, S.H. Kim, S.C. Lim, and D.Y. Kim, "Performance Comparison of Commercial and Customized CNN for Detection in Nodular Lung Cancer," *Journal of Korea Multimedia Society*, Vol. 23, No. 6, pp. 729-737, 2020.
- [2] S.C. Lim and D.Y. Kim, "Semantic Segmentation using Convolutional Neural Network with Conditional Random Field," *The Journal of the Korea Institute of Electronic Communication Sciences*, Vol. 12, No. 3, pp. 451-456, 2017.
- [3] S.C. Lim, S.W. Park, J.C. Kim, and C.S. Ryu, "Object Tracking Algorithm using Feature Map based on Siamese Network," *Journal of Korea Multimedia Society*, Vol. 24, No. 6, pp. 796-804, 2020.
- [4] Y. LeCun, B. Boser, J.S. DDenker, D. Hender-son, R.E. Howard, W. Hubbard, and L.D. Jackel, "Backpropagation Applied to Hand-written Zip Code Recognition," *Neural Computation*, Vol. 1, No. 4, pp. 541-551, 1989.
- [5] O. Russakovsky, et al, "Imagenet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, Vol. 115, No. 3 pp. 211-252, 2015.
- [6] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," *Proceeding of International Conference on Advances in Neural Information Processing Systems*, pp. 1097-

- 1105, 2012.
- [7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016.
- [8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9, 2015.
- [9] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated Recognition, Localization and Detection Using Convolutional Networks," *arXiv preprint*, arXiv:1312.6229, 2013.
- [10] S.C. Lim and D.Y. Kim, "Apply Locally Weight Parameter Elimination for CNN Model Compression," *The Korea Institute of Information and Communication Engineering*, Vol. 22, No. 9, pp. 1165-1171, 2018.
- [11] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shuffle Net: An Exteremey Efficient Convolutional Neural Network for Mobile Devices," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6848-6856, 2018.
- [12] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint*, arXiv:1704. 04861, 2017.
- [13] M. Denil, B. Shakini, L. Dinh, M.A. Ranzato, and N.D. Freitas, "Predicting Parameters in Deep Learning," *Proceedings of the 26th International Conference on Neural Information Processing Systems*, Vol. 2, pp. 2148-2156, 2013.
- [14] E.L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting Linear Structure within Convolutional Networks for Efficient Evaluation," *Advances in Neural Information Processing Systems*, pp. 1269-1277, 2014.
- [15] Y. Gong, L. Liu, M. Yang, and L. Bourdev, "Compressing Deep Convolutional Networks Using Vector Quantization," *arXiv preprint*, arXiv:1412.6115, 2014.
- [16] V. Vanhoucke, A. Senior, and M.Z. Mao, "Improving the Speed of Neural Networks on Cpus," *Proceedings of Deep Learning and Unsupervised Feature Learning NIPS Workshop*, Vol. 1. pp. 1-8, 2011.
- [17] W. Chen, J.T. Wilson, S. Tyree, K.Q. Weinberger, and Y. Chen, "Compressing Neural Networks with the Hashing Trick," *Proceedings of International Conference on Machine Learning*, pp. 2285-2294, 2015.
- [18] S. Srinivas and R.V. Babu, "Learning Neural Network Architectures using Backpropagation," *arXiv preprint*, arXiv:1511.05497, 2015.
- [19] Convolutional Neural Networks (CNN): Step 1-Convolution Operation. <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-1-convolution-operation> (accessed November 06, 2021).



임 수 창

2015년 순천대학교 컴퓨터공학과
졸업(공학사)
2017년 순천대학교 대학원 컴퓨
터공학과 졸업(공학석사)
2019년 순천대학교 대학원 컴퓨
터공학과 박사수료

2020년~현재 (주)티에프 기업부설연구소 책임연구원
관심분야: 컴퓨터비전, 영상처리, 딥러닝, 객체추적, 시
계열분석



김 종 찬

2000년 순천대학교 컴퓨터공학과
졸업(이학사)
2002년 순천대학교 대학원 컴퓨
터공학과 졸업(이학석사)
2007년 순천대학교 대학원 컴퓨
터공학과 졸업(이학박사)

2013년 서울대학교 자동화 시스템 연구소 선임연구원
2021년~현재 순천대학교 컴퓨터공학과 조교수
관심분야: 영상처리, HCI, 콘텐츠, 컴퓨터그래픽스, 기계
학습, 데이터 분석 및 예측