

## GPU 를 이용한 콘볼루션 뉴럴 네트워크 기반 초해상화 설계 및 구현

기세환, 최재석, 김수예, 김문철  
한국과학기술원 전기 및 전자 공학과

shki@kaist.ac.kr, jschoi14@kaist.ac.kr, sooyekim@kaist.ac.kr, mkim@ee.kaist.ac.kr

### Accelerating Deep learning based Super resolution algorithm using GPU

Sehwan Ki, Jaeseok Choi, Sooye Kim, Munchurl Kim

Korea Advanced Institute of Science and Technology Dep. Of Electronic Engineering

#### 요 약

본 논문에서는 딥 콘볼루션 신경망 구조를 사용하여 학습된 초해상화 알고리즘을 GPU 프로그래밍을 통해 실시간 동작이 가능하도록 하는 방법을 제시하였다. 딥 러닝이 많이 대중화 되면서 많은 영상처리 알고리즘이 딥 러닝을 기반으로 연구가 되었다. 하지만 계산 량이 많이 필요로 하는 딥 러닝 기반 알고리즘은 UHD 이상의 고해상도 영상처리에는 실시간 처리가 어려웠다. 이런 문제를 해결하기 위해서 고속 병렬 처리가 가능한 GPU 를 사용해서 2K 입력영상을 4K 출력 영상으로 확대하는 딥 초해상화 알고리즘을 30 fps 이상의 처리 속도로 동작이 가능하도록 구현을 하였다.

#### 1. 서론

최근 들어 딥 러닝이 영상 처리에 매우 압도적인 성능을 보이면서 많은 딥 러닝 기반 알고리즘이 제안되고 있다. 기존의 사람이 수식적으로 얻은 영상 내 특징 값을 사용하는 것보다 깊은 신경망을 통해 추출되는 특징들로 입력 영상과 출력 영상 간의 매핑을 하는 것이 더 효과적이라는 것을 결과적으로 보였다. 기존의 딥 러닝 기반 영상 처리는 영상 내의 객체를 구별하거나 영상의 정보를 추출해서 분류하는 문제를 푸는 데 많이 사용되어 왔다. 최근에는 저해상도와 저화질을 가지는 영상을 딥 러닝 구조의 입력으로 하고 출력으로 고해상도 고화질의 영상이 나오도록 학습하는 방법에 대해서도 많은 연구가 진행되고 있다. 그 중 저해상도의 영상을 입력을 고해상도 영상으로 출력하는 방법인 초해상화 연구에서도 딥 러닝을 사용한 알고리즘이 가장 높은 성능을 보였다[1]. 하지만 최근 실제 방송과 실시간 스트리밍 영상의 해상도가 급격히 높아지면서 초해상화 연구의 목표 영상의 크기도 FHD 를 넘어서 UHD 해상도가 되어가고 있다. 이러한 상황에서 깊은 신경망 구조로 인해 계산 량이 많고 복잡한 딥 러닝 기반의 초해상화 알고리즘은 실시간 환경에서는 적용이 어려웠다. 하지만 콘볼루션 신경망 구조의 경우 독립적인 단순 반복 곱셈이 반복되는 것이기 때문에 병렬 처리에 최적화되어 있는 GPU 를 사용한다면 상당한 가속화를 할 수 있을 것이다. 딥 러닝 알고리즘을 학습하고 실험하는데 편의를 제공하는 플랫폼인 Caffe, TensorFlow, Matconvnet 의 경우에도 모두 GPU 를 사용할 수 있도록 구현이 되어 있다[2][3][4]. 하지만 이는 단순히 사용자가 수정할 수 없고 사용자의 알고리즘에 최적화된 GPU 프로그래밍을 제공하지는 않는다. 따라서 본 논문에서는 CUDA 라는 GPU 프로그래밍 언어를 사용해서 딥

러닝 기반 초해상화 알고리즘을 구현하고 동작 속도 관점에서 최적화를 통해 실시간 처리가 가능하도록 하였다.

본 논문의 구성은 2 절에서 딥 러닝 기반 초해상화 알고리즘에 대해서 소개하고 그 알고리즘을 CUDA 로 구현한 내용에 대해서 설명한다. 그리고 3 절에서는 구현 결과에 대한 실험 결과를 분석하고 마지막으로 4 절에서 논문의 결론을 맺는다.

#### 2. 딥 러닝 기반 초해상화 알고리즘 가속화

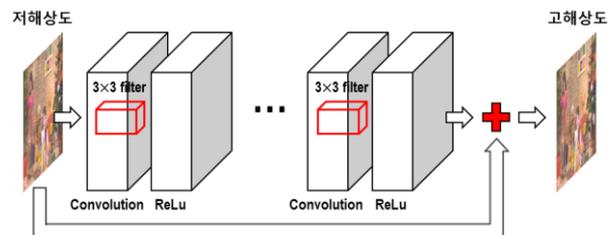


그림 1. 딥 러닝 기반 초해상화 알고리즘

본 논문에서 사용한 딥 러닝 기반 초해상화 알고리즘은 그림 1 과 같다. 원본 영상과 바이큐빅 보간 법으로 확대한 한 영상 간의 차이를 학습하는 레지듀얼 구조를 가지면서 3x3 필터 사이즈 콘볼루션과 ReLu 로 이루어지는 콘볼루션 레이어로 이루어져 있다. 이와 같은 구조로 미리 Matconvnet 을 통해 학습을 시켜 얻은 필터 변수 값과 바이어스 값을 사용하여 테스트 하는 환경을 CUDA C 를 사용해서 구현하였다.

2048×1080, 300 프레임	Matconvnet	GPU 가속화 구현		
사용된 GPU 의 수 (개)	1	1	2	3
GPU 초기화 시간	-	1.321 s	1.736 s	2.467 s
YUV 파일 열기 시간	294 s (0.980 s)	0.034 s	0.569 s	0.040 s
YUV 프레임 읽기 시간 & GPU 처리 시간 (1 프레임 처리 시간)		26.04 s (0.086 s)	12.65 s (0.042 s)	8.07 s (0.027 s)
속도 향상	1	<b>10.73 배</b>	<b>19.66 배</b>	<b>27.80 배</b>
PSNR	38.88 dB			
SSIM	0.9624			

표 1. Matconvnet 플랫폼 동작 속도와 GPU 가속화 구현 속도 비교 결과

GPU 프로그래밍을 위해서는 CPU 의 메인메모리에서 저장되어 있는 영상 정보들을 GPU 전역 메모리로 옮겨야 하는 추가적인 시간이 걸리지만 병렬로 동작되는 데에서 얻는 이점이 그런 단점을 보완하고도 충분하다고 볼 수 있다. 구현 시 콘볼루션의 계산은 NVIDIA 에서 제공하는 딥 러닝 라이브러리인 CuDNN 의 함수를 사용하여 구현을 하였고 Leaky ReLU 와 바이어스와 Unfold 부분은 CUDA C 를 사용해서 직접 구현한 커널 함수를 사용하여 구현하였다 [5]. GPU 프로그래밍의 가속화 최적화에서 메모리 접근 횟수가 속도에 크게 영향을 주기 때문에 전역 메모리에 접근하는 횟수를 최소한으로 줄이도록 연속적인 함수들을 하나로 합치고 접근 속도가 빠른 상수메모리를 사용한 구현을 통해 최적화 하였다. 그리고 추가적인 가속화를 위해서 프레임 별 독립적으로 GPU 에 할당해서 처리가능하도록 구현하였기 때문에 사용자가 동작 시에 사용할 GPU 수를 선택이 가능하다.

### 3. 가속화 구현 결과

실험 환경은 Ubuntu 14.04 버전에서 Intel Xenon CPU 와 NVIDIA TITAN X pascal GPU 를 사용해서 실험을 진행하였다. 실험에 사용된 영상은 2048x1080 해상도를 가진 30fps 영상 300 프레임을 4096x2160 해상도로 초해상화하는 실험을 진행하였다. 표 1 에서 볼 수 있듯이, GPU 를 사용한 Matconvnet 환경에서 동일한 테스트를 한 결과 YUV 프레임 저장 시간을 제외하고 걸리는 시간은 프레임당 1 초 정도 걸리는 것은 볼 수 있었다. 그에 비해 CUDA C 를 이용해서 가속화를 한 경우 동일한 개수의 GPU 를 사용했을 경우 프레임당 0.086 초로 약 11 배 더 빠르게 동작되는 것을 확인 할 수 있었다. 한 개의 GPU 를 사용하였을 경우 11.6 fps 의 처리 속도를 가지는 것을 확인 할 수 있었다. 실시간 구현에 단일 GPU 만으로 최적화를 통해 속도 향상이 부족해서 다중 GPU 를 사용할 수 있는 환경을 구축하였다. 초해상화 알고리즘은 프레임 별 동작이 이루어 지는 것이기 때문에 다중 GPU 를 각각 독립적으로 한 개의 프레임 마다 초해상화를 할 수 있도록 구현을 한 결과 사용된 GPU 개수만큼 동작 시간의 속도 향상이 선형적으로 증가되는 것을 확인 할 수 있었다. 따라서 다중 GPU 를 사용한 결과 프레임당 처리 시간이 0.027 초로 즉, 37 fps 로 실시간 처리가 가능한 것을 보였다. 물론 GPU 를 통한 가속화를 통한 결과 영상은 Matconvnet 을 통해 얻은 영상과 완전히 동일한 결과인지를 동일한 PSNR 값과 SSIM 값을 가지는 것으로 확인하였다.

### 4. 결론

이 논문에서는 최근 많이 연구되고 있는 딥 러닝 기반 초해상화 알고리즘을 실시간 처리가 가능하도록 GPU 를 통한 가속화 방법을 제시하였다. 딥 러닝이 많은 관심을 받고 있지만 대부분의 연구들이 제공되는 딥 러닝 플랫폼 위에서 구현을 하는 정도에 그친다. 그에 반해 본 연구에서는 그 아래의 CUDA 구현을 통해 원하는 목표 딥 러닝 구조에 최적화된 GPU 가속화를 가능하도록 할 수 있었다.

### Acknowledgement

이 논문은 2017 년도 정부(미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2017R1A2A2A05001476)

### 참고문헌

- [1] Kim, Jiwon, Jung Kwon Lee, and Kyoung Mu Lee. "Accurate image super-resolution using very deep convolutional networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.
- [2] Jia, Yangqing, et al. "Caffe: Convolutional architecture for fast feature embedding." Proceedings of the 22nd ACM international conference on Multimedia. ACM, 2014.
- [3] Abadi, Martín, et al. "Tensorflow: Large-scale machine learning on heterogeneous distributed systems." arXiv preprint arXiv:1603.04467, 2016.
- [4] Vedaldi, Andrea, and Karel Lenc. "Matconvnet: Convolutional neural networks for matlab." Proceedings of the 23rd ACM international conference on Multimedia. ACM, 2015.
- [5] Chetlur, Sharan, et al. "cudnn: Efficient primitives for deep learning." arXiv preprint arXiv:1410.0759, 2014.