

젯슨 나노 기반 활성화 함수에 따른 초해상화 알고리즘 성능 분석 연구

임재윤¹, 김유민¹, 김용우¹

¹상명대학교 시스템반도체공학과

jaeyun0126@gmail.com, 022501@naver.com, yongwoo.kim@smu.ac.kr

A study on the Performance Analysis of Super-Resolution Algorithms by the activation functions using Jetson Nano

Jae-Yoon Lim¹, Yu-Min Kim¹, Yongwoo Kim¹

¹Dept. of System Semiconductor Engineering, Sangmyung University

요 약

최근 고해상도 영상이 필요하게 되었으며, 저해상도 영상을 고해상도 영상으로 변환하는 딥러닝 기반의 초해상도 알고리즘에 대한 연구가 활발히 진행되고 있다. 그럼에도 불구하고 딥러닝 기반의 초해상도 알고리즘은 하드웨어의 한계로 인해 임베디드 시스템에서 실행시간이 느린 단점이 있다. 본 논문에서는 심층신경망 기반의 초해상도 알고리즘의 네트워크 구조를 제시하고 다양한 활성화 함수에 따른 화질 및 실행시간 성능을 분석한다. 실험 결과, 젯슨 나노보드의 다양한 활성화 함수 중 화질과 실행 시간의 관계에서 도출한 최적의 활성화 함수가 PReLU 함수임을 확인하였다.

1. 서론

최근 범죄에 대한 안전이 중요시되는 만큼 CCTV의 개수도 증가하고 있다. 행정안전부가 최근에 공개한 행정안전통계연보에 따르면 공공기관이 설치한 CCTV는 114만8천대로 매년 10% 가까이 증가하는 추세이다[1]. 4K/8K UHD (Ultra-High Definition)의 초고해상도 디스플레이가 등장하고 있는 시대에도 불구하고 지자체 CCTV 절반이 저화질로 인해 식별이 어렵다는 조사 결과가 나왔다[2]. 저화질의 기기를 모두 교체하는 것은 현실적으로 실현하기 어려워 기술적인 개발이 필요하다는 목소리가 나오고 있다. 대표적인 것이 저화질로 획득된 영상을 고화질로 변환하는 것인데, 고해상도의 영상을 얻기 위해서는 저해상도의 영상을 고해상도의 영상으로 변환하는 초해상화(Super Resolution) 알고리즘이 요구된다. 기존에는 보편적인 양선형 보간법(Bilinear Interpolation), 바이큐빅 보간법(Bicubic Interpolation), 비스플라인 보간법(B-spline Interpolation) 등이 사용됐으나, 최근에는 화질 개선이 뛰어난 딥러닝 기반 심층 신경망(Convolutional Neural Network)을 이용한 초해상화 알고리즘이 많이 연구된다. 기존의 딥러닝을 이용한 초해상화 알

고리즘은 화질 개선 성능은 뛰어나지만, 여전히 연산시간이 오래 걸린다는 문제점이 제기된다.

본 논문에서는 젯슨 나노 보드 상에서 기존에 제안된 초해상화 알고리즘인 SRCNN[3], FSRCNN[4], ESPCN[5]을 구현하고, 그중에서 젯슨 나노 보드에서 이미지 화질 개선 정도 및 수행시간을 비교 분석하여 성능이 가장 적합한 최적의 활성화 함수를 도출하고자 한다.

2. 심층 신경망에 사용되는 함수

모든 CNN 연산에서 이미지와 필터의 컨볼루션 연산 후, 필터값을 비선형으로 바꿔주기 위해 활성화 함수 연산을 거친다. 활성화 함수 Tanh, ReLU, LeakyReLU, PReLU, ELU의 기능에 대해 검토해 보았다. 식 (1)은 기존 머신 러닝에서 자주 사용되었던 Sigmoid 함수를 변형하여 만든 Tanh(Hyperbolic Tangent) 함수이다. 중간값을 0으로 옮겨 음수 값, 양수 값 모두 가질 수 있다. 하지만 입력값이 증가할 시 기울기 값이 소실되는 현상 (Gradient Vanishing)이 존재한다.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (1)$$

표 1 젓슨 나노 보드에서 측정한 SRCNN, FSRCNN, ESPCN 네트워크의 활성화 함수별 이미지 셋 Set5, Set14, Urban100에 대한 PSNR, SSIM, Computation Time 평균값

Activation Function	SRCNN[3]			FSRCNN[4]			ESPCN[5]		
	PSNR (dB)	SSIM	Computation Time (sec)	PSNR (dB)	SSIM	Computation Time (sec)	PSNR (dB)	SSIM	Computation Time (sec)
ReLU	28.33	0.8883	3.0607	27.28	0.8194	0.2508	27.48	0.8254	0.2969
Tanh	28.67	0.8943	3.1045	27.28	0.8181	0.2810	27.17	0.8154	0.2993
LeakyReLU	28.27	0.8878	3.0537	27.27	0.8189	0.2641	27.48	0.8251	0.3026
PReLU	28.15	0.8863	3.1689	27.35	0.8214	0.2374	27.49	0.8256	0.3037
ELU	28.38	0.8896	3.1078	27.26	0.8183	0.2433	27.22	0.8171	0.3119
GELU	28.10	0.8869	3.1732	27.21	0.8164	0.2459	27.17	0.8143	0.2987
Swish	28.10	0.8870	3.1026	27.24	0.8166	0.2465	26.98	0.8088	0.3083
Mish	28.25	0.8894	3.5398	27.23	0.8160	0.2589	27.03	0.8104	0.2970
HardSwish	28.23	0.8882	3.1402	27.13	0.8127	0.2369	26.58	0.7978	0.3142

식 (2)는 SRCNN에 사용되는 ReLU(Rectified Linear Unit) 함수이다. ReLU 함수는 입력값 x 가 0보다 작으면 0을, 양수를 입력받으면 그대로 출력한다. 연산 비용이 많이 들지 않으며 구현이 매우 간단하다. 하지만 음수값이 들어오게 되면 기울기 값이 0 즉, 모든 뉴런이 0이 되는 Dying ReLU 현상이 발생한다.

$$ReLU(x) = \max(0, x) \quad (2)$$

식 (3)을 보면 ReLU와 달리 음수 부분을 '0'으로 매핑하지 않고 압축하므로 x 값이 음수로 들어오더라도 미분 값이 0이 되지 않고 항상 작은 양수 값을 출력하는 LeakyReLU 함수를 볼 수 있다.

$$LeakyReLU(x) = \max(0.01x, x) \quad (3)$$

식 (4)은 Dying ReLU 현상을 보완하기 위해 만들어진 활성화 함수 PReLU(Parametric Rectified Linear Unit)가 있다. 음수인 입력값 x 에 작은 값을 곱하는 LeakyReLU와 달리 음수인 입력값을 학습하여 출력한다.

$$PReLU(x) = \max(0, x) + \alpha \min(0, x) \quad (4)$$

ELU(Exponential Linear Unit) 함수인 식(5)은 ReLU와 LeakyReLU처럼 기울기 소실 현상을 해소하는 특징을 가지고 있다. x 값이 0에 가까워질수록 학습 속도가 빨라진다는 장점이 있다.

$$ELU(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha(\exp(x) - 1), & \text{if } x \leq 0 \end{cases} \quad (5)$$

이외에도 비선형성을 띄지 않고, 정규분포를 띄는 특성을 가진 GeLU, ReLU와 다르게 작은 음수값의 입력에도 출력값이 음수라는 장점이 있고, 모든 입력값에 대해 미분이 가능한 Swish[6]와 Mish[7]가 있다. ReLU 함수와 선형 함수 사이를 비선형적으로 보간하는 Hardswish[8] 함수도 존재한다.

3. 실험 및 결과

3.1 실험 환경

활성 함수에 따른 초해상화 알고리즘을 평가하기 위해 NVIDIA사의 젓슨 나노 보드로 구현하였다. 젓슨 나노 보드의 사양은 CPU에는 ARM의 A57 프로세서가 탑재되어 있고, GPU는 128-core Maxwell 코어가 포함되어 있다. 메모리는 64-bit LPDDR4 4GB의 용량을 가지고 있다. 본 논문에서는 젓슨 나노 보드의 GPU를 사용하였으며, 운영체제는 Ubuntu 18.04의 환경에서 진행하였다. 초해상화 알고리즘을 구현하는 프레임워크로 Pytorch로 사용하였고, 1.8.0의 버전을 이용하였다. 또한 이미지의 화질 개선 정도를 시각적으로 보여주기 위해 파이썬 기반의 Flask 프레임워크를 이용하여 웹으로 구현하였다.

훈련 데이터셋은 91개의 이미지로 구성된 데이터셋[9]을 이용하였다. 초해상화 네트워크 훈련을 위해 원본 91개의 이미지로부터 24,800개의 서브 이미지로 분해하였다. 화질 개선 성능을 평가하기 위한 데이터셋은 널리 쓰이는 Set5, Set14, Urban100을 사

용하였다. Set5는 사람, 동물들로 이루어진 5장의 이미지 파일 셋이며, Set14는 흑백사진을 포함한 14장, Urban100은 도시 사진 100장으로 이루어져 있다. 이미지의 크기는 228×344부터 1280×960까지로 이루어져 있다.

SRCNN[3], FSRCNN[4], ESPCN[5] 훈련과정은 모두 동일하게 진행하였다. 스케일은 3으로 설정하여 이미지를 3배 키웠으며, 초기 학습률(learning rate)은 1e-3으로 설정하였다. 배치 사이즈(batch size)는 16으로 하였고, 에폭 수(num epoch)는 200으로 설정하여 가장 훈련 값이 좋은 에포크를 best epoch로 저장하였다. SRCNN, FSRCNN, ESPCN, 각 네트워크 별로 Tanh, ReLU, LeakyReLU, PReLU, ELU, GELU, Swish, Mish, HardSwish 함수에 대해 훈련하였다. 각 네트워크의 훈련의 옵티마이저로는 ADAM을 사용하였다.

3.2 초해상화 성능 평가 지표

이미지의 화질을 평가하기 위해 세 가지 측면에서 고려하였다. 첫 번째 항목은 최대 신호 대 잡음비 (Peak Signal-to- Noise Ratio, PSNR)이다. 영상, 이미지에 대한 손실정보를 측정하며, 높은 PSNR을 가질수록 화질이 좋음을 뜻한다. 평균 제곱 오차 함수(Mean Squared Error, MSE)를 사용하여 PSNR을 구할 수 있다.

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (6)$$

식 (6)은 MSE 함수 식이며 실제로 측정한 값 y_i 에서 예측한 값 \hat{y}_i 를 뺀 후 제곱하여 데이터의 크기 N 으로 나눈다. 오차의 제곱에 대해 평균을 구한 값이다. MSE 값이 적을수록 오차의 크기가 작음을 뜻하므로 정확하게 예측했음을 알 수 있다. 수식 (6)를 이용하여 PSNR을 계산한다. PSNR은 로그 스케일로 계산하여 단위는 dB를 사용한다. 무손실 영상일 경우 MSE 값이 0이므로 PSNR은 측정되지 않는다. 두 번째는 인간의 시각적 화질 차이를 기준으로 평가하는 Wang *et al.* [10]이 제안한 Structural Similarity Index (SSIM)이다. 원본 이미지와 초해상화한 이미지 두 장을 가지고 유사도를 측정한다. 0~1 사이의 값을 가지며 1에 가까울수록 화질이 좋음을 뜻한다. 휘도, 대비, 구조 세 가지 측면으로 이미지의 품질을 측정한다. 딥러닝 알고리즘은 연산

복잡도가 높으므로 젓슨 나노 보드에서 결괏값이 도출되는 시간을 측정하였다.

3.3 활성 함수에 따른 성능 분석 결과

SRCNN, FSRCNN, ESPCN 각 네트워크에서 PSNR, SSIM, Computation Time 세 가지 지표를 이용하여 위에서 언급한 평가용 이미지 119장에 대해 평가하였고, 결괏값의 평균을 구하여 표 1에 정리하였다. 가장 좋은 성능을 나타내는 항목은 붉은색, 상위 3개의 값은 파란색으로 표시하였다. 수행시간은 젓슨 나노 보드에서 초해상화 알고리즘을 실행하였을 때 제일 적게 걸리는 시간이 가장 좋음을 뜻한다.

그림 1은 SRCNN, FSRCNN, ESPCN 에서 주로 사용되는 5개 함수에 대한 PSNR 값을 방사형 그래프로 나타낸 것이다. FSRCNN과 ESPCN 에서는 PReLU가 27.35dB, 27.49dB로 가장 높은 PSNR 값을 보였고, SRCNN에서는 가장 좋은 수치인 Tanh의 28.67dB보다 0.7% 낮음을 보였다.

다음은 인간의 시각적 차이를 이용한 지표인 SSIM을 측정하여 그림 2에 나타내었다. FSRCNN과 ESPCN 에서는 PReLU가 0.8214, 0.8256으로 가장 원본과 유사성을 띄었고, SRCNN에서 가장 높은 값 대비 0.9% 낮음을 보였다. 마지막으로 그림 3은 젓슨 나노 보드에서 알고리즘 수행시간을 측정하여 그래프로 나타낸 것이다. SRCNN, FSRCNN, ESPCN 에서 PReLU가 각각 가장 적은 시간이 걸린 함수 대비 1.1%, 1.0%, 1.0% 오래걸림을 확인하였다.

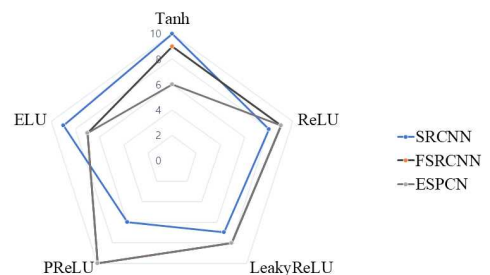


그림 1. SRCNN, FSRCNN, ESPCN 에서의 상위 5개 함수에 대한 PSNR 결괏값 비교

결과적으로 보았을 때 PReLU함수가 PSNR, SSIM에서 골고루 좋은 성능을 보였고, 수행시간 측면에서도 다른 활성 함수 대비 상대적으로 빠른 수행시간을 가지고 있어 가장 적합함을 확인하였다.

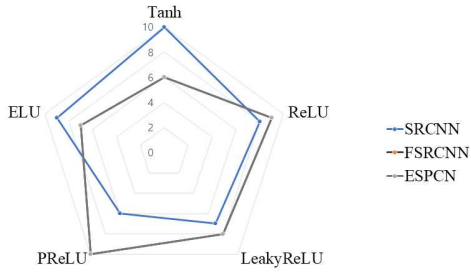


그림 2. SRCNN, FSRCNN, ESPCN 에서의 상위 5개 함수에 대한 SSIM 결괏값 비교

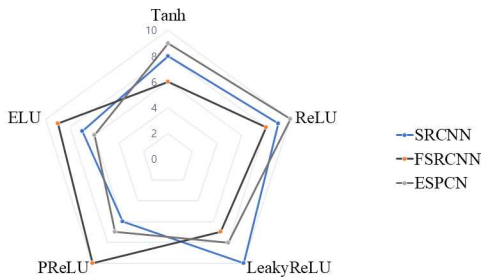


그림 3. SRCNN, FSRCNN, ESPCN 에서의 상위 5개 함수에 대해 걸리는 시간 비교

4. 결론

본 논문에서는 고해상도 영상이 중요시됨에 따라 CNN 기반의 초해상화 기법인 SRCNN, FSRCNN, ESPCN을 검토해 보았고, 이 알고리즘을 웹 시스템으로 구현하였다. 젯슨 나노 보드에서 이미지의 화질 개선 성능이 가장 뛰어난 활성화 함수를 찾기 위해 91개의 훈련 이미지 셋을 바탕으로 Set5, Set14, Urban100에 대해 평가하였다. 활성화 함수로는 Tanh, ReLU, LeakyReLU, PReLU, ELU, GELU, Swish, Mish, Hardswish를 사용하였고, 평가 데이터 셋 총 119장에 대해 PSNR, SSIM, 수행시간을 평가하여 평균값을 도출하였다. 그 결과 젯슨 나노 보드에서 가장 최적화된 함수 PReLU를 찾을 수 있었다. 실행 시간이 오래 걸리는 문제점이 남아있으므로 본 연구를 확장하여 경량화를 통해 젯슨 나노 보드에서 빠른 수행시간과 높은 화질 성능을 가지는 네트워크 구조를 기대할 수 있다.

감사의 글

다음의 성과는 과학기술정보통신부와 연구개발특구진흥재단이 지원하는 과학벨트 지원사업으로 수행된 연구결과입니다.

참고문헌

- [1] MOIS, "2020 Statistical Yearbook(2019.12.31.)", 2021 Statistical Yearbook, pp. 157, August 2021.
- [2] YTN Science, 2015, Noh Woongrae, "It's hard to identify faces with low resolution, half of local governments' CCTVs.", Accessed: Sep. 8, 2015.
- [3] Dong, Chao, et al. , "Image super-resolution using deep convolutional networks.", IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 2, pp. 295-307, February 2015.
- [4] Dong, Chao, Chen Change Loy, and Xiaoou Tang, "Accelerating the super-resolution convolutional neural network.", European conference on computer vision, Springer, Cham, pp. 391-407, October 2016.
- [5] Shi, Wenzhe, et al. , "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network.", Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1874-1883, 2016.
- [6] Ramachandran, Prajit, Barret Zoph, and Quoc V. Le. , "Searching for activation functions.", arXiv preprint arXiv:1710.05941, 2017.
- [7] Misra, Diganta. , "Mish: A self regularized non-monotonic neural activation function.", arXiv preprint arXiv:1908.08681 4, vol. 4, no. 2, 2019.
- [8] Howard, Andrew, et al. , "Searching for mobilenetv3." , Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1314-1324, 2019.
- [9] Timofte, R., De Smet, V., & Van Gool, L. , "Anchored neighborhood regression for fast example-based super-resolution", IEEE international conference on computer vision, pp. 1920-1927, 2013
- [10] Wang, Zhou, et al. , "Image quality assessment: from error visibility to structural similarity.", IEEE transactions on image processing, vol. 13, no. 4, pp. 600-612, April 2004.