



# CNN을 활용한 Super Resolution Platform

저해상도를 고해상도로

졸업과제 최종발표

지도교수 : 차의영

참여학생 : 김근식, 박형탁, 황남기

팀번호 : C35



# 목차

---

## 1. 과제 개요

- 1.1. 문제 분석
- 1.2. 과제 소개
- 1.3. 시스템 흐름도

## 2. 상세 기술

- 2.1. SRCNN (Super-Resolution Using Deep Convolutional Networks)
- 2.2. VDSR (Super-Resolution Using Very Deep Convolutional Networks)
  - Residual Learning, Adjustable Gradient Clipping

## 3. 수행 내용

## 4. 결론

- 4.1. 기대효과 및 활용방안
  - 4.2. 시연 동영상
-

# 과제 개요



# 과제 개요

## 1.1. 문제 분석

과제 개요



지난해 5월, 지상파 방송사가 초고화질(4K UHD) 본방송 시대를 선언했으나, 실제로 이용할 수 있는 콘텐츠는 아직도 많지 않은 상황

이제 막 출시되기 시작한 8K TV에 맞는 영상은 사실상 찾아보기 힘들다

따라서 디스플레이 기술의 비약적인 발전과는 달리 제한된 고품질 콘텐츠 문제를 해결하기 위해 화질을 자동으로 높이는 인공지능 알고리즘이 필요한 시점이다.

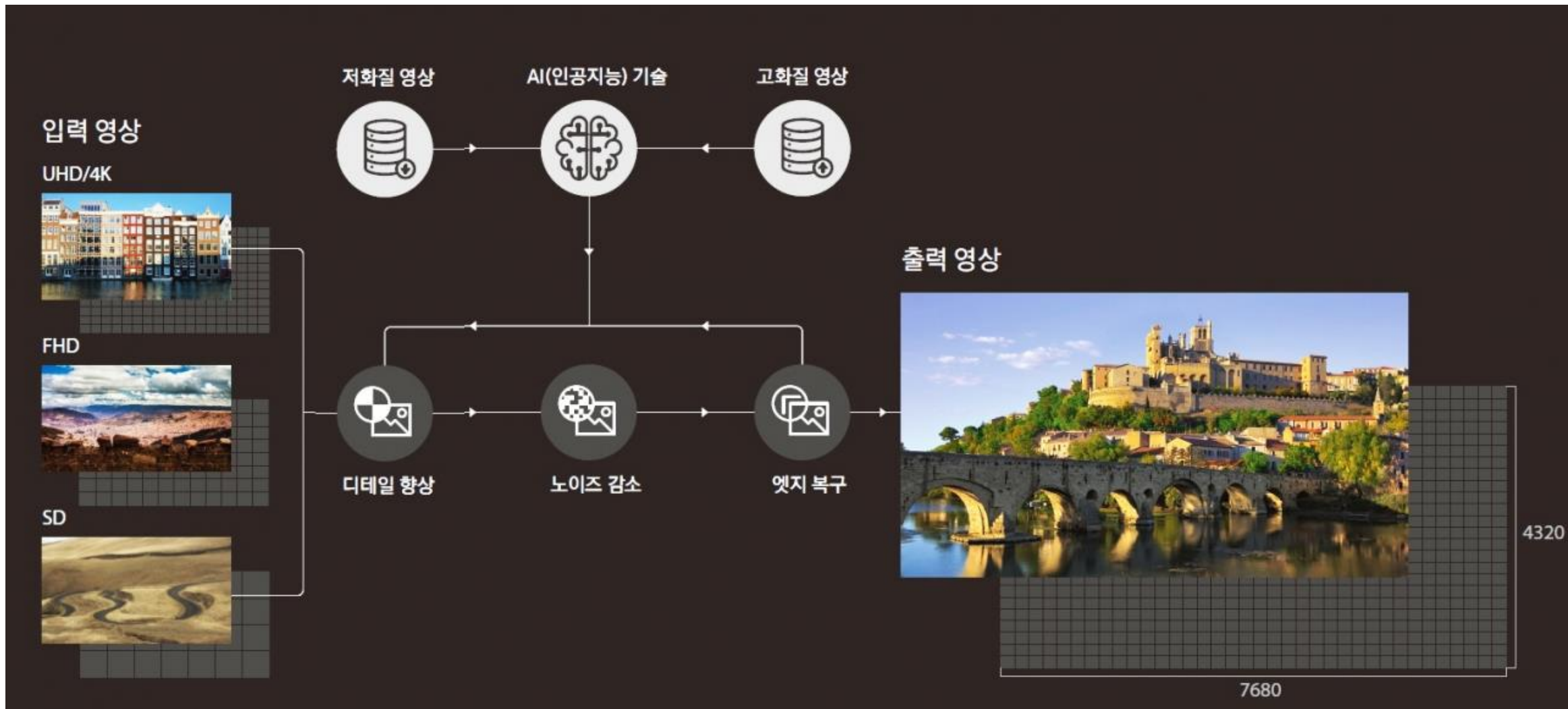


## 과제 개요

### 1.2. 과제 소개

과제 개요

# Super Resolution이란?





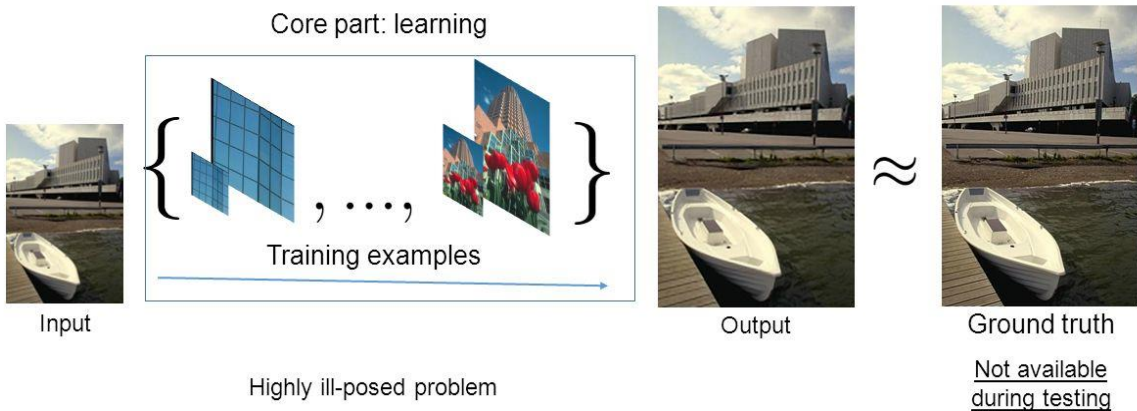
## 과제 개요

### 1.2. 과제 소개

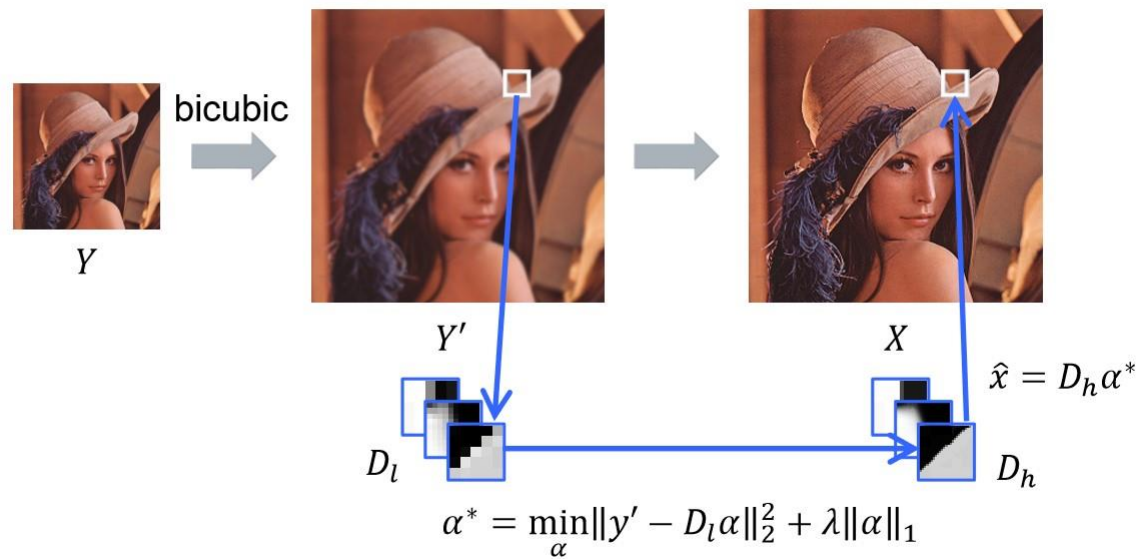
과제 개요

# SISR 관련 선행 연구들

## Example-based approaches



## Sparse Coding based Super-Resolution [J. Yang, et al, 2010]



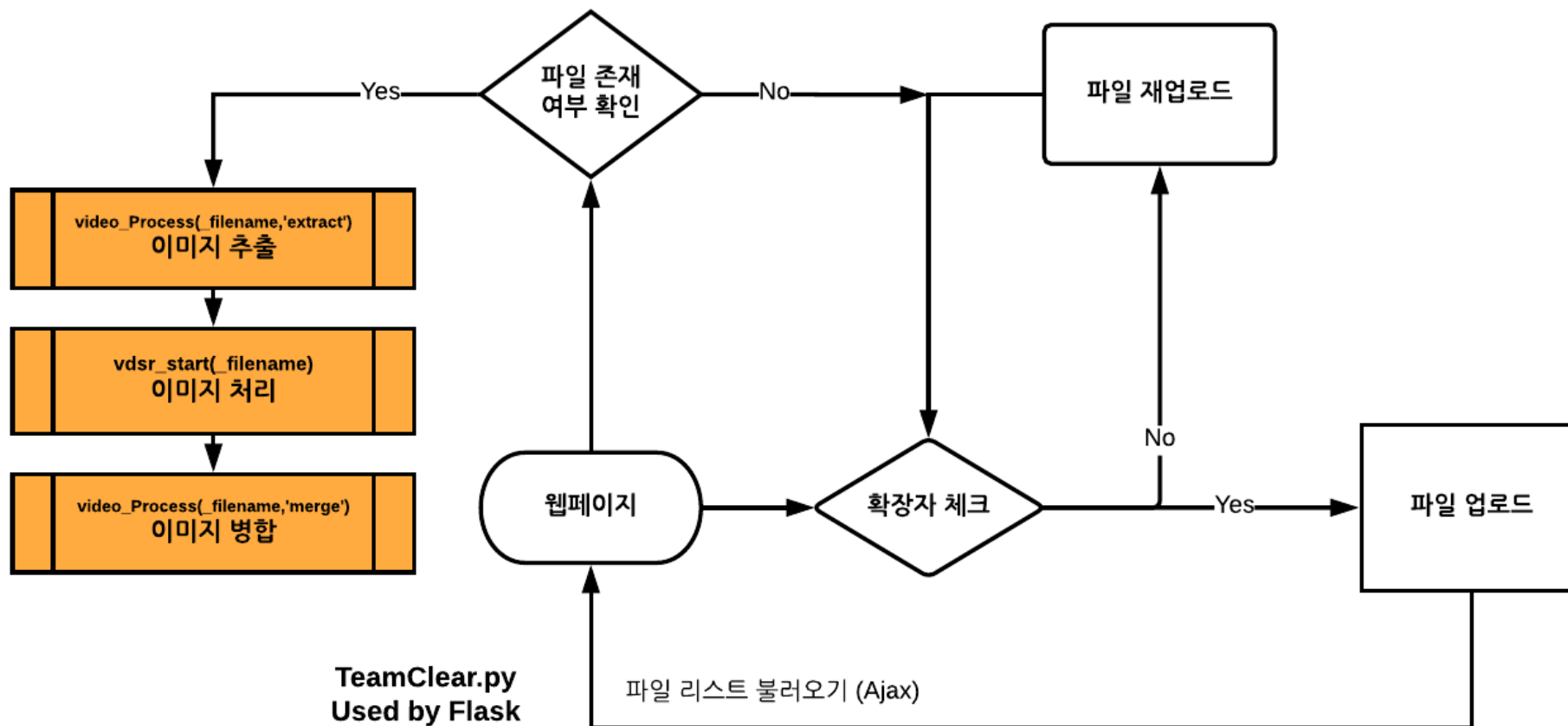


# 과제 개요

## 1.3. 시스템 흐름도

상세 기술

### CNN을 활용한 Super Resolution Platform



**상세 기술**

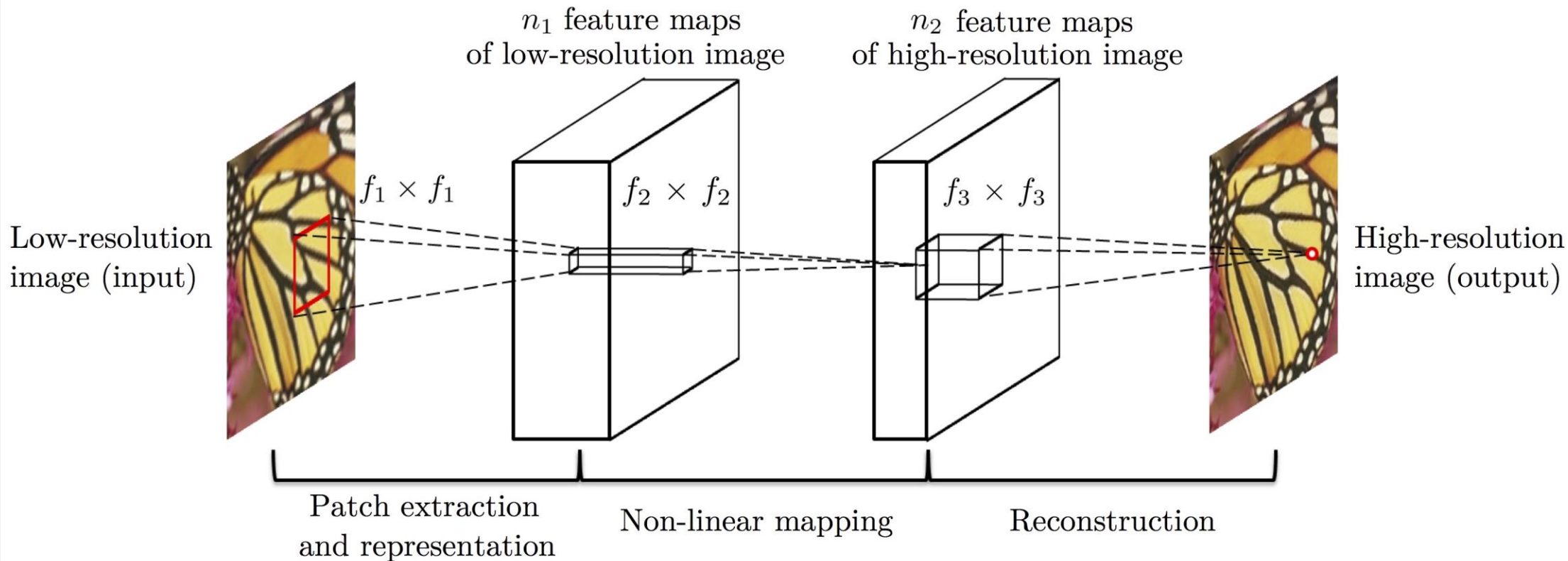




## 상세 기술

### 2.1. SRCNN (Super-Resolution Using Deep Convolutional Networks)

상세 기술



Chao Dong 연구팀은 Super Resolution 문제에 있어서 최초로 딥러닝 적용

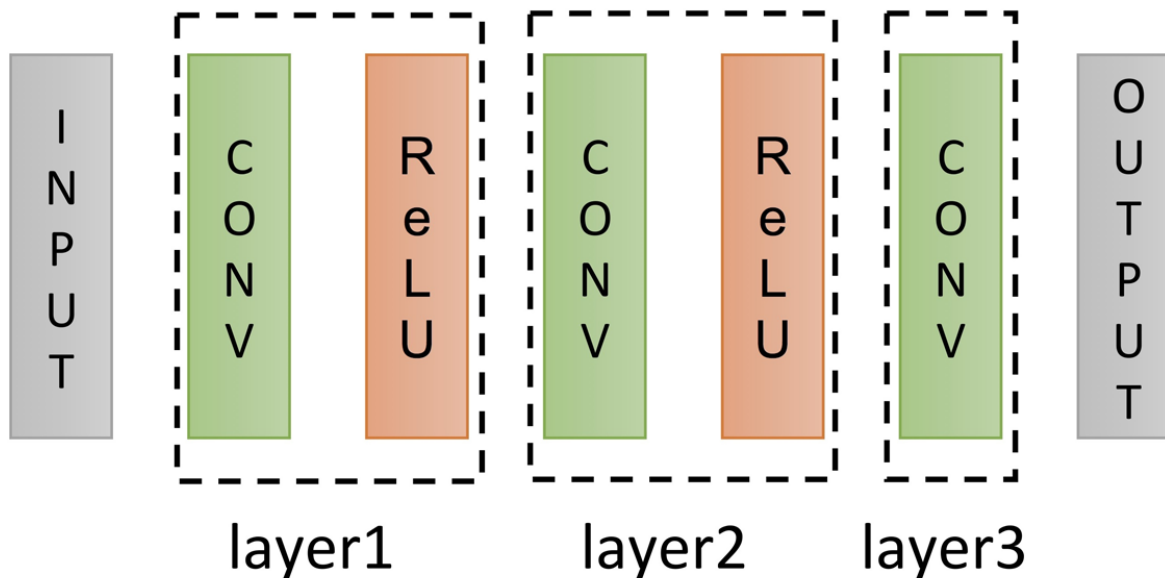
Super Resolution 문제에 있어서 이 모델이 기준이 되어 조금씩 바뀌어 나간다



# 상세 기술

## 2.1. SRCNN (Super-Resolution Using Deep Convolutional Networks)

상세 기술



$$F_1(\mathbf{Y}) = \max(0, W_1 * \mathbf{Y} + B_1)$$

$$F_2(\mathbf{Y}) = \max(0, W_2 * F_1(\mathbf{Y}) + B_2)$$

$$F(\mathbf{Y}) = W_3 * F_2(\mathbf{Y}) + B_3$$

$$L(\Theta) = \frac{1}{n} \sum_{i=1}^n \|F(\mathbf{Y}_i; \Theta) - \mathbf{X}_i\|^2$$

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX_I^2}{MSE} \right)$$



## 상세 기술

### 2.1. SRCNN (Super-Resolution Using Deep Convolutional Networks)

## 한계

1. 작은 이미지 영역의 상황에 의존
2. 학습이 너무 느림
3. 네트워크 단일 규모로만 작동(single scale)



상세 기술





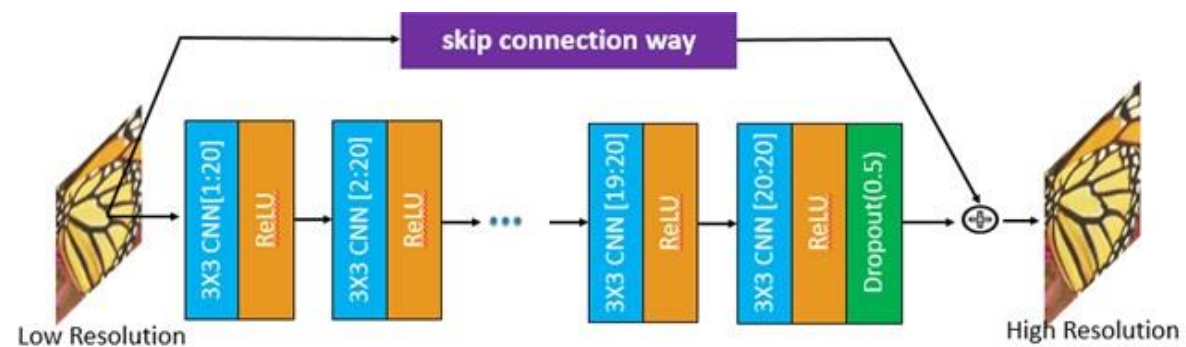
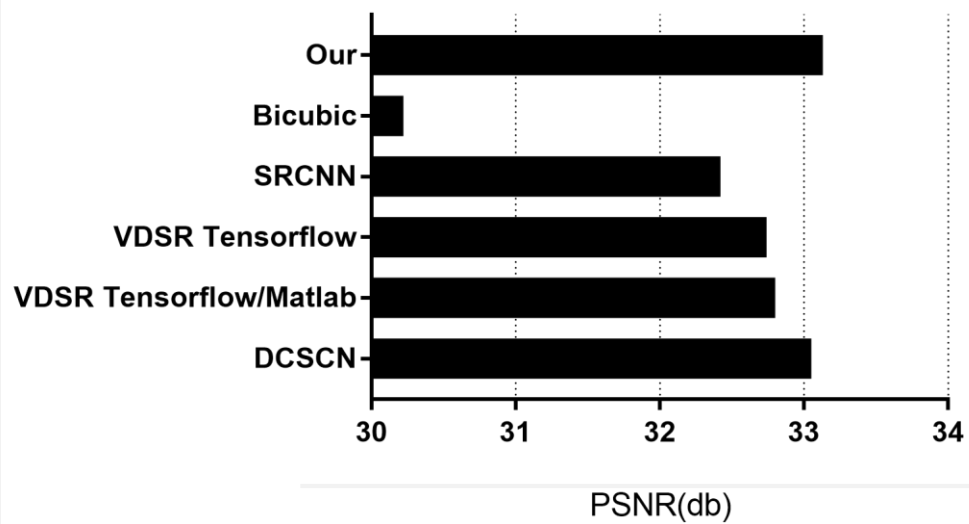


# 상세 기술

## 2.2. VDSR

### 과제 개요

Benchmark Result

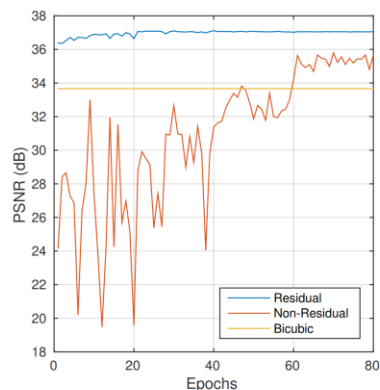




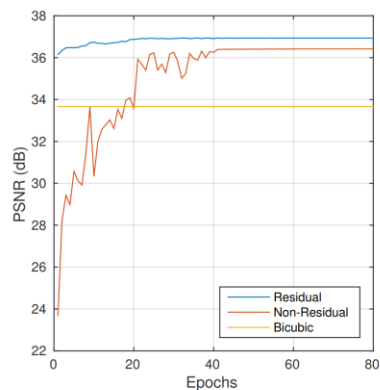
# 상세 기술

## 2.2. VDSR - Residual Learning

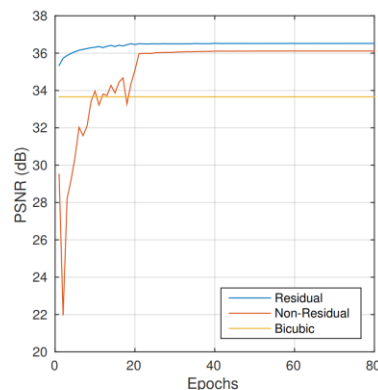
### 과제 개요



(a) Initial learning rate 0.1



(b) Initial learning rate 0.01



(c) Initial learning rate 0.001

Epoch	10	20	40	80
Residual	36.90	36.64	37.12	37.05
Non-Residual	27.42	19.59	31.38	35.66
Difference	9.48	17.05	5.74	1.39

(a) Initial learning rate 0.1

Epoch	10	20	40	80
Residual	36.74	36.87	36.91	36.93
Non-Residual	30.33	33.59	36.26	36.42
Difference	6.41	3.28	0.65	0.52

(b) Initial learning rate 0.01

Epoch	10	20	40	80
Residual	36.31	36.46	36.52	36.52
Non-Residual	33.97	35.08	36.11	36.11
Difference	2.35	1.38	0.42	0.40

(c) Initial learning rate 0.001

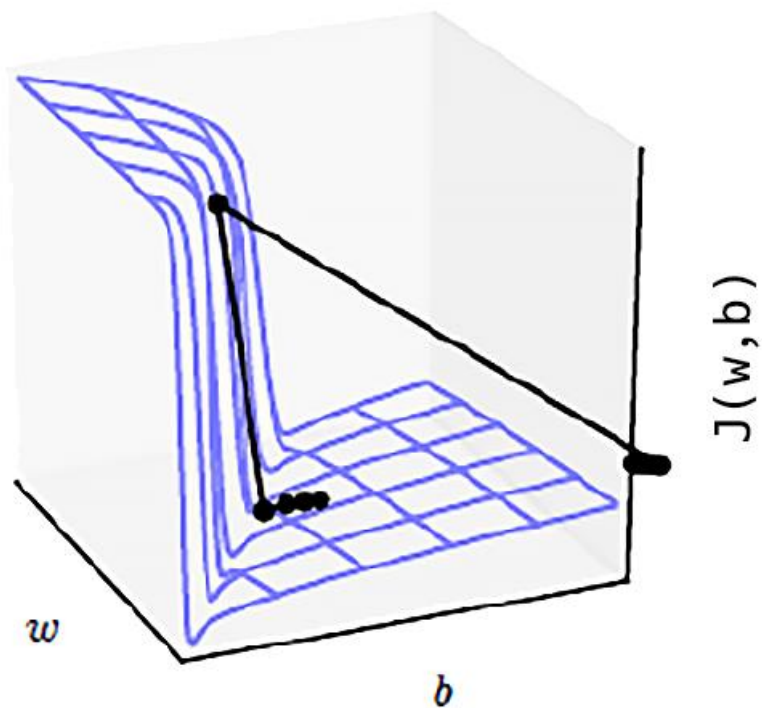


# 상세 기술

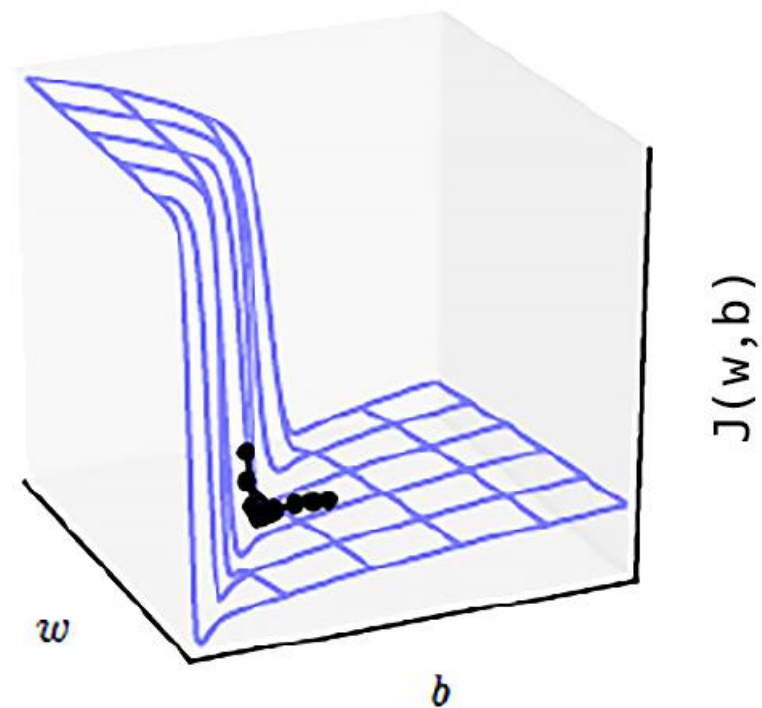
## 2.2. VDSR - Gradient Clipping

과제 개요

Without clipping



With clipping





## 상세 기술

### 2.2. VDSR - Multi Scale

과제 개요

Test / Train	×2	×3	×4	×2,3	×2,4	×3,4	×2,3,4	Bicubic
×2	37.10	30.05	28.13	37.09	37.03	32.43	37.06	33.66
×3	30.42	32.89	30.50	33.22	31.20	33.24	33.27	30.39
×4	28.43	28.73	30.84	28.70	30.86	30.94	30.95	28.42

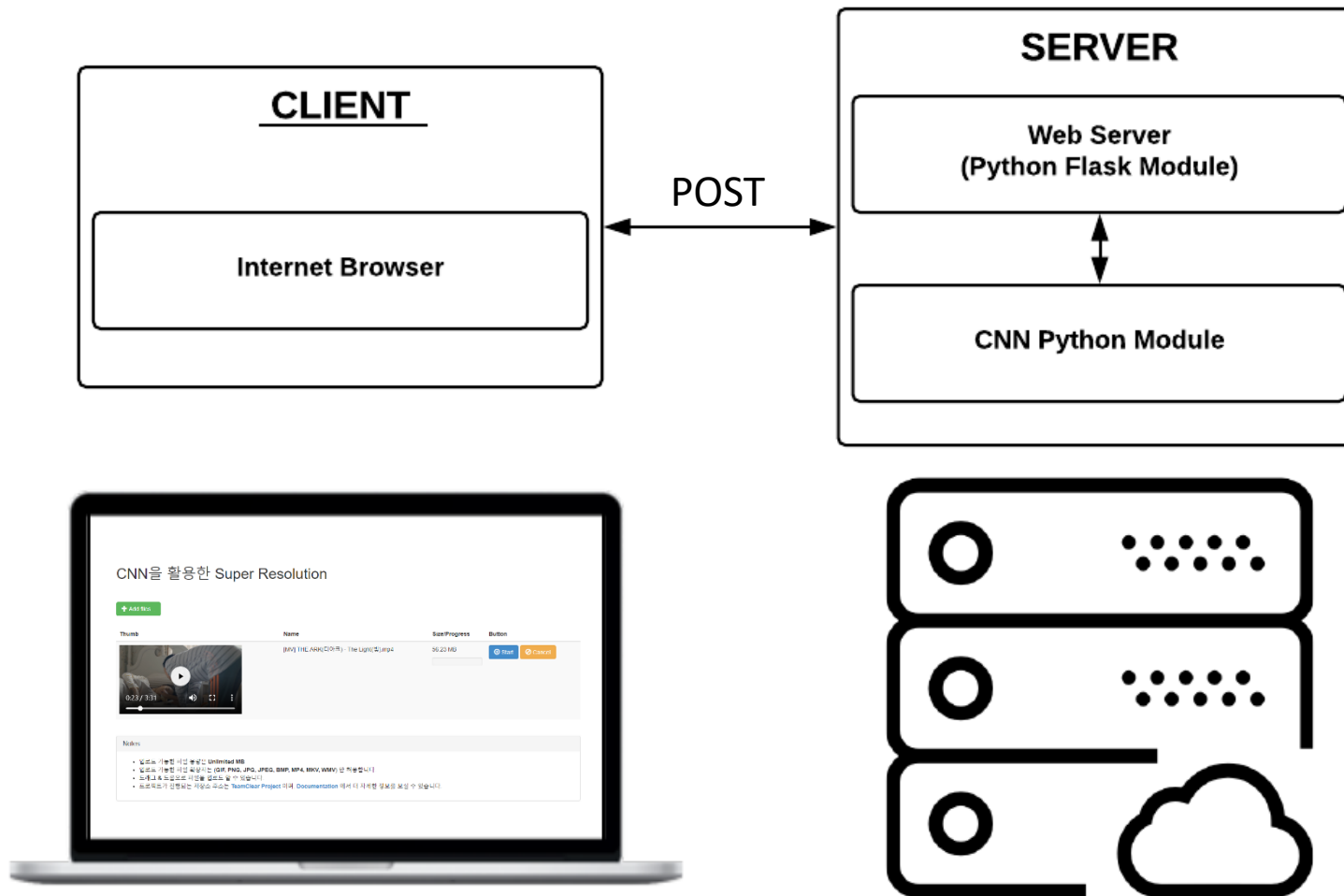


# 수행 개요



# 수행 환경 (Client-Server 구조)

수행 개요





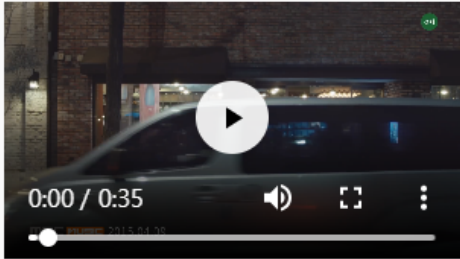
# 수행 환경 (Client-Server 구조)

- 메인 인터페이스 화면

수행 개요

## CNN을 활용한 Super Resolution

[+ Add files...](#) [Start upload](#) [Cancel upload](#) [Delete](#) ☐

Thumb	Name	Size/Progress	Button
	[MV] THE ARK(디아크) - The Light(빛) 30sec.mp4	8.98 MB <div></div>	<a href="#">Start</a> <a href="#">Cancel</a>

### Notes

- 업로드 가능한 파일 용량은 **Unlimited MB**
- 업로드 가능한 파일 확장자는 (GIF, PNG, JPG, JPEG, BMP, MP4, MKV, WMV) 만 허용합니다.
- 드래그 & 드롭으로 파일을 업로드 할 수 있습니다.
- 프로젝트가 진행되는 저장소 주소는 이며, [Documentation](#)에서 더 자세한 정보를 보실 수 있습니다.



# 수행 환경 (Client-Server 구조)

- 업로드 된 화면

수행 개요

## CNN을 활용한 Super Resolution

+ Add files...

Thumb	Name	Size/Progress	Button
	[MV] THE ARK(디아크) - The Light(빛).mp4	56.23 MB	<a href="#">▶ PLAY</a> <a href="#">▶ CONV</a> <a href="#">🗑 Delete</a>

### Notes

- 업로드 가능한 파일 용량은 **Unlimited MB**
- 업로드 가능한 파일 확장자는 (GIF, PNG, JPG, JPEG, BMP, MP4, MKV, WMV) 만 허용합니다.
- 드래그 & 드롭으로 파일을 업로드 할 수 있습니다.
- 프로젝트가 진행되는 저장소 주소는 [TeamClear Project](#) 이며, [Documentation](#) 에서 더 자세한 정보를 보실 수 있습니다.

### Convert Button

Info

Cut (30 sec)

Extract

Merge

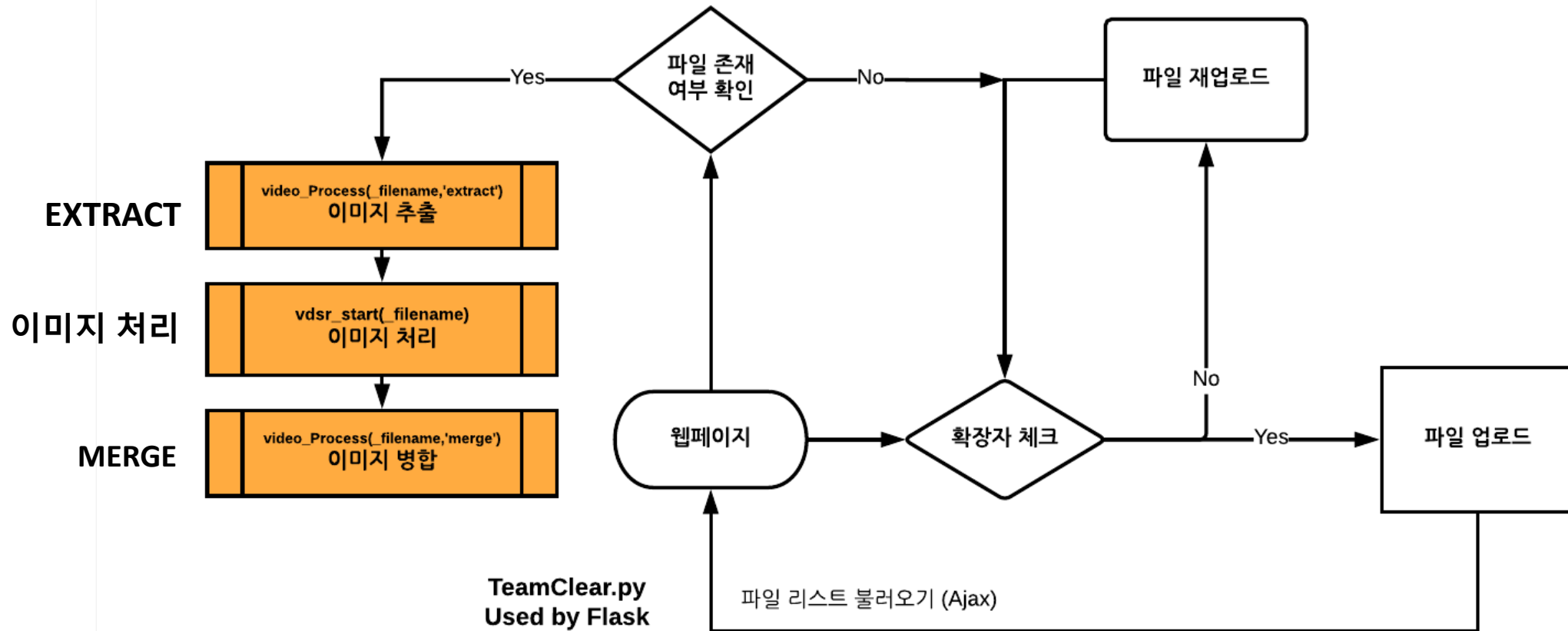


# 수행 환경 (Client-Server 구조)

## - 흐름도

수행 개요

### CNN을 활용한 Super Resolution Platform





## 수행 환경 (학습 서버)

- Google Cloud, VM Instance

수행 개요

### 개발 환경

Google Cloud

Ubuntu 16.04

Tensorflow 1.8

Cuda 9.2

### 개발 언어

Python 3.x

HTML

CSS

### 개발 자원

4 CPU

15GB RAM

Nvidia Tesla K80

**\*18EA**

### 알고리즘

VDSR

Residual Learning

Gradient Clipping



## 수행 환경 (학습 서버)

### 수행 개요

[epoch 2] loss 2720542.2500	lr 0.00010000	2714.337 sec	[epoch 21] loss 2846843.5000	lr 0.00000100	19859.143 sec
[epoch 3] loss 3233628.5000	lr 0.00010000	3616.416 sec	[epoch 22] loss 2867921.2500	lr 0.00000100	20761.933 sec
[epoch 4] loss 2951822.5000	lr 0.00010000	4518.526 sec	[epoch 23] loss 2378737.0000	lr 0.00000100	21664.509 sec
[epoch 5] loss 3419303.2500	lr 0.00010000	5420.835 sec	[epoch 24] loss 2882099.0000	lr 0.00000100	22566.840 sec
[epoch 6] loss 2889710.7500	lr 0.00010000	6323.232 sec	[epoch 25] loss 2712489.7500	lr 0.00000100	23469.550 sec
[epoch 7] loss 3062536.5000	lr 0.00010000	7225.435 sec	[epoch 26] loss 2885852.5000	lr 0.00000100	24372.078 sec
[epoch 8] loss 3415028.5000	lr 0.00010000	8128.015 sec	[epoch 27] loss 2902103.2500	lr 0.00000100	25274.564 sec
[epoch 9] loss 3282760.2500	lr 0.00010000	9030.389 sec	[epoch 28] loss 2632343.2500	lr 0.00000100	26177.013 sec
[epoch 10] loss 3180505.2500	lr 0.00001000	9932.878 sec	[epoch 29] loss 3210632.7500	lr 0.00000100	27079.659 sec
[epoch 11] loss 3206081.5000	lr 0.00001000	10835.923 sec	[epoch 30] loss 2728325.7500	lr 0.00000010	27982.100 sec
[epoch 12] loss 2609696.2500	lr 0.00001000	11738.602 sec	[epoch 31] loss 2713013.7500	lr 0.00000010	28885.299 sec
[epoch 13] loss 2762004.0000	lr 0.00001000	12641.018 sec	[epoch 32] loss 2649125.0000	lr 0.00000010	29788.382 sec
[epoch 14] loss 2273351.2500	lr 0.00001000	13543.344 sec	[epoch 33] loss 2728728.2500	lr 0.00000010	30691.451 sec
[epoch 15] loss 3151433.2500	lr 0.00001000	14445.708 sec	[epoch 34] loss 2855189.5000	lr 0.00000010	31594.367 sec
[epoch 16] loss 2729447.2500	lr 0.00001000	15347.753 sec	[epoch 35] loss 2990909.5000	lr 0.00000010	32497.408 sec
[epoch 17] loss 3006702.5000	lr 0.00001000	16249.564 sec	[epoch 36] loss 2379701.7500	lr 0.00000010	33400.408 sec
[epoch 18] loss 3291885.7500	lr 0.00001000	17151.766 sec	[epoch 37] loss 2414987.5000	lr 0.00000010	34303.392 sec
[epoch 19] loss 3281442.0000	lr 0.00001000	18053.949 sec	[epoch 38] loss 2920313.5000	lr 0.00000010	35206.402 sec
[epoch 20] loss 2393501.7500	lr 0.00000100	18956.440 sec	[epoch 39] loss 2683978.2500	lr 0.00000010	36109.580 sec

Epoch 40, Conv 10 설정  
테스트 데이터 : 약 28만 개  
시간 약 36110초 ( = 10시간)

Conv 20 → Conv 10 (튜닝을 통해 성능을 최적화)



# 수행 환경 (학습 데이터)

수행 개요

Test / Train Data Set	Name	Image Source
Test	Set 5	Bevilacqua et al. BMVC2012
Test	Set 14	Zeyde et al. LNCS 2010
Test	BSD 100	Martin et al. ICCV 2001
Test	Urban 100	Huang et al. CVPR 2015
Train	Yang 91	Yang et al. CVPR 2008
Train	291	Yang 91 + Berkeley Segmentation Dataset 200

- Deep learning 이전 / 'Computer Vision', 'Image Processing' 기법 / Super Resolution
- 벤치마킹과 정확성을 위해 기존 Data Set 사용





# 수행 환경 (학습 데이터)

- data augmentation 기법

수행 개요



〈원본〉



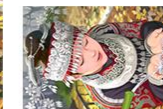
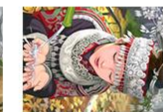
〈Rotate 90〉



〈Rotate 180〉



〈Rotate 270〉



〈Flip〉



〈x2, x3, x4 Data Resize〉



〈Size 복구, 저해상도 이미지 생성〉  
→ 원본과 비교하여 학습을 위해



++Data

정확도 향상 ↑



## 성능 비교 (PSNR)

- 학습 모델 평가

수행 개요



PSNR = 40dB



PSNR = 30dB



PSNR = 20dB



PSNR = 10dB



PSNR = 0dB

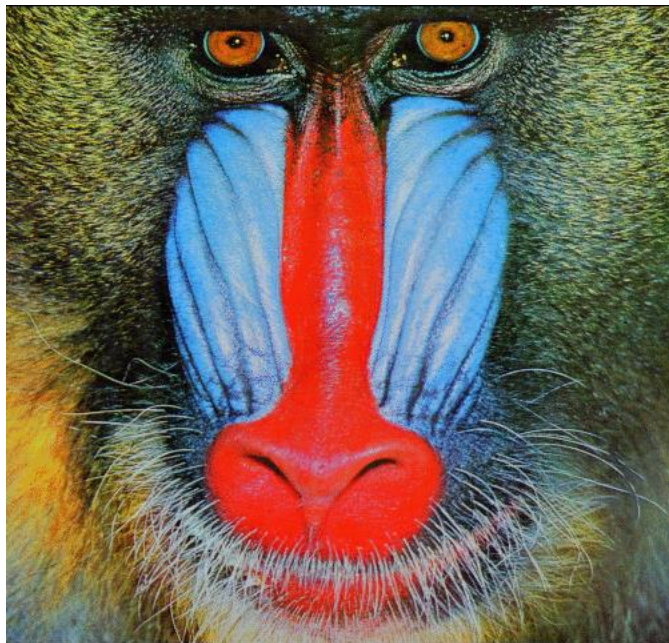
$$PSNR = 10 \cdot \log_{10} \left( \frac{\text{신호전력}}{\text{잡음전력}} \right)$$



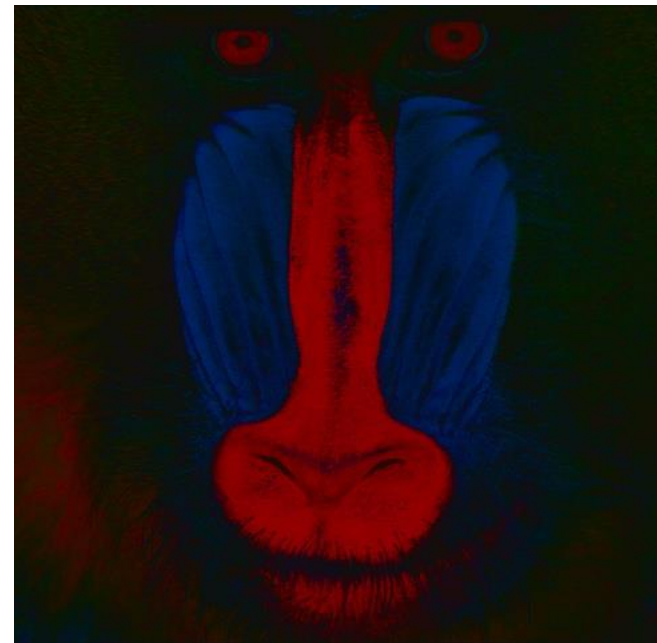


## 첫 테스트 결과

수행 개요



< Original >



< PSNR 7 >

- 이론 부족
- 첫 테스트한 결과 'PSNR 7'
- 성능 최적화, 수치 튜닝



# 수행 환경 (학습 데이터)

- Learning Rate 학습 데이터 정리

수행 개요

## LearningRate

깊이	10
미니배치	64
매개변수갱신	Adam
가중치감소	0.0001
초기화방법	Xavier
활성화함수	Relu
에폭	40

PSNR									
학습율	0.01	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.9
X2	32.825	32.859	32.847	32.859	32.818	32.819	32.851	32.824	32.819
X3	29.645	29.649	29.640	29.657	29.645	29.642	29.647	29.645	29.650
X4	27.801	27.819	27.808	27.820	27.814	27.800	27.794	27.817	27.783

Residual Learning → PSNR 값 차이가 별로 없음



# 수행 환경 (학습 데이터)

- Activation 학습 데이터 정리

수행 개요

## Activation

깊이	10
미니배치	64
매개변수갱신	Adam
가중치감소	0.0001
에폭	40
학습률	0.1

Relu 활성화 함수 +  
He 초기화 방법이 가장 우수

PSNR					
초기화방법	Xavier	Xavier	Xavier	He	Xavier
활성화함수	tanh	Maxout	Random_normal	RELU	Sigmoid
X2	30.227	31.329	NaN	32.793	31.767
X3	27.542	27.904	NaN	29.603	28.993
X4	26.001	26.163	NaN	27.796	27.166



# 수행 환경 (학습 데이터)

## - Optimizer 학습 데이터 정리

수행 개요

### Optimizer

깊이	10
미니배치	64
가중치감소	0.0001
초기화방법	Xavier
활성화함수	Relu
에폭	40
학습률	0.1

PSNR				
매개변수갱신	RMSProp	Adagrad	Momentum	Adam
X2	32.799	31.532	19.644	32.859
X3	29.637	28.740	22.164	29.649
X4	27.812	26.852	23.177	27.819



# 수행 환경 (학습 데이터)

## - Initializer 학습 데이터 정리

수행 개요

### Initializer

깊이	10
미니배치	64
매개변수갱신	Adam
가중치감소	0.0001
에폭	40
학습률	0.1

PSNR					
초기화방법	Truncated _normal	He	Uniform _unit_scaling _initializer	Ones _initializer	Orthogonal _initializer
X2	NaN	32.793	32.790	NaN	19.644
X3	NaN	29.603	29.625	NaN	22.164
X4	NaN	27.796	27.814	NaN	23.177

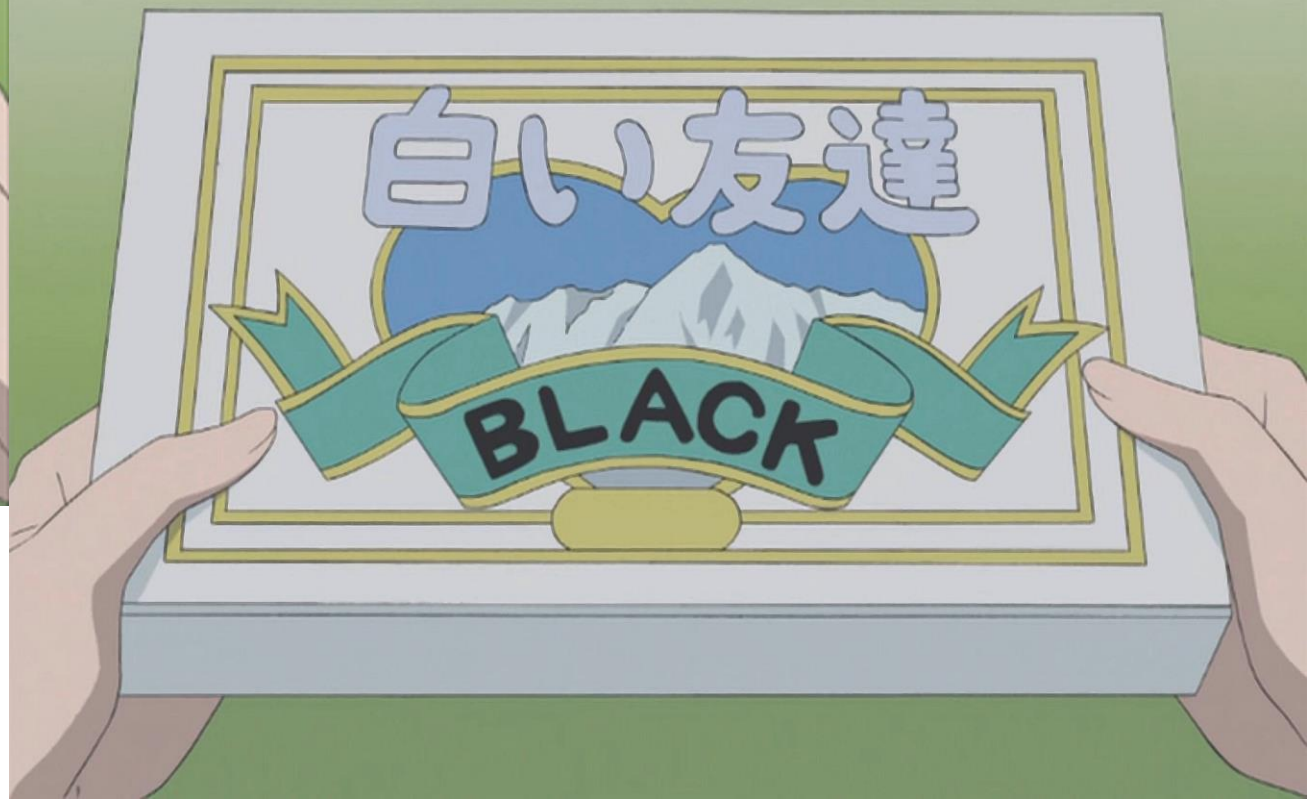


## 최종 결과

Before



After





# 결론



## 기대효과 및 활용방안

### 결론

- 콘텐츠 제작비 절감으로 더욱 양질의 콘텐츠 제작 가능
- 플랫폼화로 인해 접근성이 향상되어 고해상도 콘텐츠 생산
- 저해상도 영상 복원으로 범인 인상착의 파악 등 다양한 분야에서 폭넓은 활용 기대
  - 기존 노후화된 CCTV 카메라 대체 없이 고해상도 영상 취득 가능
  - 수신 받는 TV 해상도 보다 높은 해상도의 영상 변환



결론



# 결론 시연 동영상

## CNN을 활용한 Super Resolution

+ Add files...

Thumb	Name	Size/Progress	Button
<div>Notes</div> <ul style="list-style-type: none"><li>• 업로드 가능한 파일 용량은 <b>Unlimited MB</b></li><li>• 업로드 가능한 파일 확장자는 (<b>GIF, PNG, JPG, JPEG, BMP, MP4, MKV, WMV</b>) 만 허용합니다.</li><li>• 드래그 &amp; 드롭으로 파일을 업로드 할 수 있습니다.</li><li>• 프로젝트가 진행되는 저장소 주소는 <a href="#">TeamClear Project</a> 이며, <a href="#">Documentation</a> 에서 더 자세한 정보를 보실 수 있습니다.</li></ul>			

**감사합니다.**

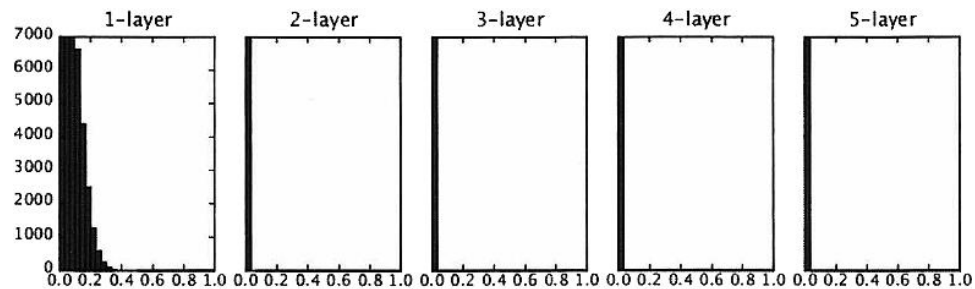
**Q & A**



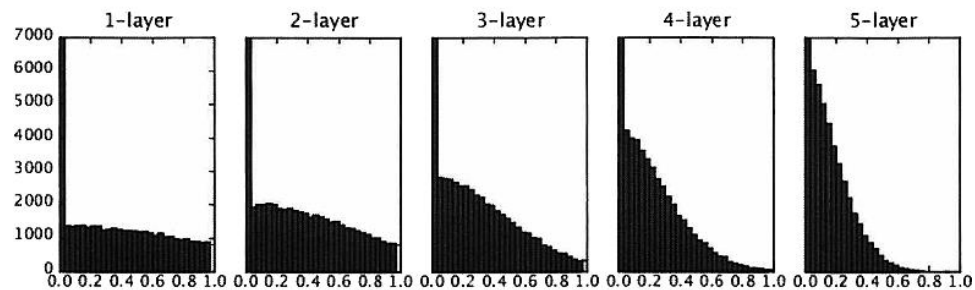
## 초기값 비교 (Xavier, He)

상세 기술

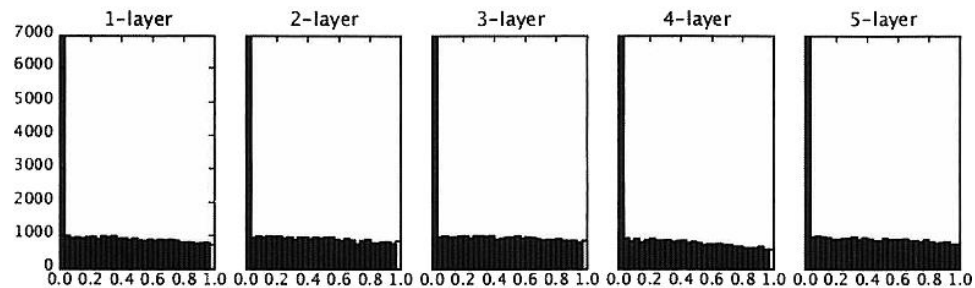
그림 6-14 활성화 함수로 ReLU를 사용한 경우의 가중치 초기값에 따른 활성화값 분포 변화



표준편차가 0.01인 정규분포를 가중치 초기값으로 사용한 경우



Xavier 초기값을 사용한 경우

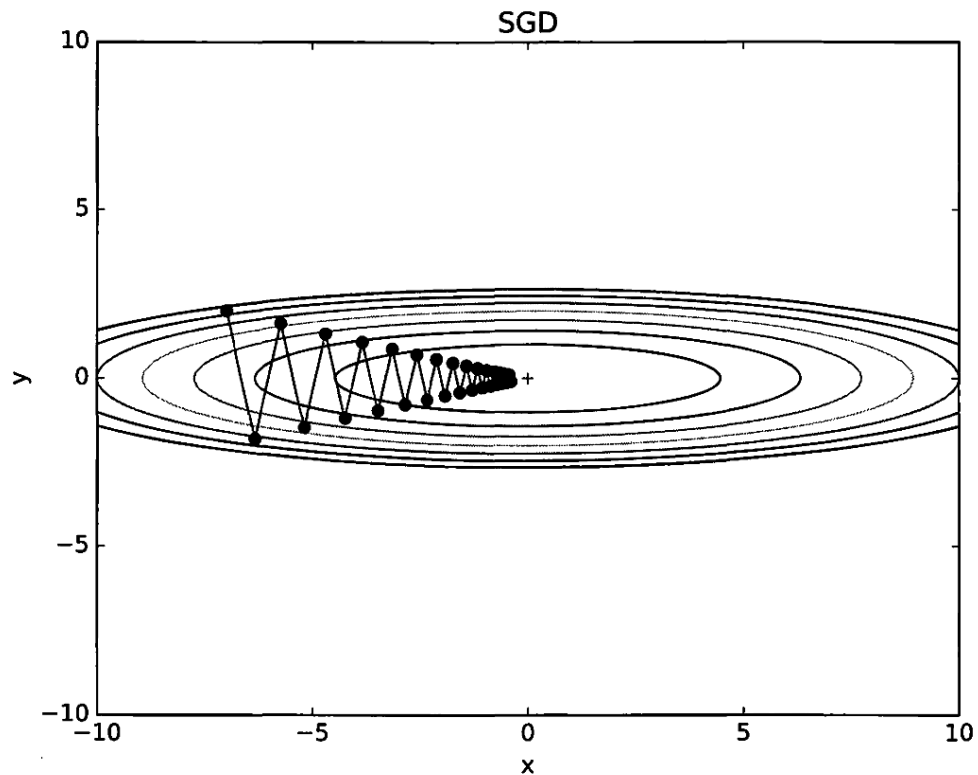




## 매개변수 갱신 방법 비교 (SGD, Momentum)

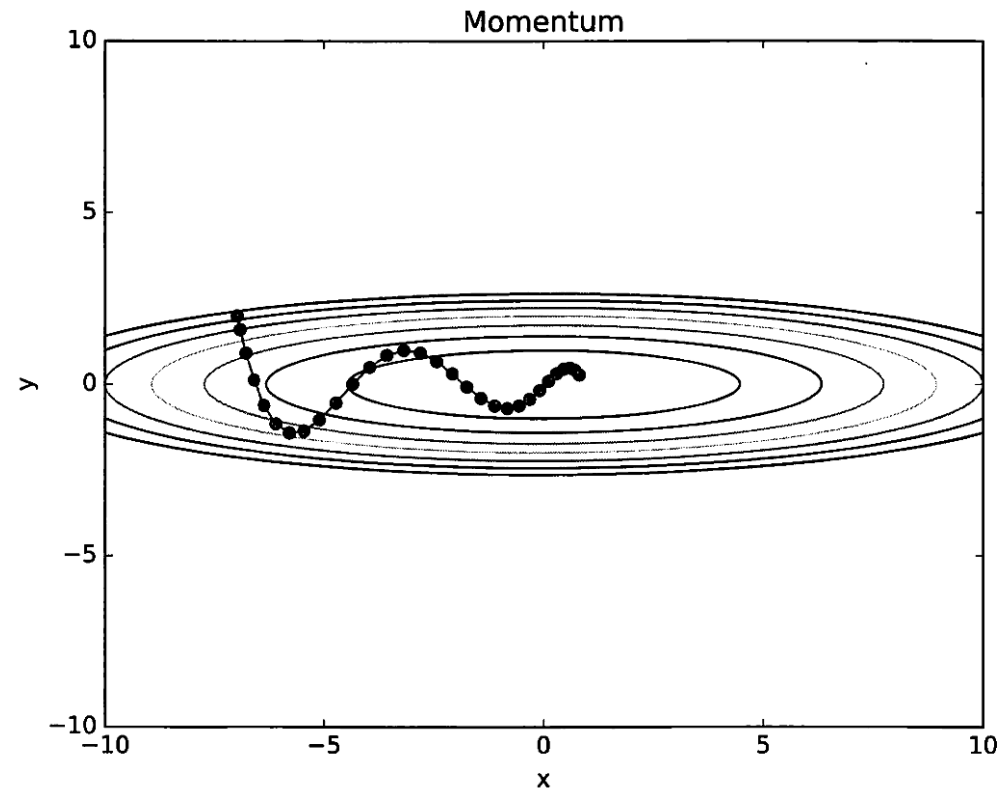
### 확률적 경사 하강법(SGD)

**그림 6-3** SGD에 의한 최적화 갱신 경로 : 최솟값인 (0, 0)까지 지그재그로 이동하니 비효율적이다.



### 모멘텀(Momentum)

**그림 6-5** 모멘텀에 의한 최적화 갱신 경로



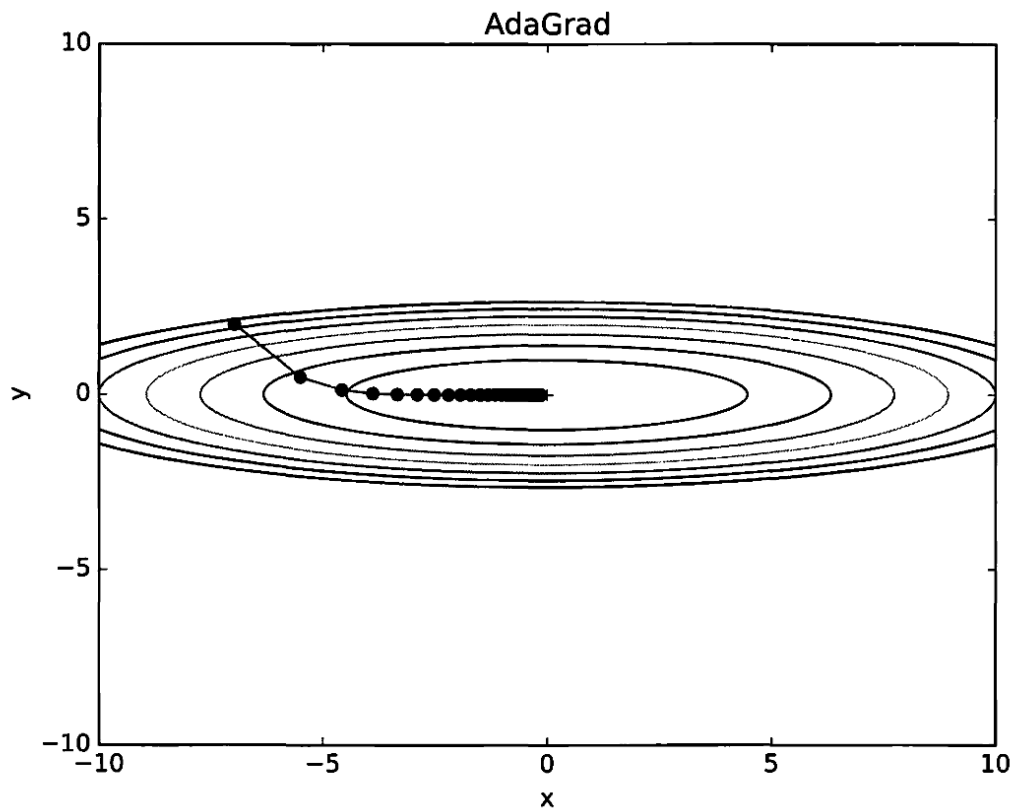


# 매개변수 갱신 방법 비교 (AdaGrad, Adam)

상세 기술

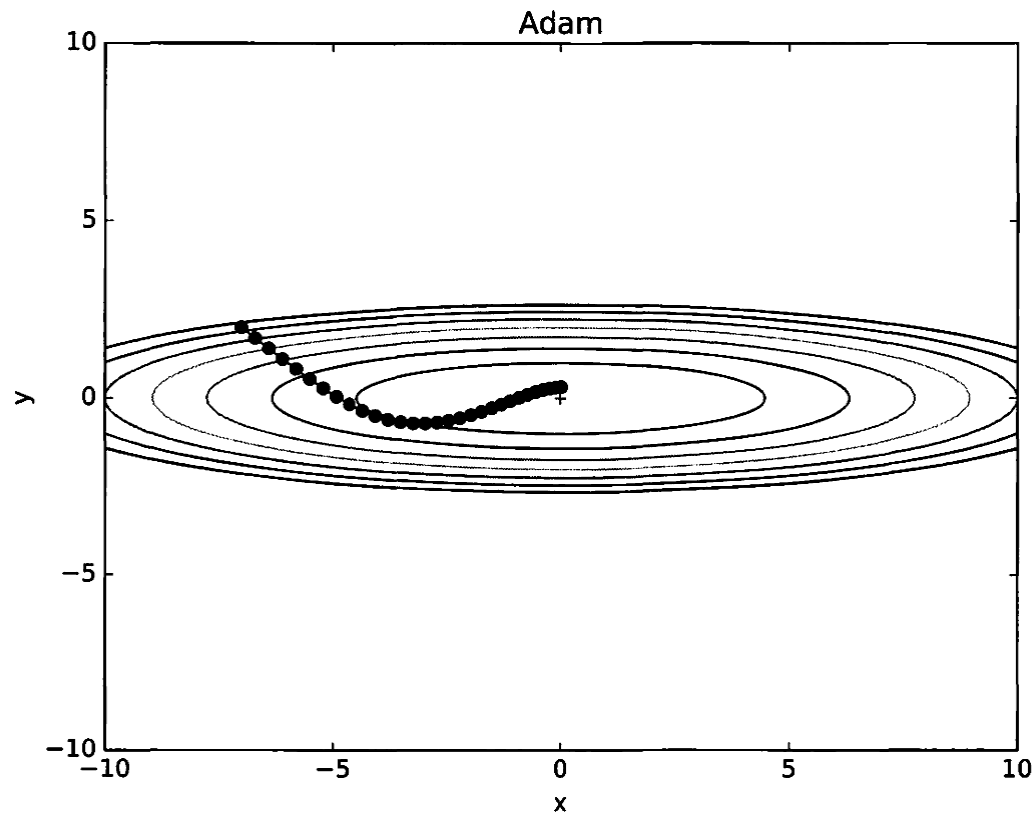
## AdaGrad

그림 6-6 AdaGrad에 의한 최적화 갱신 경로



## Adam

그림 6-7 Adam에 의한 최적화 갱신 경로



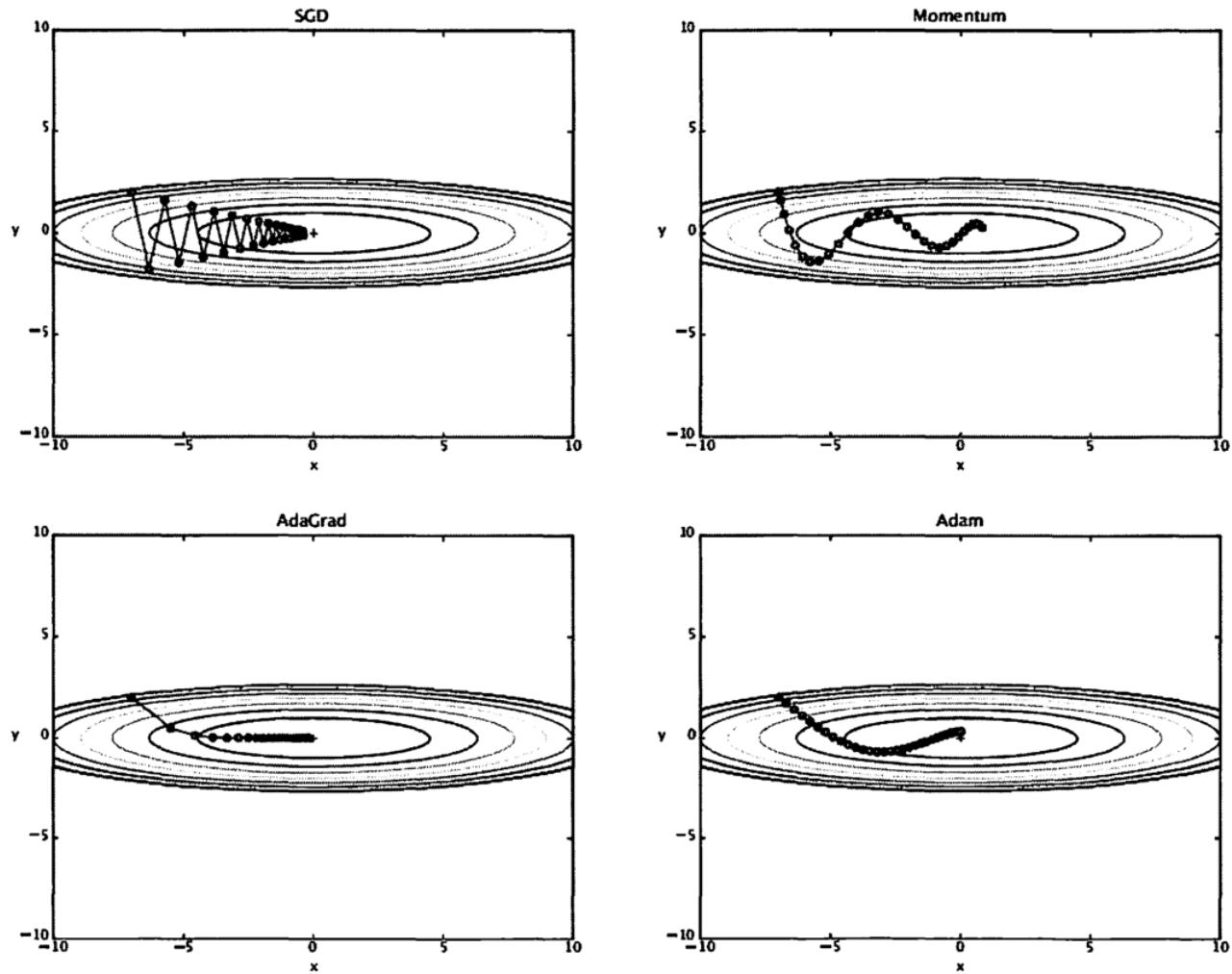




# 매개변수 갱신 방법 비교

상세 기술

그림 6-8 최적화 기법 비교 : SGD, 모멘텀, AdaGrad, Adam





## 오버피팅 개선을 위한 드롭아웃 배치학습 방법

그림 6-16 배치 정규화를 사용한 신경망의 예

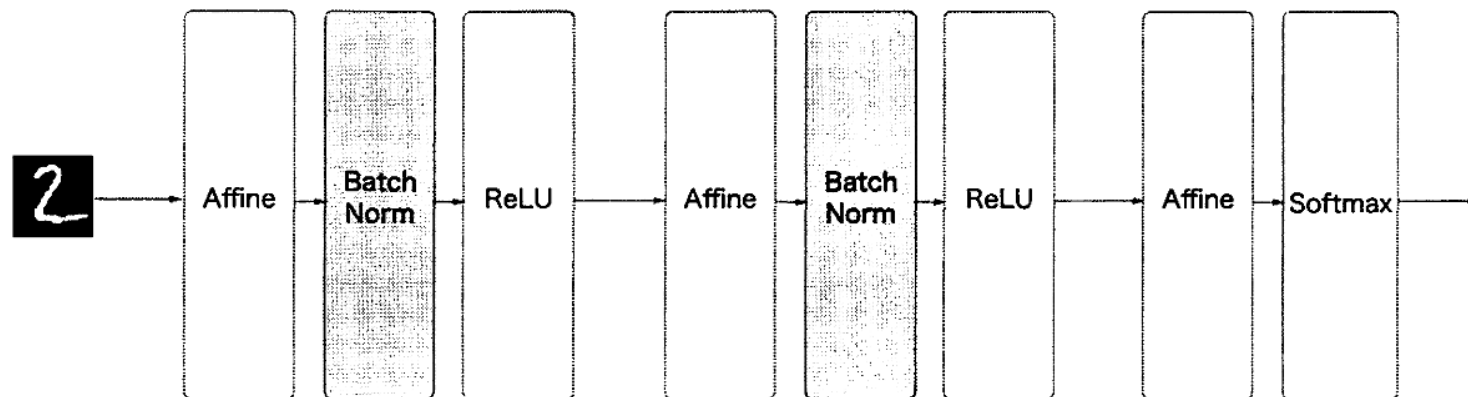
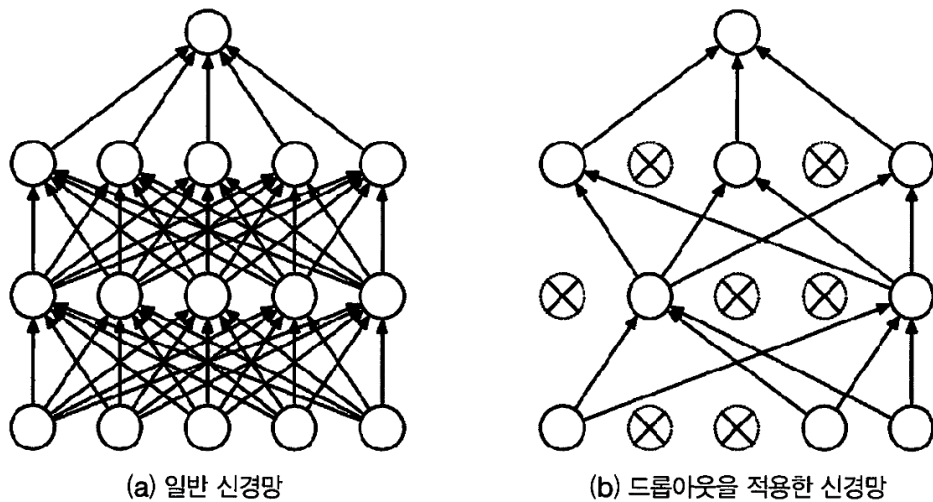


그림 6-22 드롭아웃의 개념(문헌<sup>14)</sup>에서 인용) : 왼쪽이 일반적인 신경망, 오른쪽이 드롭아웃을 적용한 신경망. 드롭아웃은 뉴런을 무작위로 선택해 삭제하여 신호 전달을 차단한다.





---

상세 기술

---

