



FPGA를 이용한 빠르고 지능적인 Visual Cloud Computing

김동규 블루닷

CONTENTS

1. 개발 배경
2. DeepField-SR Deep Dive
 - a. 제품 요구 사항
 - b. Development Flow
 - c. 해상도와 변환 비율
 - d. 네트워크 구조와 학습
 - e. 가속기 설계 - HLS (High-Level Synthesis)
 - f. Implementation
 - g. 테스트
 - h. 성능 지표
3. Demo
4. Moving Forward

개발 배경

블루닷이 하는 일은...

블루닷은 데이터 센터를 통해 콘텐츠 스트리밍 및 프로덕션 서비스 기업에게 반도체 기술로 가속된 지능형 동영상 처리 및 압축 기능을 제공합니다

- Located in Seoul, Korea
- Founded in August 2019

다양한 영상 서비스와 커지는 해상도

다양한 비디오 스트리밍 서비스

OTT

NETFLIX

wavve

hulu

Cloud Gaming

STADIA

NVIDIA
GEFORCE NOW

XBOX ONE

e-Sports

twitch

YouTube GAMING

虎牙直播
huya.com

e-Commerce

amazon live

f LIVE

淘宝网
Taobao.com

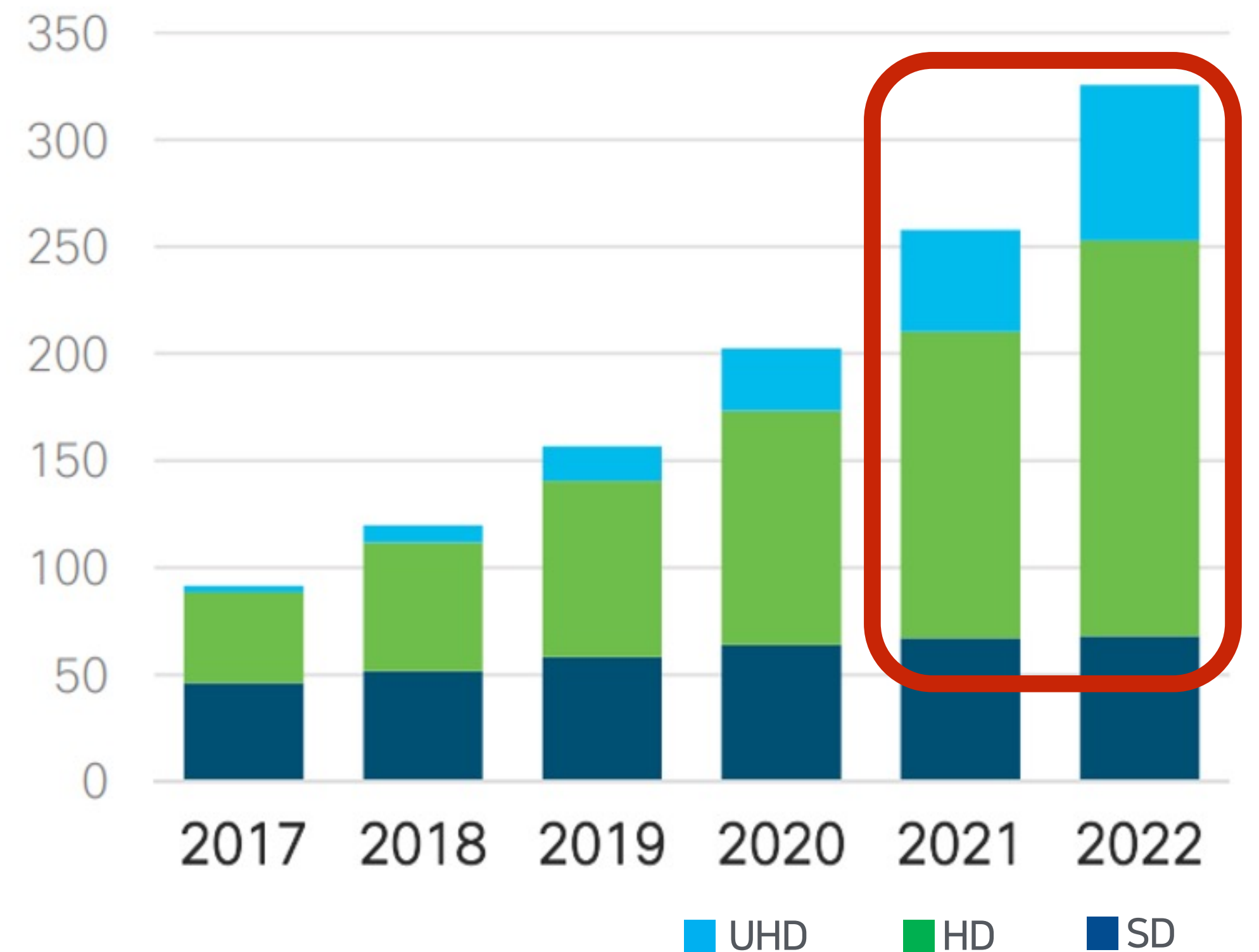
Live Streaming

f LIVE

YouTube Live

afreecaTV

Higher resolutions



(source: <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>)

인터넷 트래픽의 대부분은 영상 데이터

N DEVIEW
2021

2022

Video Data

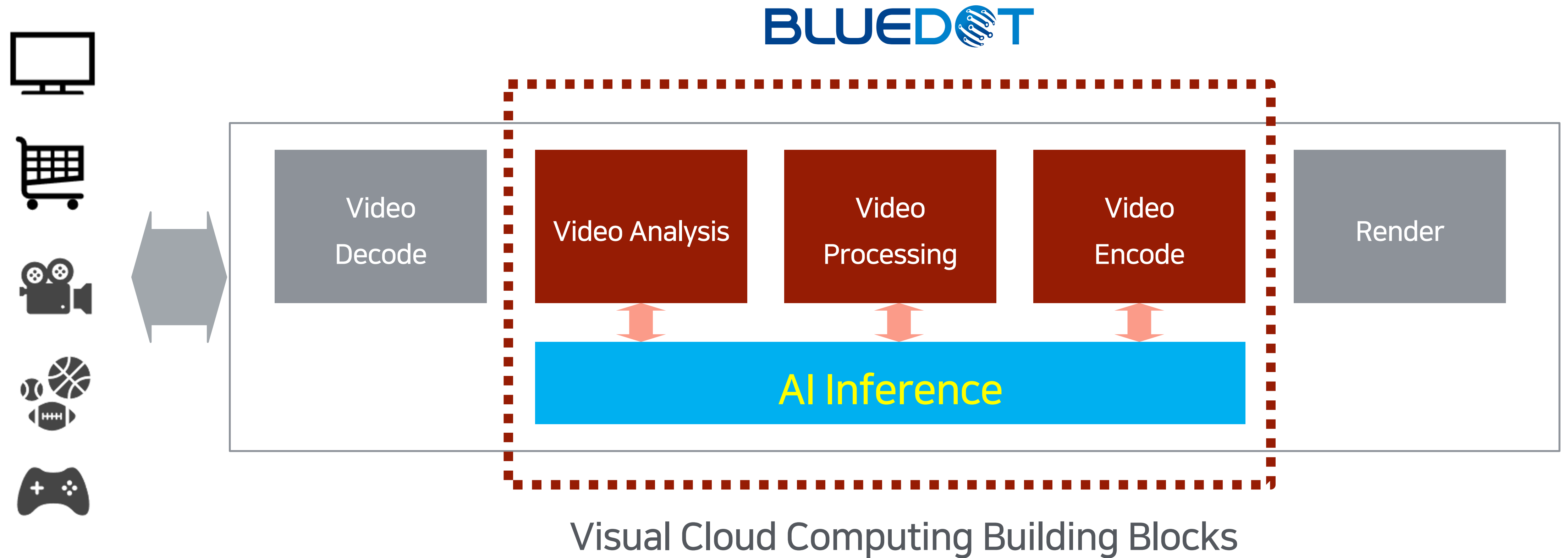
82%

Others

18%



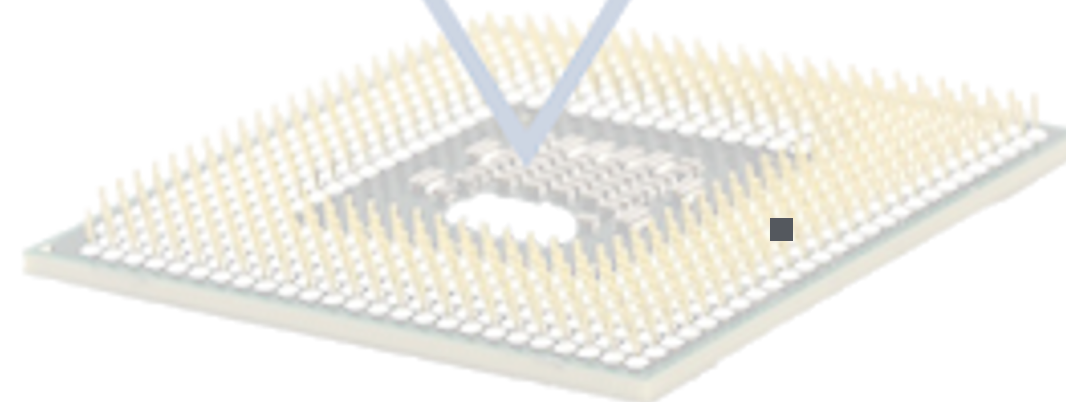
클라우드 영상처리 블록들



DeepField-SR

(AI-based super-resolution accelerator).

Xilinx Adaptive Computing Challenge 2020





Adaptive Computing Challenge 2020 Winning Projects

In the startup track of our [Adaptive Computing Challenge](#), we asked technology startups, early-stage technology companies, and research institutions to use any hardware platform built around Xilinx's 28nm/20nm/16nm devices to create an exciting new application designed to show hardware acceleration with programmable logic. Our startup track was all about real-world problem solving with technology innovation on Xilinx platforms.

LET'S MEET THE WINNING STARTUPS



First Place: **BLUEDOT**

DeepField-SR: AI-based Super-Resolution Accelerated by Xilinx FPGA Cards



Second Place: **SnowLake**

Molecular Dynamics Simulation Solution on Xilinx FPGA for Drug Discovery Market

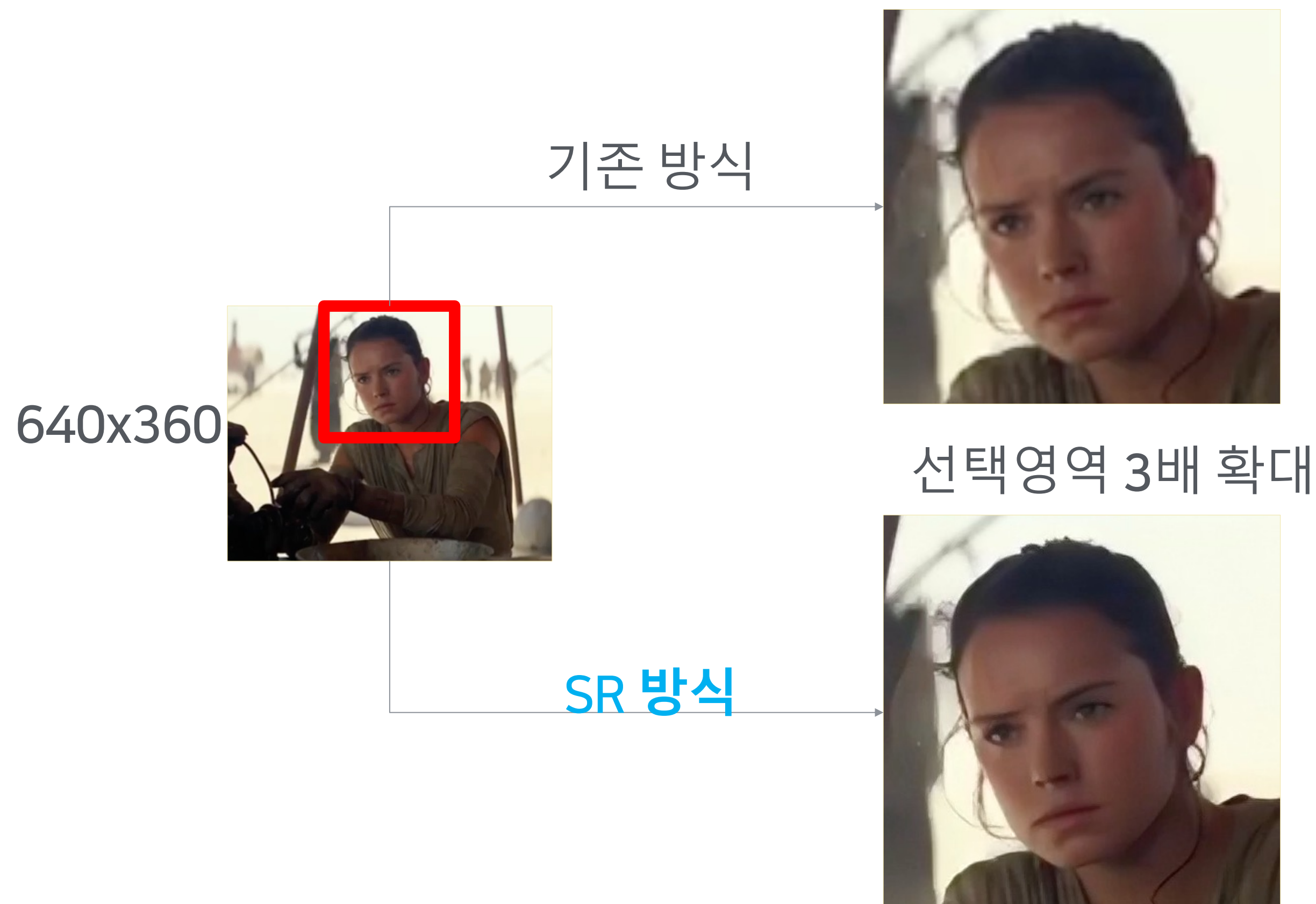


Third Place: **Katoid**

FPGA-powered Interactive Analytics

Super-Resolution 이란? 응용 분야는?

낮은 해상도에서 높은 해상도 이미지로



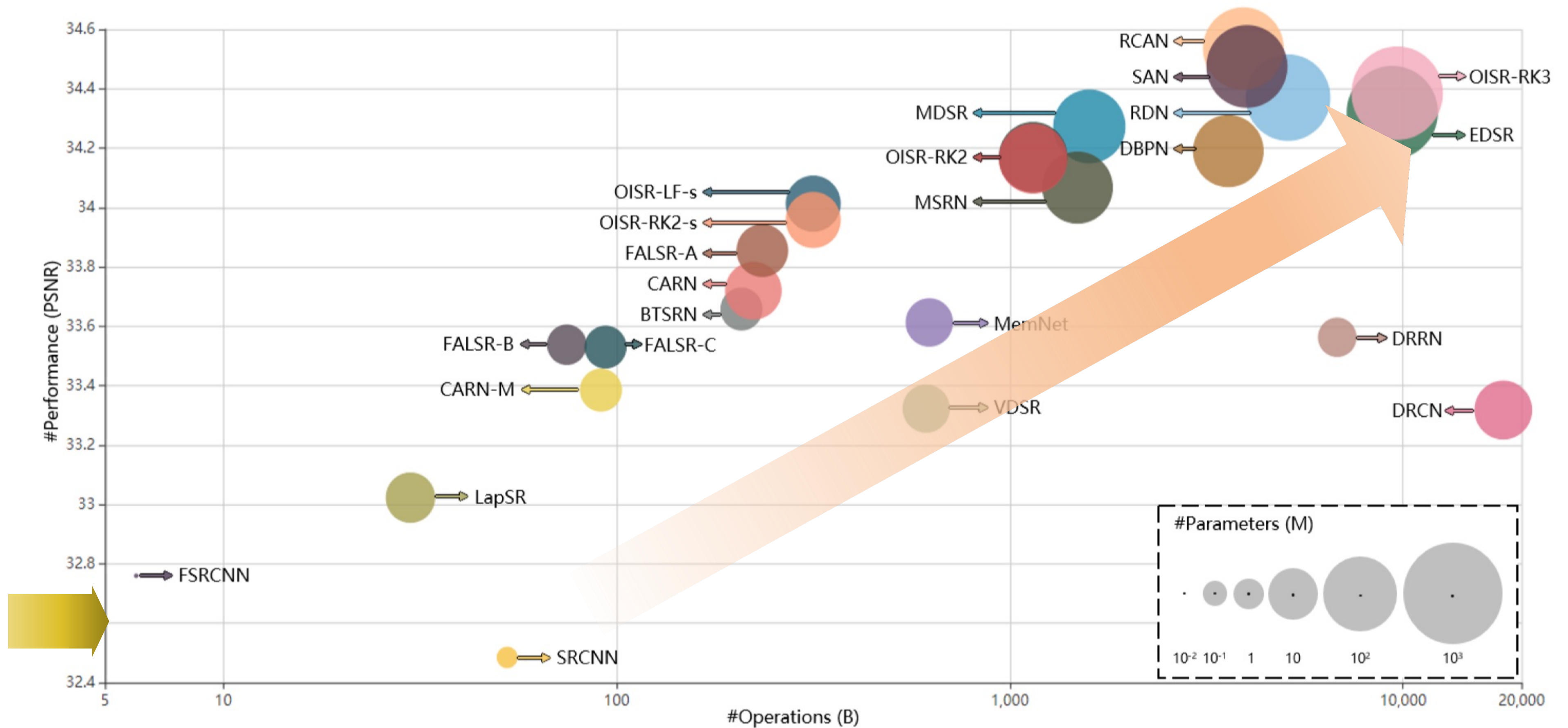
더 좋은 화질

- 비디오 스트리밍
- 영상 제작/편집

더 정확한 분석

- 감시 카메라
- 이상 감지
- 의료용 영상 분석

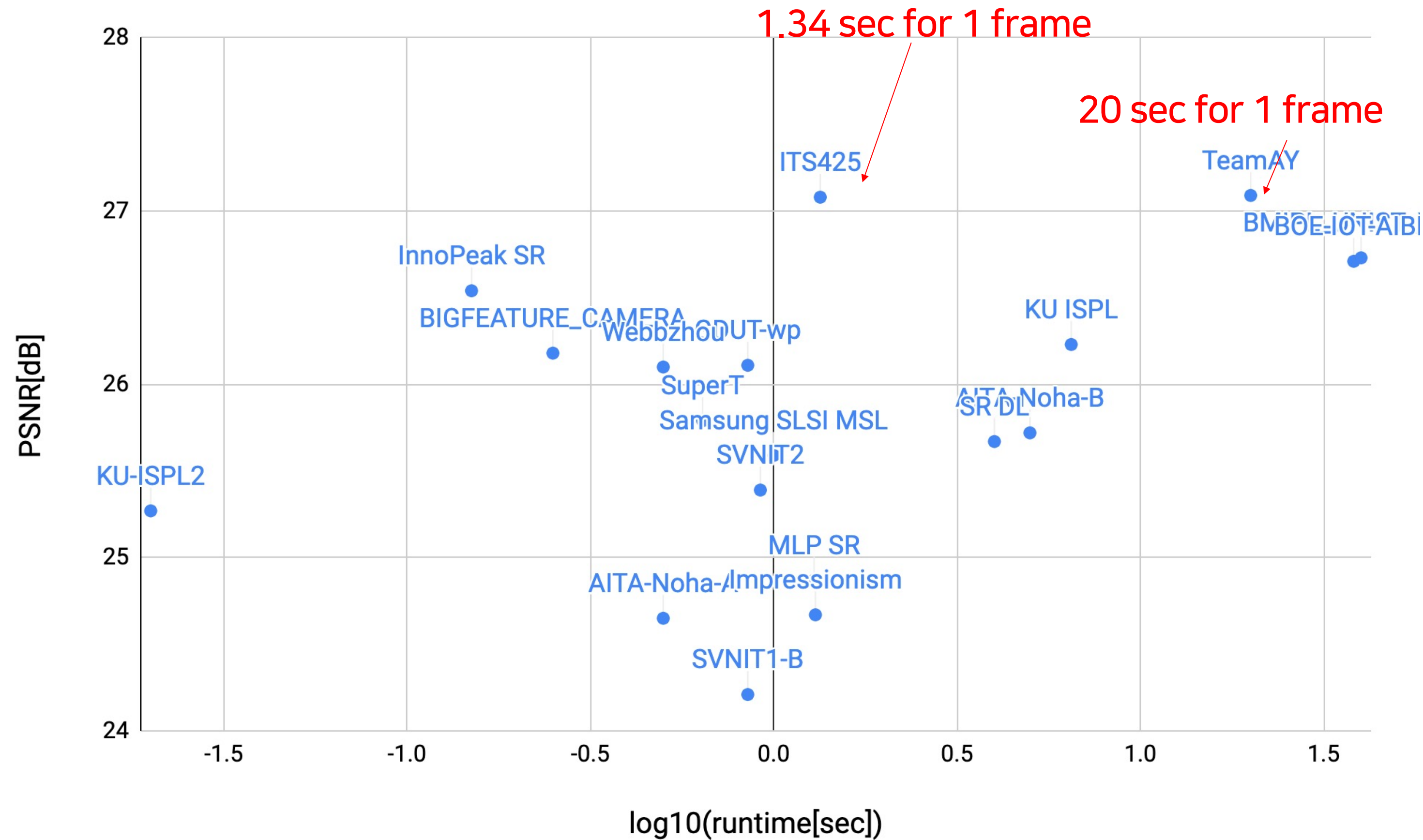
SR은 급격히 발전 중!



기존 방식 화질
이정도 수준
(SC, NE+LLE, KK,
ANR, A+, ..).

Deep-Learning based SRs

하지만 SR을 사용하기엔 문제도 있으니...



NTIRE 2020 Super-Resolution Challenge
Track1 Result (4x super-resolution)

연산량이 많아서
처리속도가 느림.
전력, 비용 문제 발생.

정지 이미지 위주.
동영상 화질은
만족스럽지 못함.

우리의 해결 방향

네트워크 구현

- 하드웨어로 설계 가능한 가벼운 네트워크
- 비디오 SR을 고려한 구조

학습 데이터

- 실제 서비스에 사용하는 형태의 데이터로 학습
- 동영상 인코딩/디코딩한 영상을 학습에 사용

추론 (Inference) with FPGA

- 빠른 수행 속도
- HLS(high-level synthesis)를 이용해서 설계시간 단축

서비스

- 클라우드 서버
- On-premises FPGA card

구체적인 수치로 보면

EDSR(winner of NTIRE 2017) 과 비교 : upscaling 360p to 1440p on NVIDIA Tesla V100

속도

44.9X

44 FPS vs. 0.98 FPS

화질

0.03

0.03 is better in
LPIPS score

필요 카드 수

1/30

30 NVIDIA V100s
vs. 1 Alveo U50
at 30 FPS

전력 소모

1/120

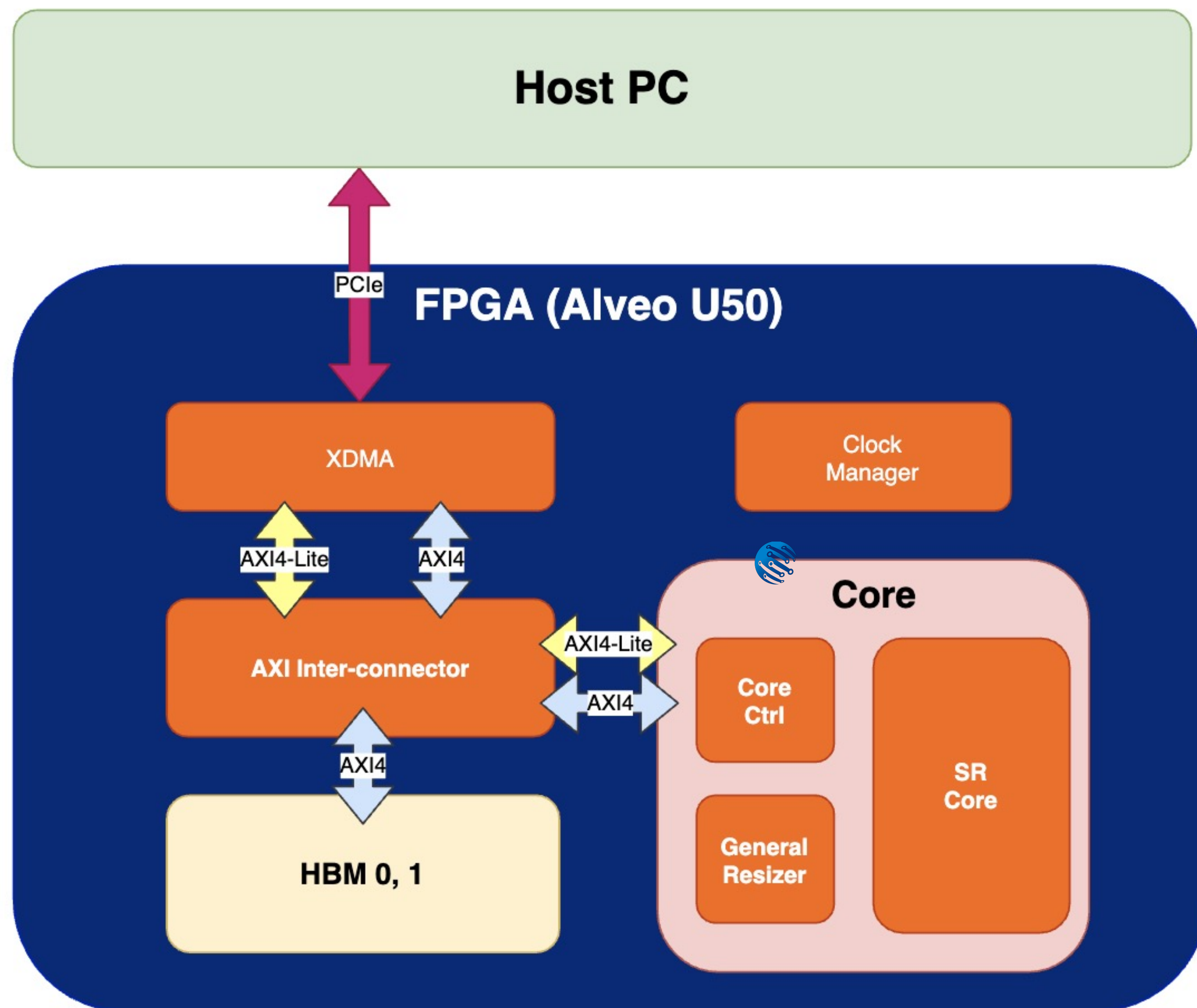
9000W(V100 x 30)
vs. 75W
at 30 FPS

DeepField-SR Deep Dive

제품 요구 사항

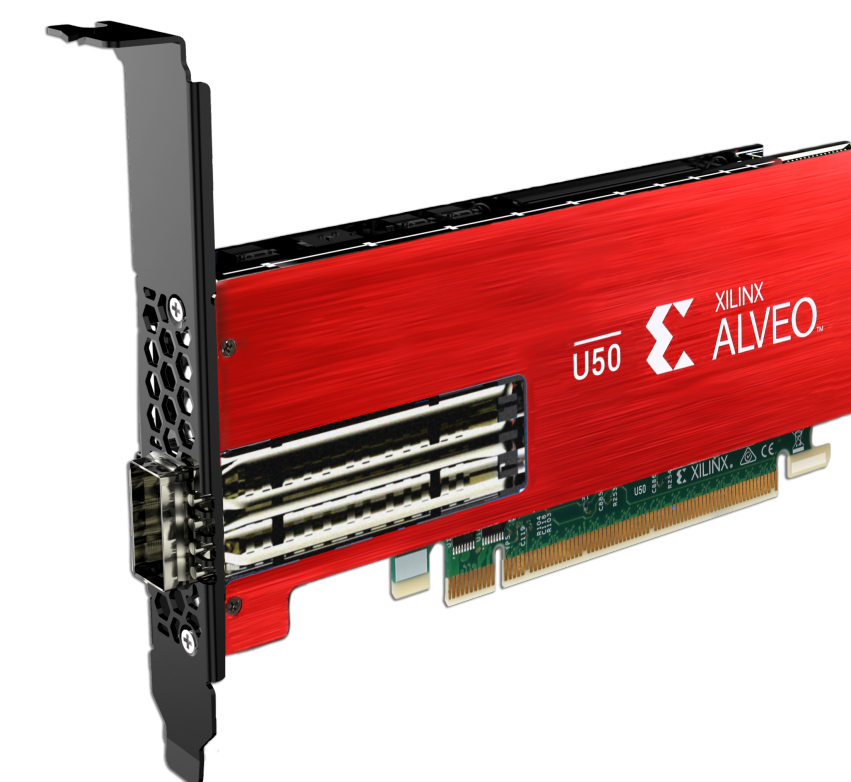
- 단순히 2x, 3x, 4x 정수배가 아닌 자유로운 배율
- FPGA에 구현 가능한 네트워크 크기
- 실제 비디오 스트리밍 영상에 화질이 좋아야 함
- 성능과 효율면에서 적당한 FPGA를 사용할 수 있어야 함 (1U 서버에 설치 가능)
- 클라우드 서버, on-premise 서버에 사용 가능
- 빠른 개발 환경
- 확장 가능한 구조 - 더 높은 해상도, 성능, 채널
- 사용자가 쉽게 사용할 수 있어야 함

Xilinx Alveo U50 선택

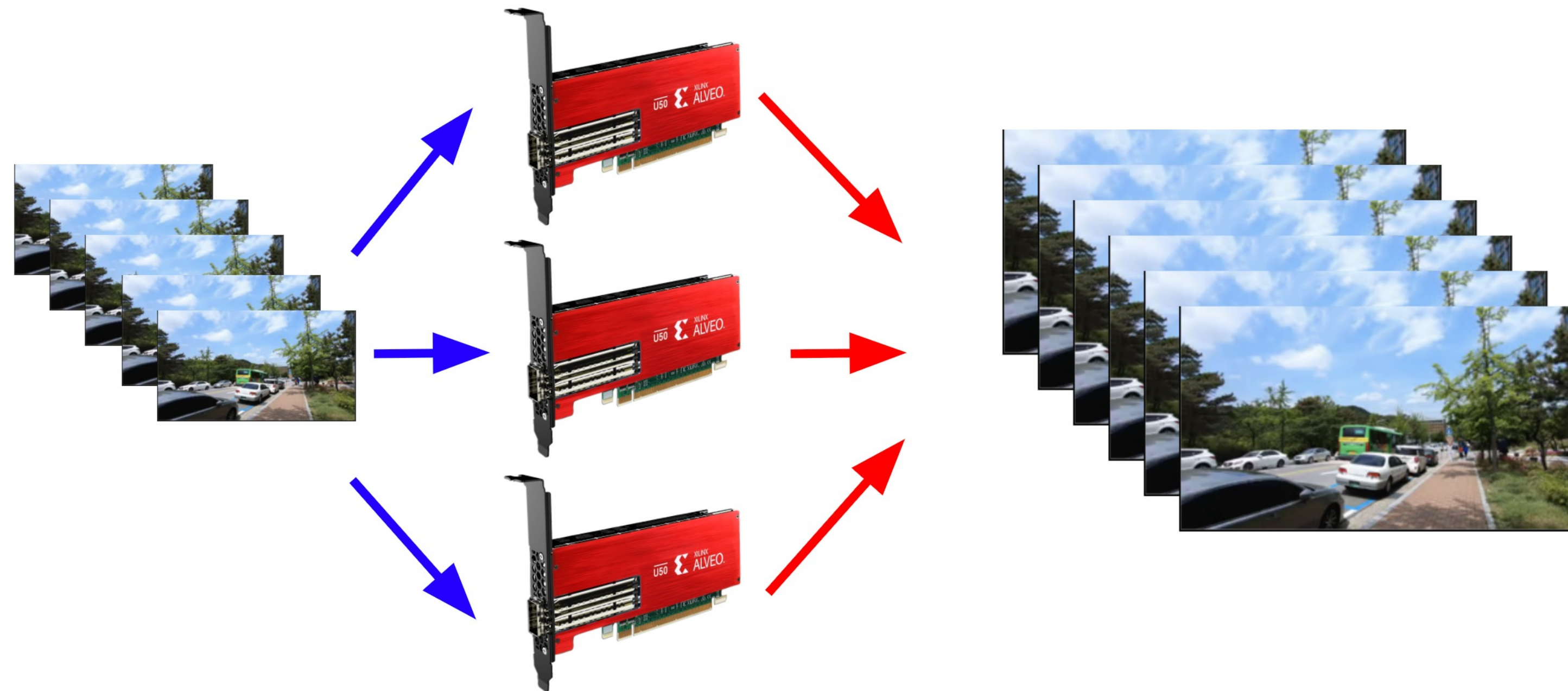


Alveo U50 FPGA card (Xilinx)

- HBM (High Bandwidth Memory)
- 1U server 설치 가능
- 약간 수정으로 AWS F1 instance에 사용 가능



확장 가능한 구조

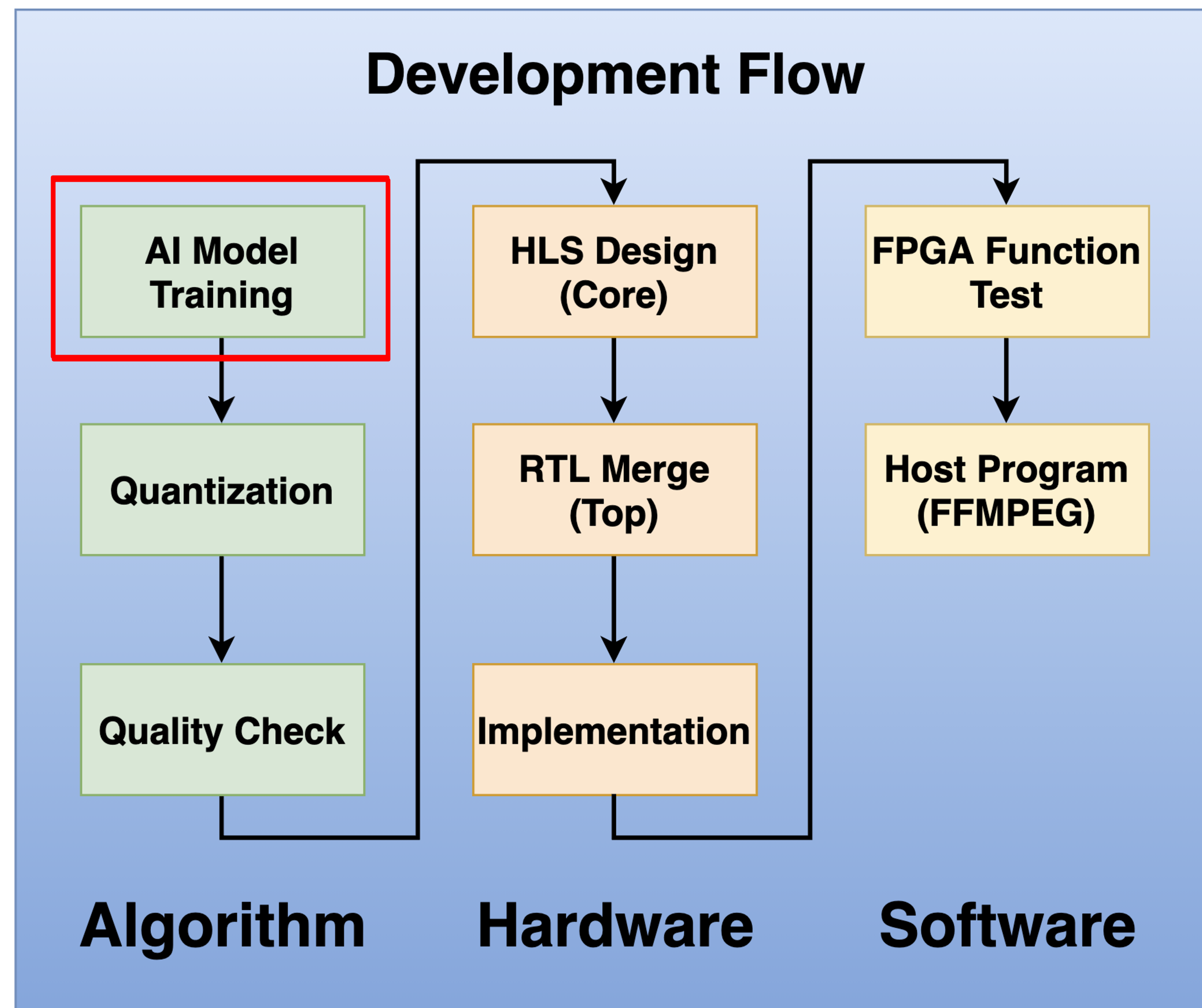
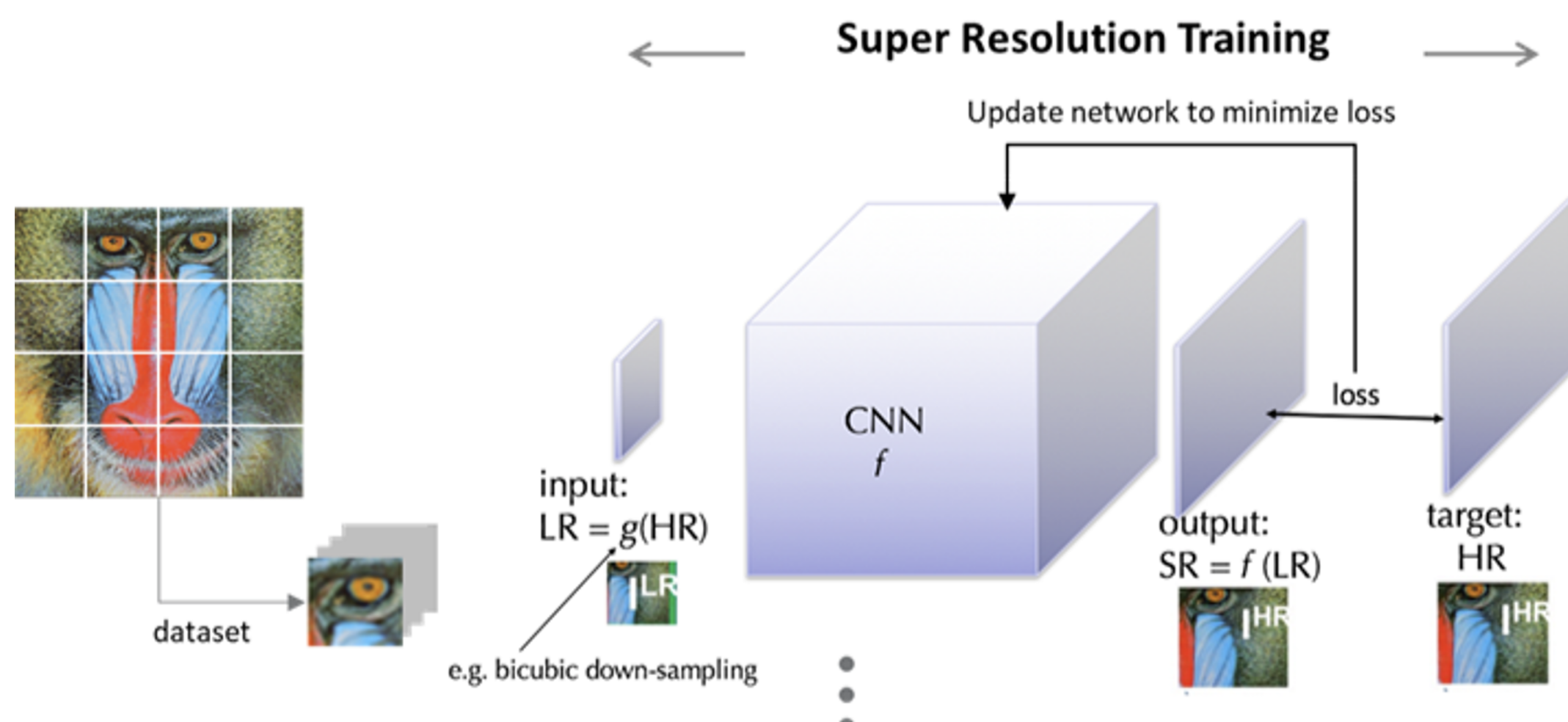


3 x U50 cards for upscaling
720p30 => 2160p30 in realtime

Deep Dive: Development Flow

Development Flow – AI Model

네트워크 구축, 학습



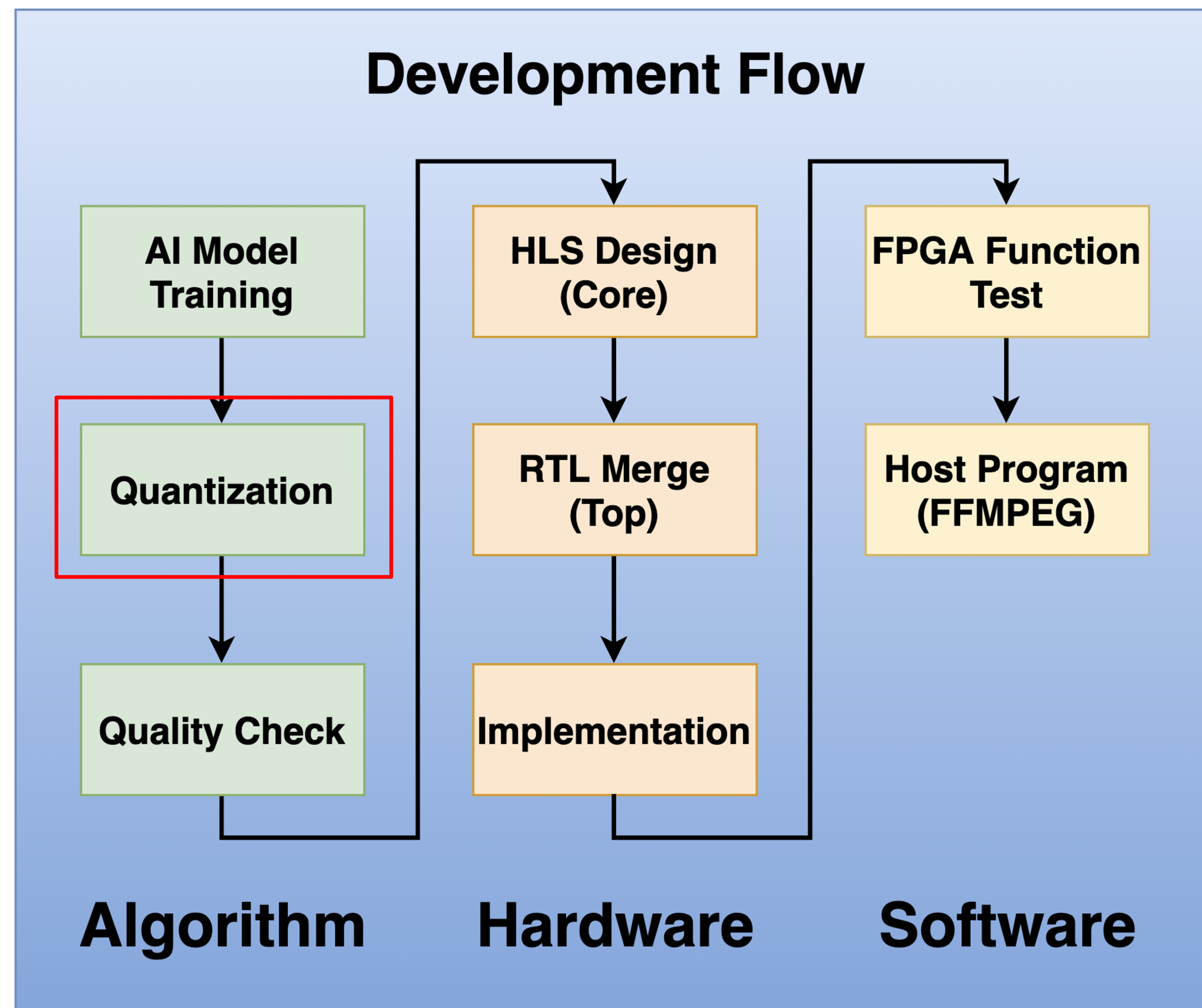
Development Flow - Quantization

Quantizing the network



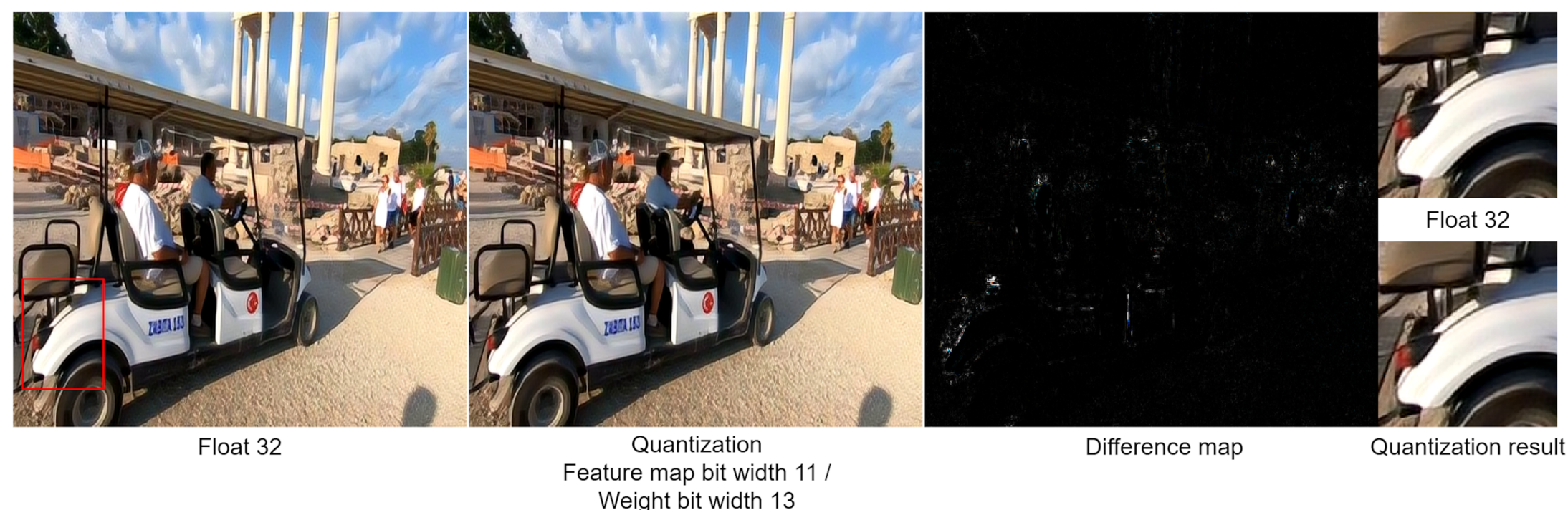
FP32

INT



Development Flow – Quality Check

Quality assessment

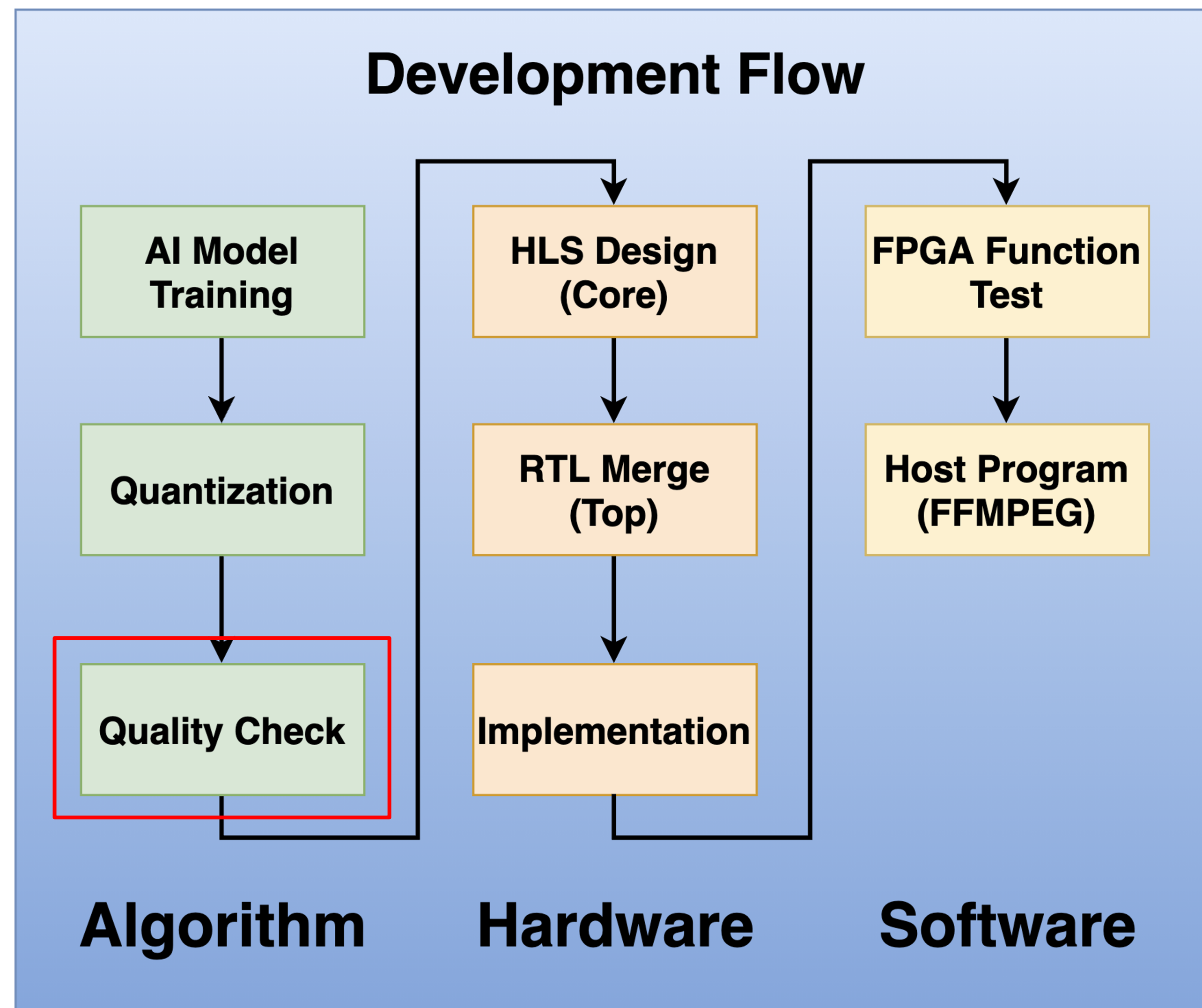


FP32

INT

difference
map

주관적 화질 중요!!



Development Flow – HLS Design

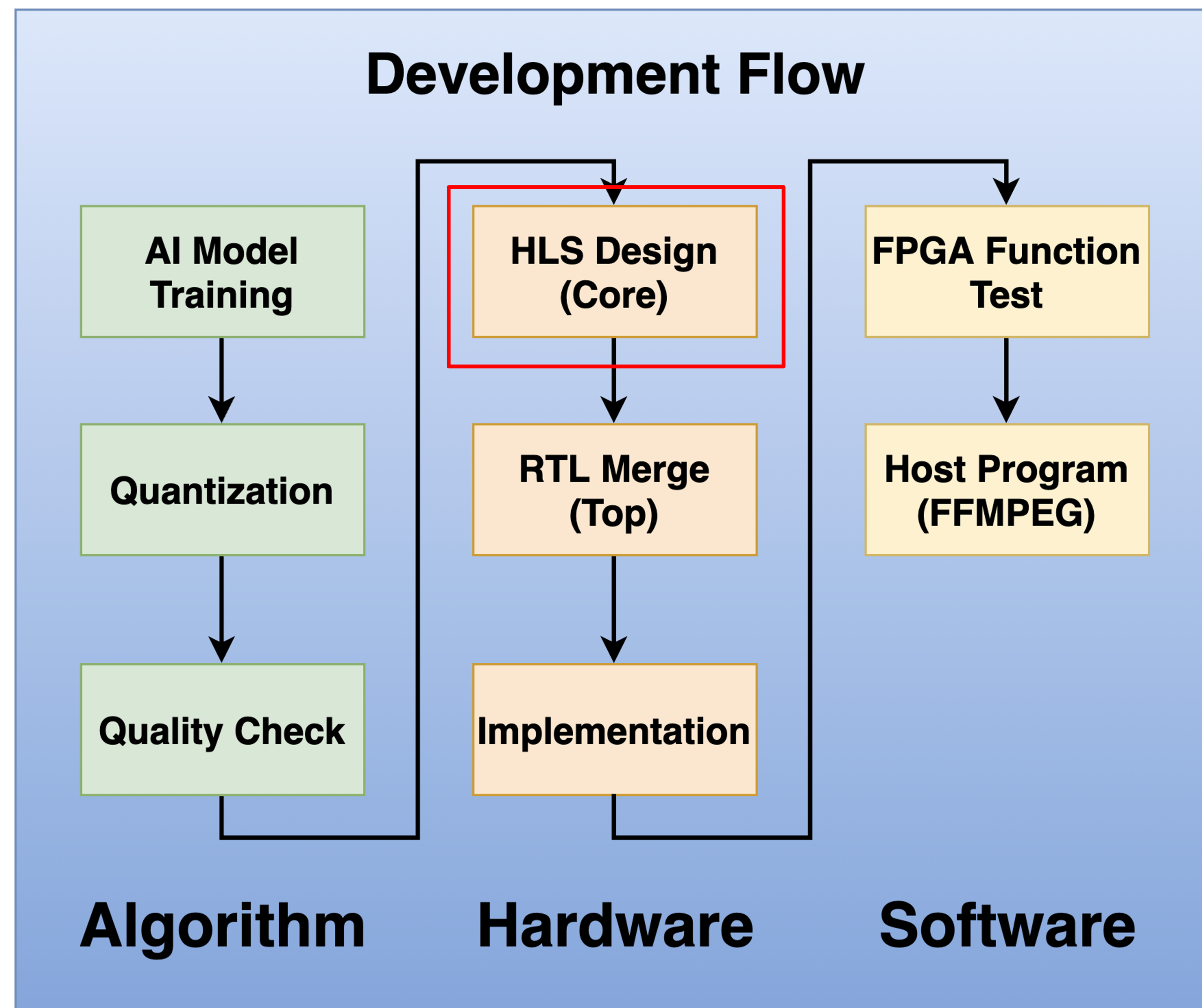
Designing modules

HLS
C/C++

```
1 template<int L_KX, int L_KY, int L_ICH, int L_OCH>
2 void conv_core(T)
3 {
4     const ap_int<WGT_W> wgt[L_ICH][L_KX][L_KY];
5     const ap_int<BIAS_W> bias[L_OCH];
6
7     // Parameter List
8     // OCH : output channel
9     // ICH : input channel
10    // KY : kernel Y
11    // KX : kernel X
12    for (och = 0; och < OCH; och++)
13    {
14        acc_result[och] = 0;
15        for (ich = 0; ich < ICH; ich++)
16        {
17            for (ky = 0; ky < KY; ky++)
18            {
19                for (kx = 0; kx < KX; kx++)
20                {
21                    mul = fmap_kernel[ich][ky][kx] * weight[och][ich][ky][kx];
22                    acc_result[och] += mul;
23                }
24            }
25        }
26        // added bias
27        ot_local.act[och] = acc_result[och] + bias[och];
28        //activation function (ReLU)
29        if (ot_local.act[och] < 0)
30        {
31            ot_local.act[och] = 0;
32        }
33    }
34 } // CORE
```

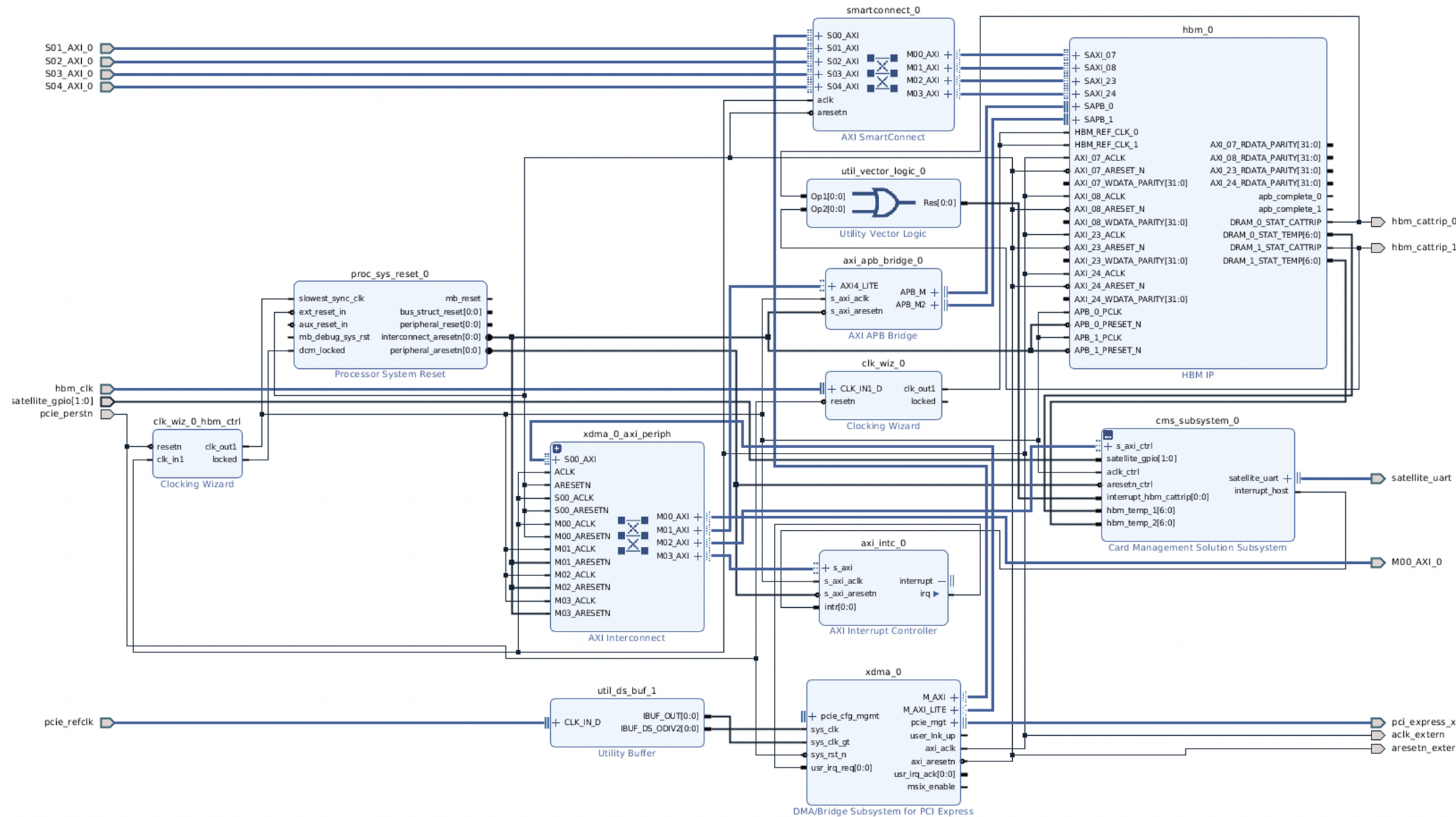
Directive

Synthesis &
Co-Sim



Development Flow – RTL Merge

Integrating modules
to build a top module



Development Flow

**AI Model
Training**

**HLS Design
(Core)**

**FPGA Function
Test**

Quantization

**RTL Merge
(Top)**

**Host Program
(FFMPEG)**

Quality Check

Implementation

Algorithm

Hardware

Software

Development Flow – Function Test

FPGA에 bit file 올려서 테스트

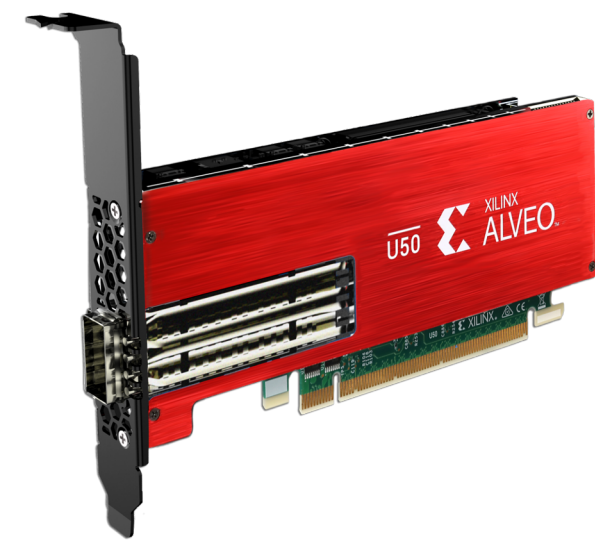
```
//reset
REG_W(pci_bar_handle, map_base_addr + REG_USER_RESET, (0b111));
sleep(1);
REG_W(pci_bar_handle, map_base_addr + REG_USER_RESET, (0b000));

//src resolution setting
REG_W(pci_bar_handle, map_base_addr + REG_PARAM_FG_PIC_IN_SIZE, (unsigned int)((res_src_v<<16) + res_src_h));
REG_W(pci_bar_handle, map_base_addr + REG_PARAM_FG_PIC_KN_SIZE, (unsigned int)((res_src_v<<16) + (res_src_h<<1)));
REG_W(pci_bar_handle, map_base_addr + REG_PARAM_FG_PIC_LP_CNT, (unsigned int)(buffer_size_lr));
REG_W(pci_bar_handle, map_base_addr + REG_PARAM_FG_PIC_OT_SIZE, (unsigned int)((res_wrp_v<<16) + res_wrp_h));
REG_W(pci_bar_handle, map_base_addr + REG_PARAM_FG_STAGE, (unsigned int)(0x0000));
//REG_W(pci_bar_handle, map_base_addr + REG_PARAM_SPACE_TO_DEPTH, (unsigned int)((0b0<<24) + (res_src_v<<12) + res_src_h));
REG_W(pci_bar_handle, map_base_addr + REG_PARAM_FG_PIC_VER_POS, (unsigned int)(0x0000));

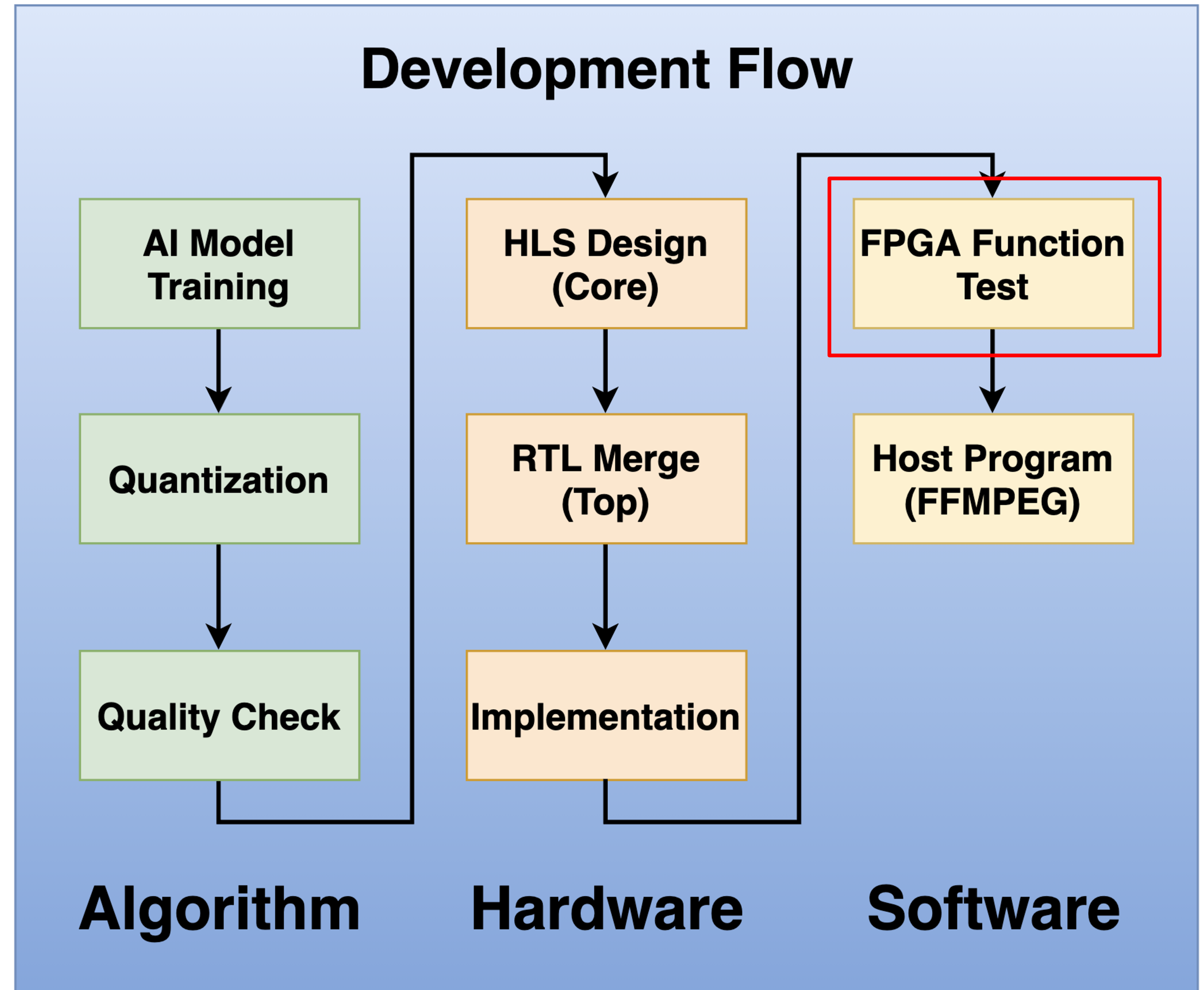
//loop counter setting
REG_W(pci_bar_handle, map_base_addr + REG_LOOPCNT_RDMA_1, (unsigned int)((res_src_v<<16) + res_src_h_ich));
REG_W(pci_bar_handle, map_base_addr + REG_LOOPCNT_CHAN_TO_CHAN9, buffer_size_wp);
REG_W(pci_bar_handle, map_base_addr + REG_LOOPCNT_WCHAN_MIX, buffer_size_lr);
REG_W(pci_bar_handle, map_base_addr + REG_LOOPCNT_WCHAN_SPLIT, buffer_size_lr);
REG_W(pci_bar_handle, map_base_addr + REG_LOOPCNT_CHAN_MERGE, buffer_size_lr);
REG_W(pci_bar_handle, map_base_addr + REG_LOOPCNT_CHAN_SPLIT_2, buffer_size_wp);
REG_W(pci_bar_handle, map_base_addr + REG_LOOPCNT_WDMA_1, (unsigned int)((res_wrp_v<<16) + res_wrp_h));

//dma option setting
REG_W(pci_bar_handle, map_base_addr + REG_ADDR_INTRA_RDMA_1, 0x200000000/AXI_BYTE_ALIGN);
REG_W(pci_bar_handle, map_base_addr + REG_ADDR_INTRA_RDMA_1_U, CORE_DDR_B_BASEADDR_U);
REG_W(pci_bar_handle, map_base_addr + REG_ADDR_INTRA_WDMA_1, 0x000000000/AXI_BYTE_ALIGN);
REG_W(pci_bar_handle, map_base_addr + REG_ADDR_INTRA_WDMA_1_U, CORE_DDR_A_BASEADDR_U);
```

Test
codes



U50

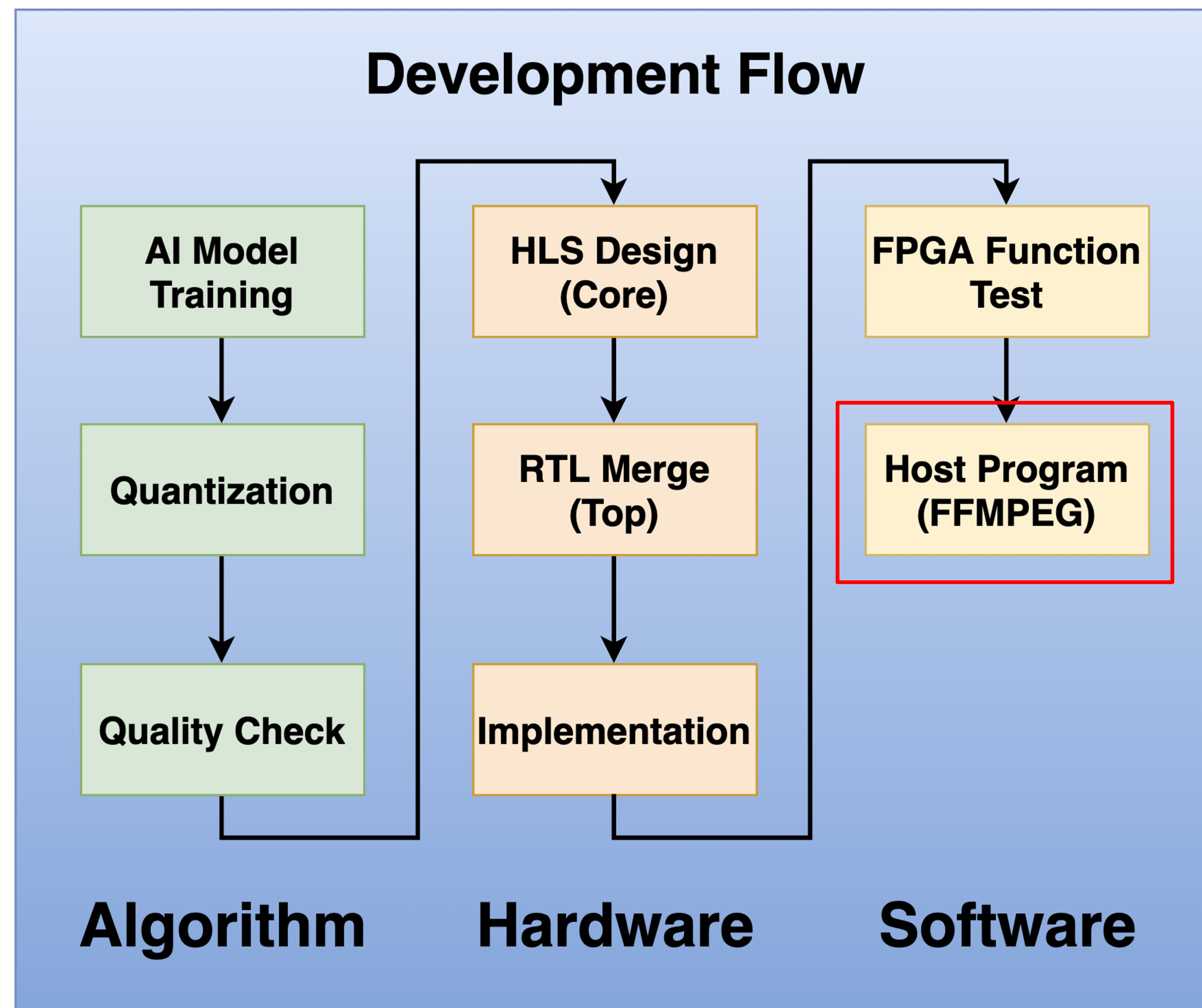


Development Flow – Host Program

FFMPEG에 필터 형태로 포팅



U50



Deep Dive: 해상도와 변환 비율

해상도와 변환 비율

동영상 서비스에 많이 사용
되는 이미지 크기

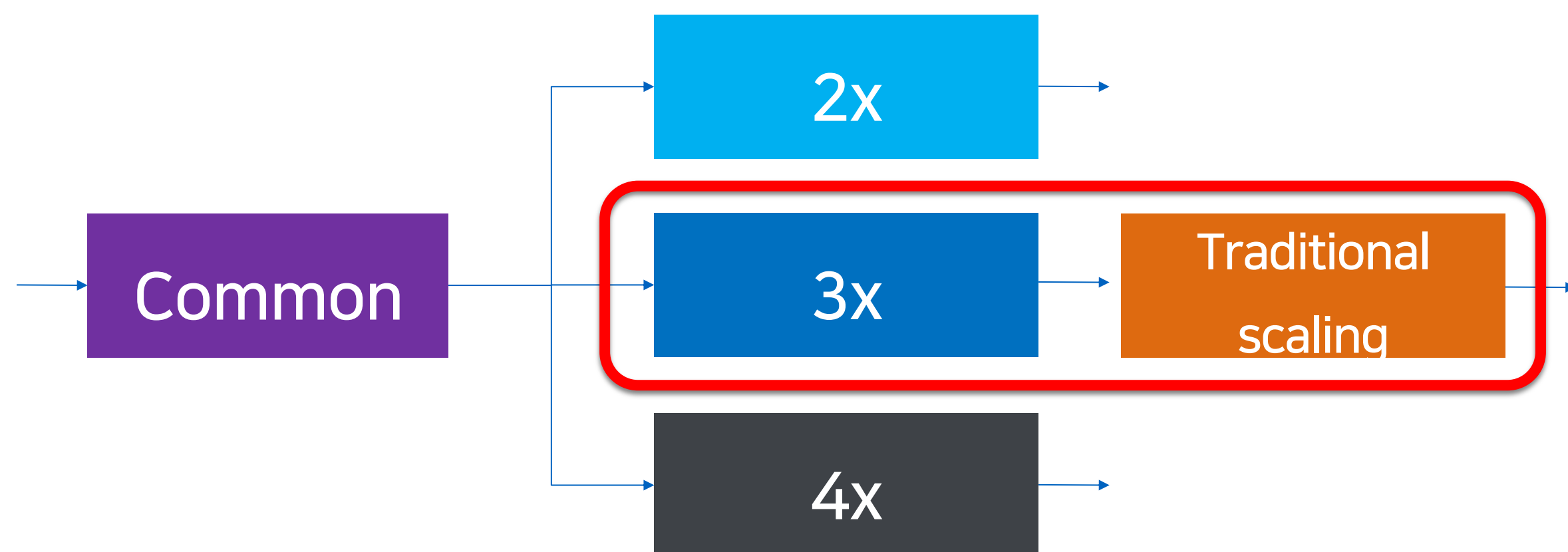
Name	Resolution
240P	426x240
360P	640x360
480P	720x480
720P	1280x720
1080P	1920x1080
1440P	2560x1440
2160P	3840x2160

변환 비율

	Output						
Input	240P	360P	480P	720P	1080P	1440P	2160P
240P	1x	1.5x	2x	3x			
360P		1x	1.33x	2x	3x	4x	
480P			1x	1.5x	2.25x	3x	
720P				1x	1.5x	2x	3x

이미지 크기도 다양하고...
변환 비율도 다양하고...

Hybrid Upscaling



네트워크로 모든 정수배 비율을 지원해야 하나?

근데...1.33 배 등등 다른 비율은?

1.0x ~ 4.0x 까지 지원

AI-based SR + traditional scaling

- 최대 입력 해상도는 1280x720.

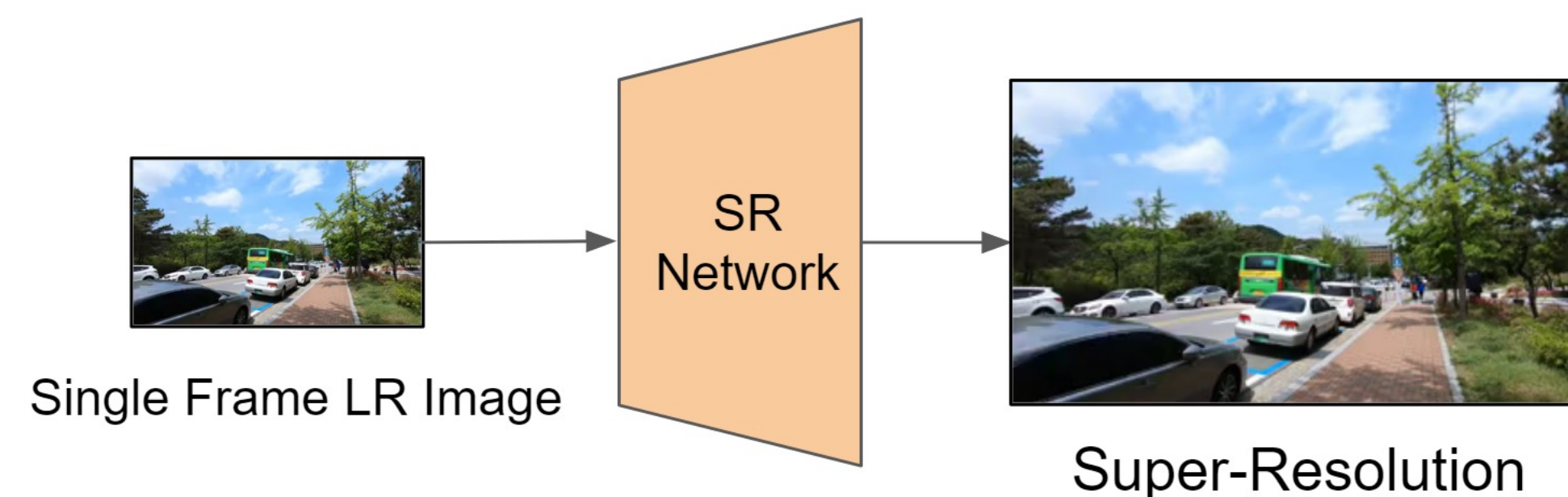
실험결과 2배, 4배를 지원하도록 학습된 다른 신경망과 비교해도 화질이 떨어지지 않음.

Deep Dive: 네트워크 구조와 학습

비디오 SR은 달라야 한다

Single Image Super Resolution

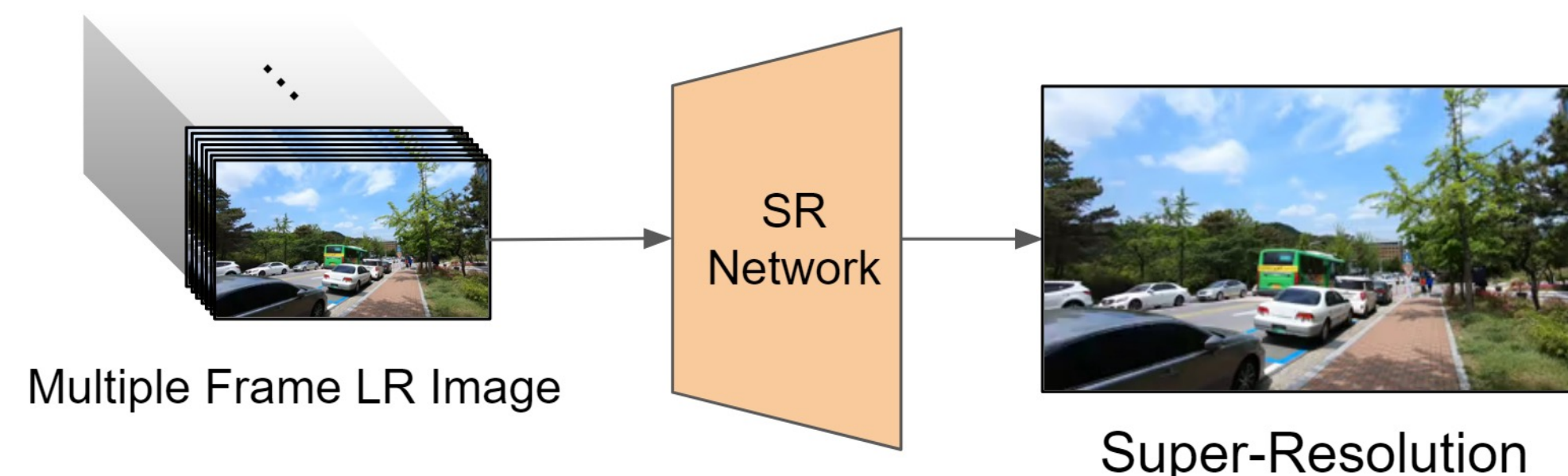
- 한장씩 추론에 사용
- motion-blurs 나 video coding 노이즈에 취약



한장을 사용하는 경우 (Single Image SR)

Multi-frame Super Resolution

- **인접한 여러장을 추론에 사용**
- 높은 연산량을 필요로 함



여러장을 사용하는 경우 (Multi-frames SR)

그 결과는

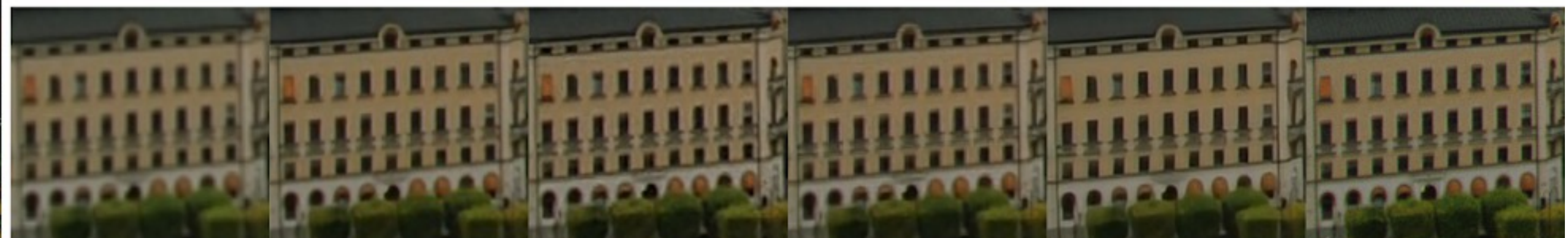


HR image



Too blurry

Visual artifacts due to
mis-alignment of frames



Bicubic

EDSR

DUF

EDVR

DeepField-SR

HR

traditional Single-Image SR

Video(multi-frame) SR

실제 서비스에서 사용하는 데이터로 학습

video coding artifacts 없
는 깨끗한 이미지

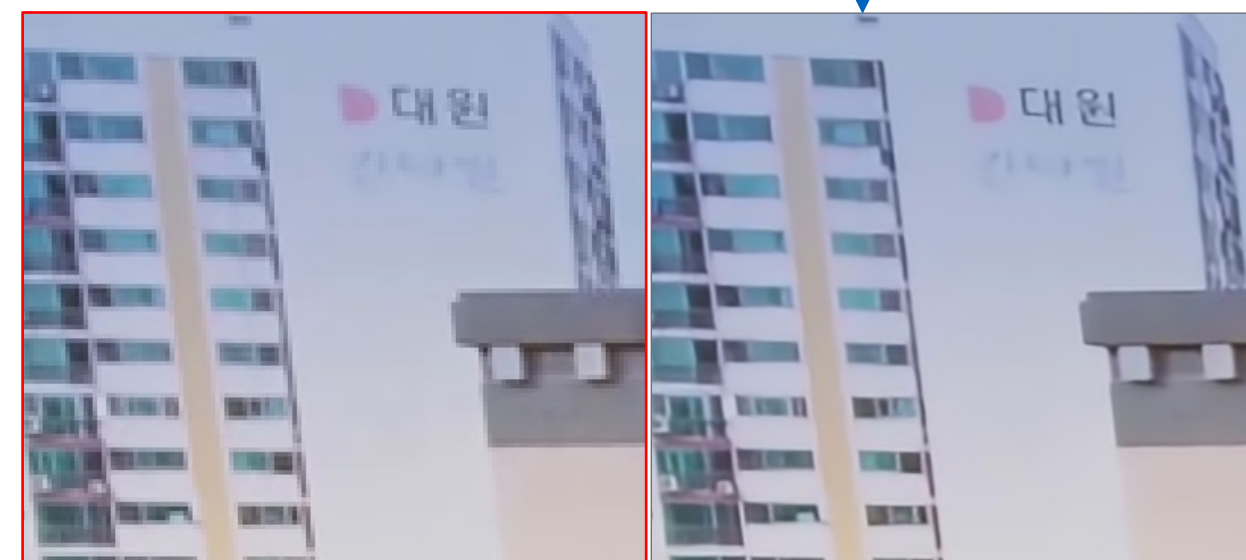
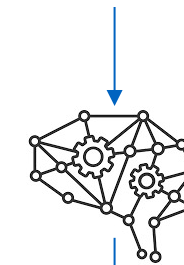
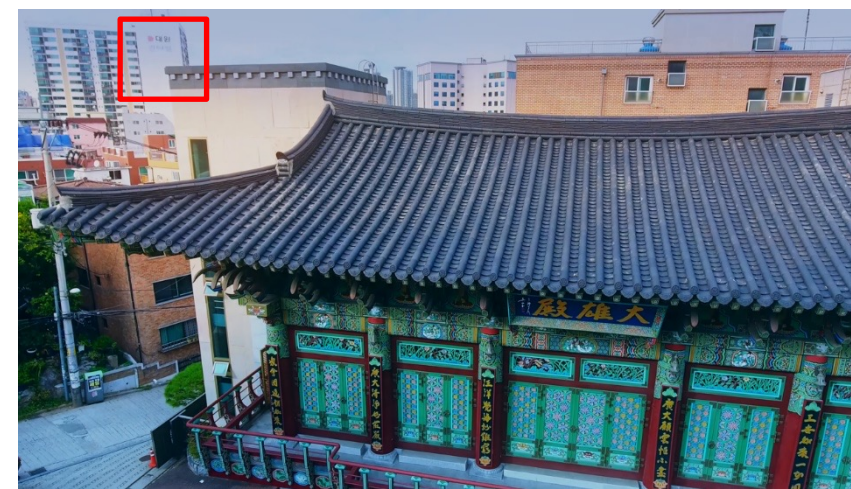


bicubic
resize



bicubic-resized LR 이미
지를 학습에 사용

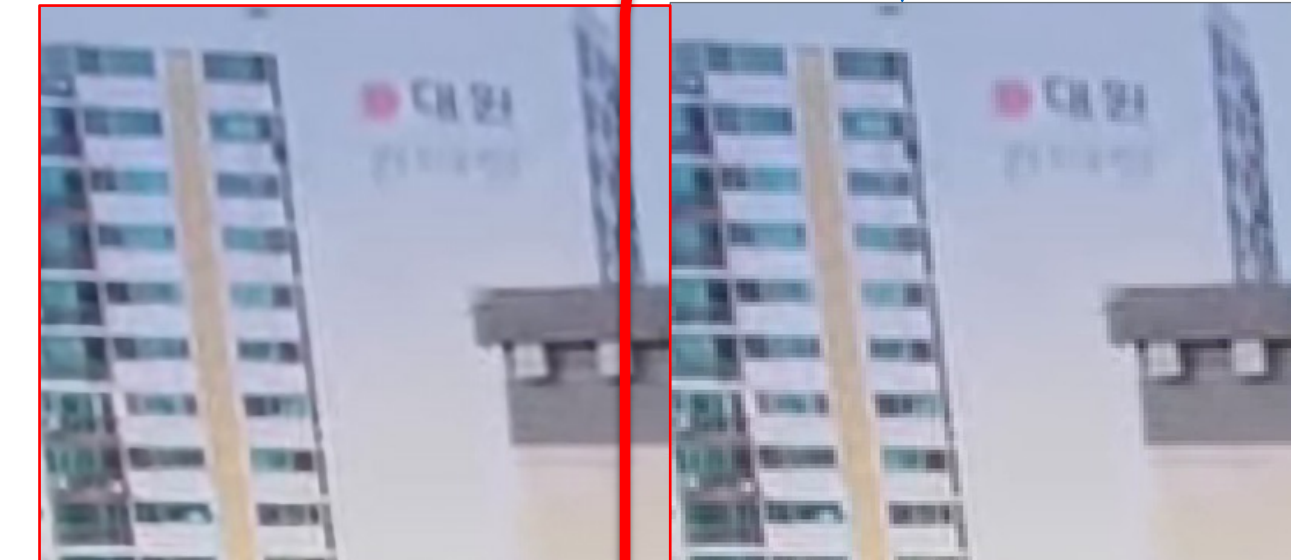
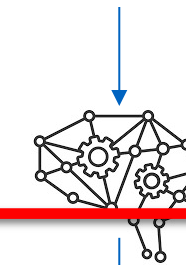
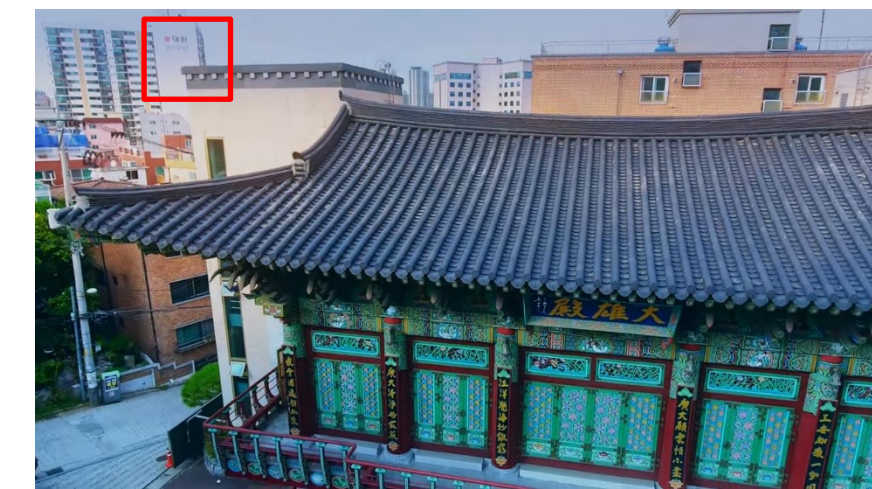
노이즈 없는 영상에서
는 잘 동작함



input

output

비디오를 입력으로 사용하면
잘 동작하지 않음



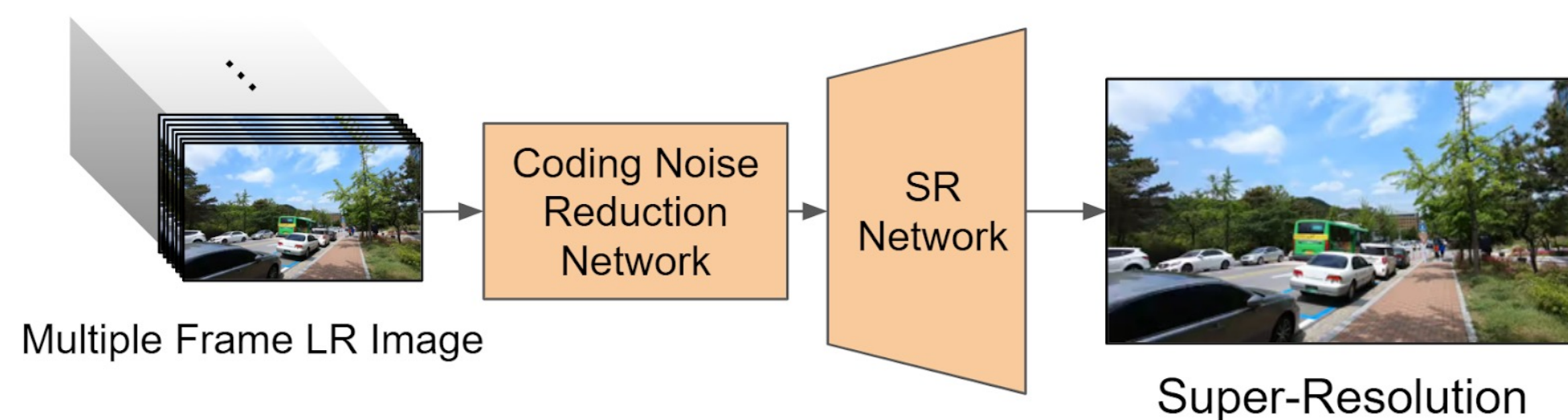
input

output

화질이 나빠진다.

코딩 노이즈를 줄이기 위한 네트워크

스케일링전에 코딩 노이즈를 제거 필요

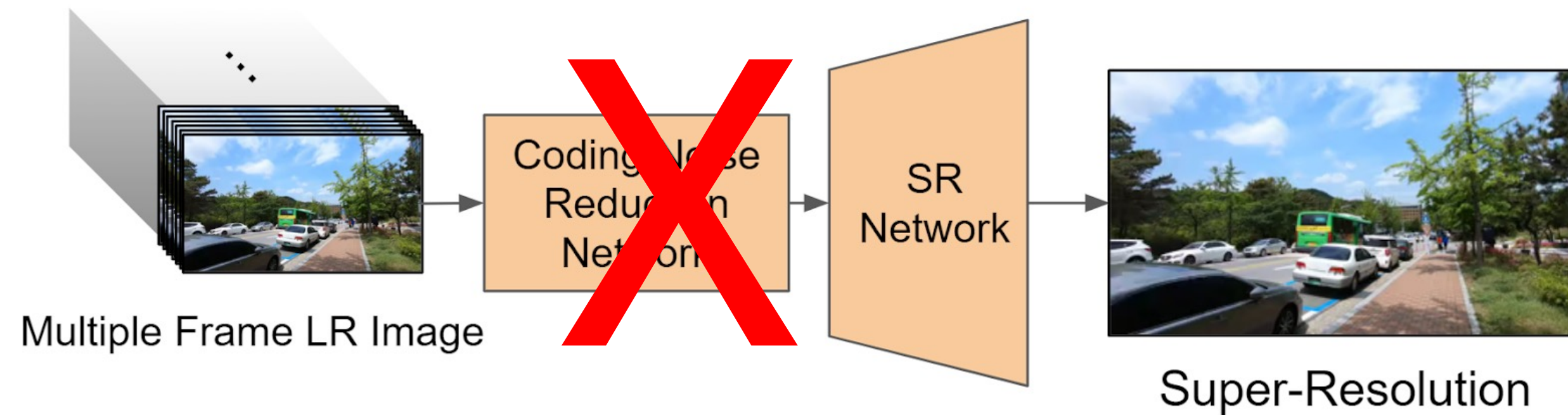


비디오 스트림에서 주로 사용하는 조건
(bitrate, condition)으로 인코딩한 데이터
로 학습

코딩 노이즈 많이 제거됨

MFQE: Yang et al., "Quality-Gated Convolutional LSTM for Enhancing Compressed Video", ICME, 2019

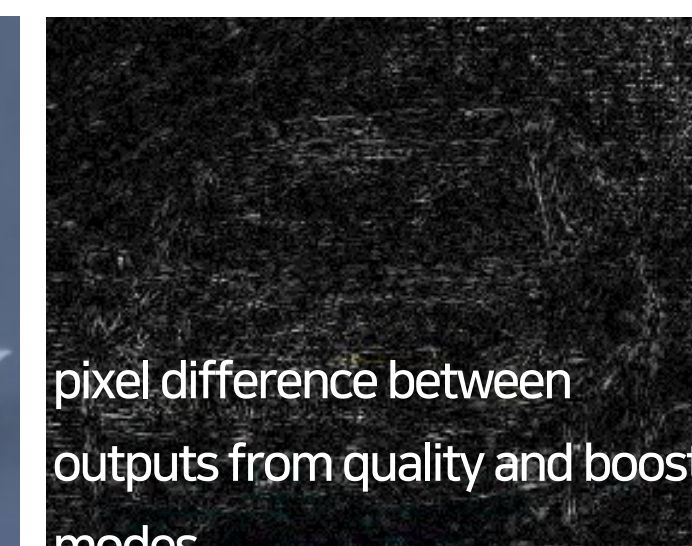
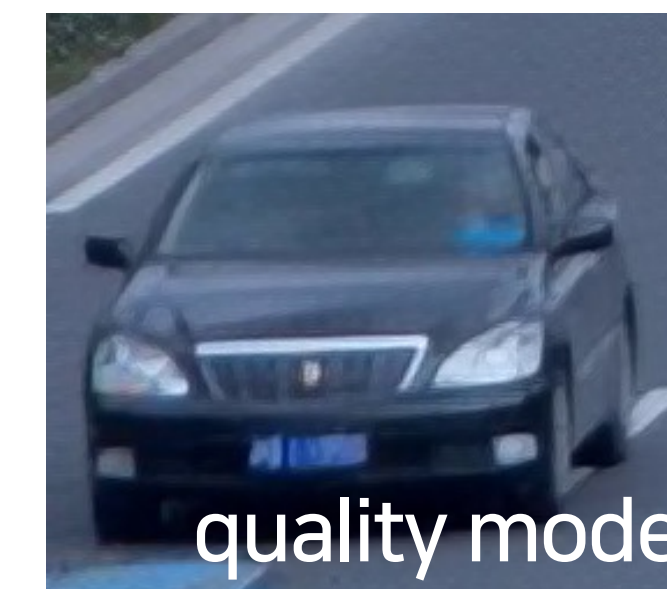
조금이라도 빨리 가고 싶으신 분은



코딩 노이즈 제거 네트워크는 생략 가능

=> **1.27x 빨라짐**

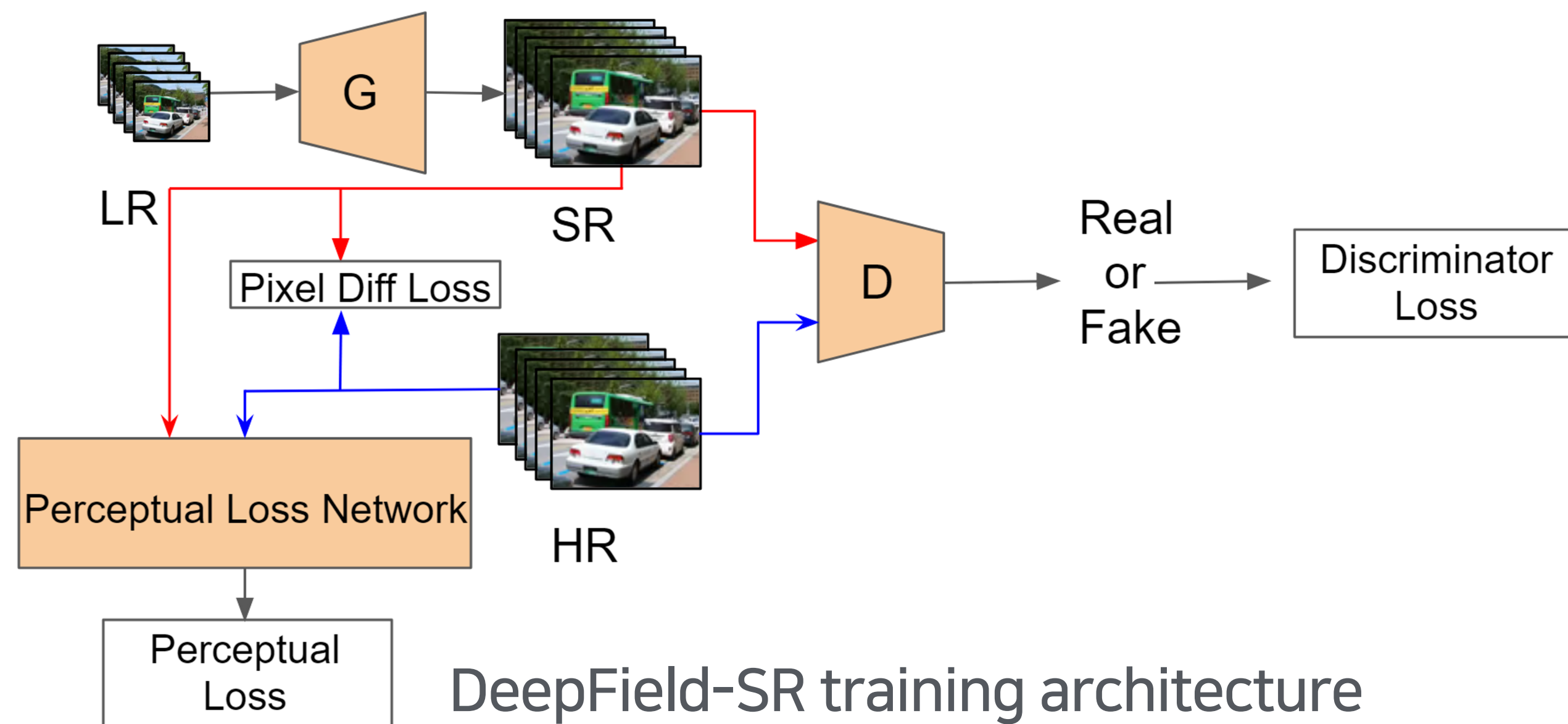
- quality mode = default process
- boost mode = bypassing noise reduction process



PSNR vs 주관적 화질?

주관적 화질이 매우 중요!

GAN, perceptual losses 사용



DeepField-SR training architecture

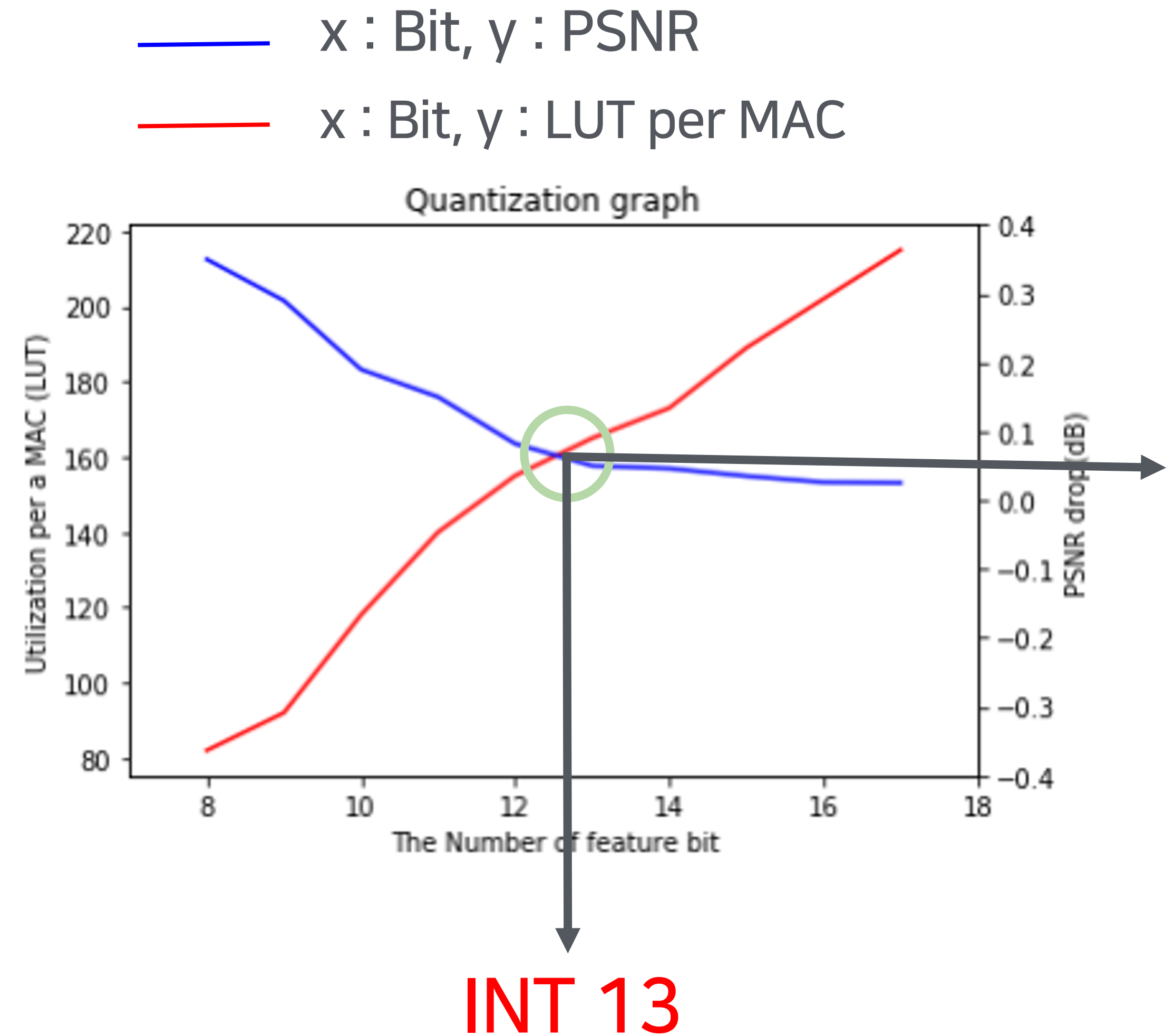
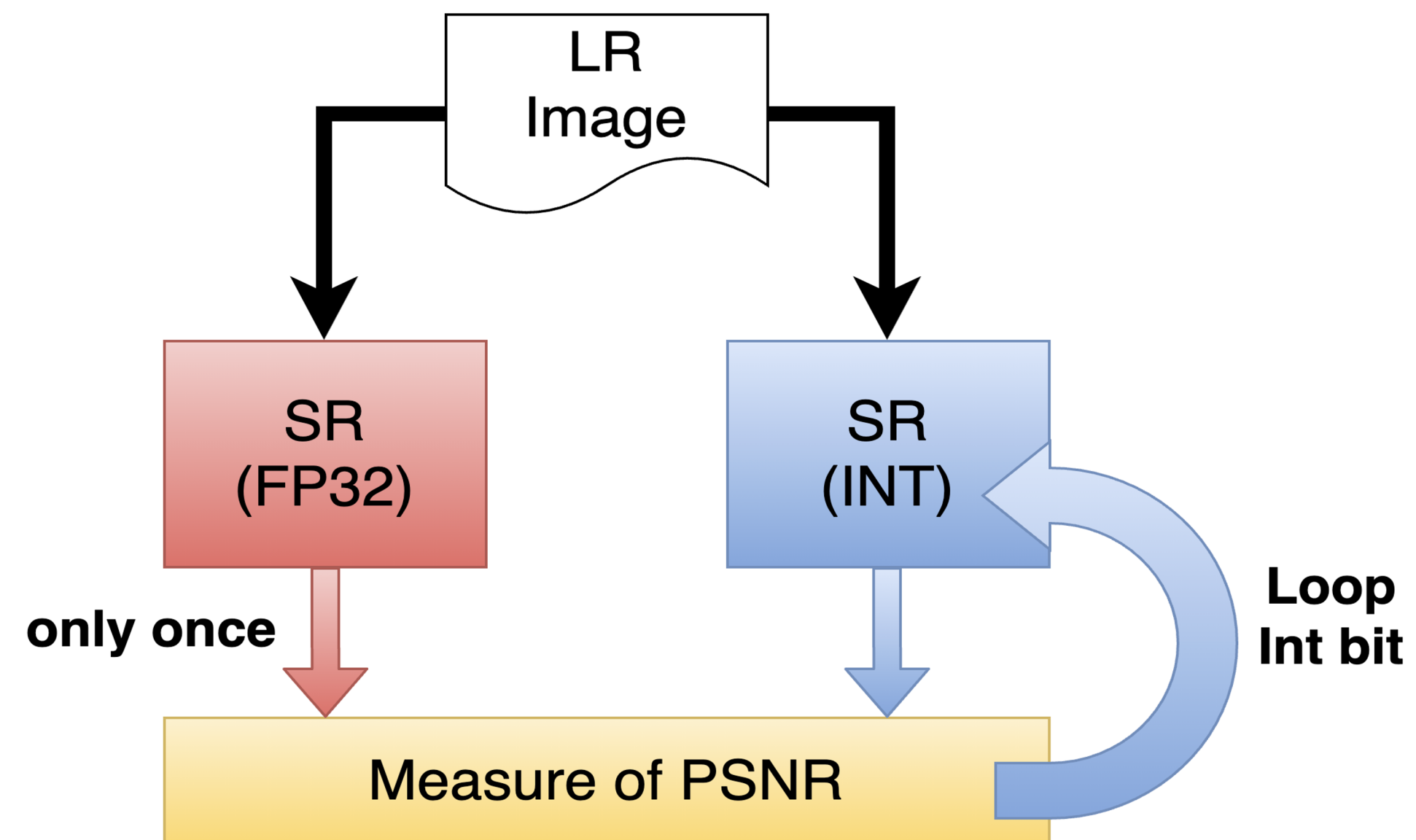


GAN 사용전

GAN 사용후

네트워크 양자화

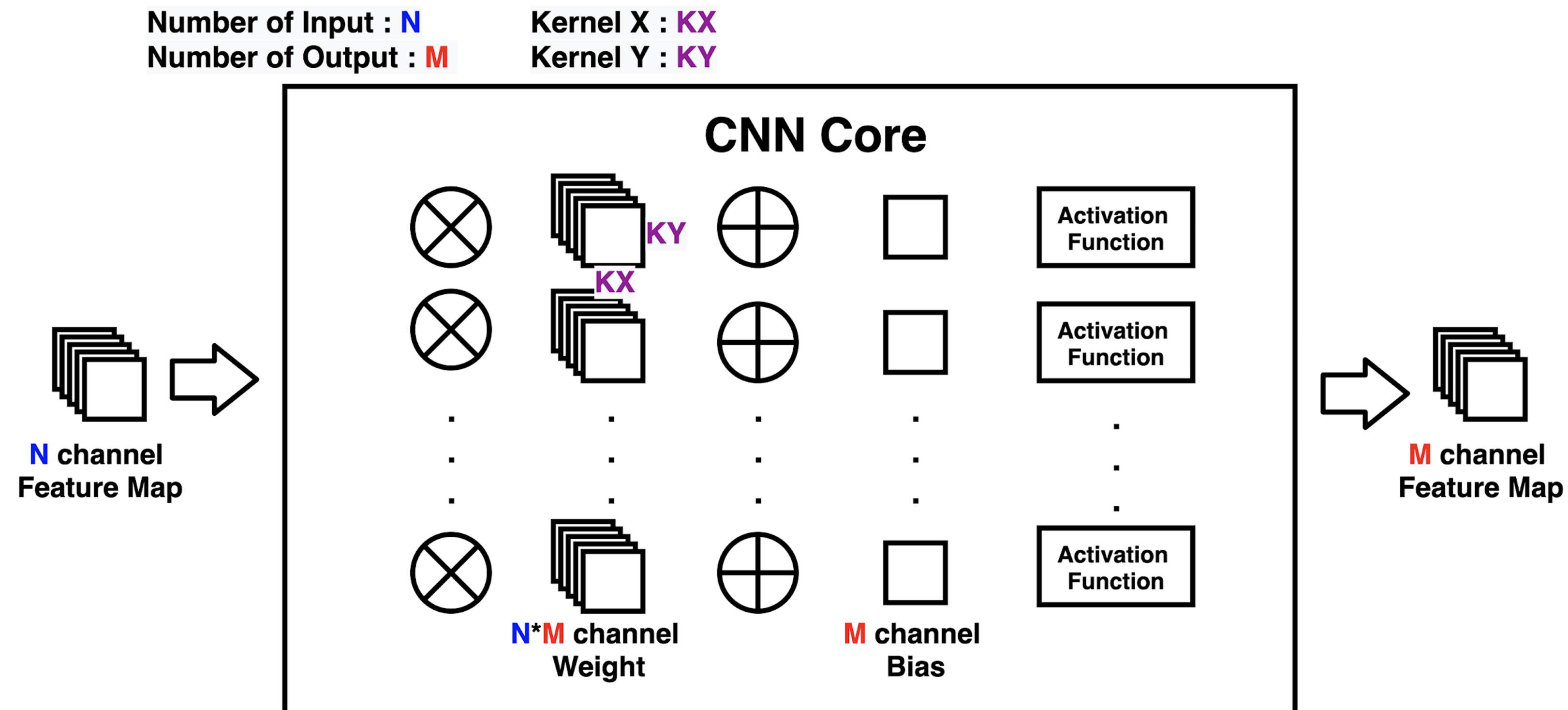
Precision을 바꿔가면서 반복 테스트



Deep Dive: 가속기 설계

뉴럴 네트워크의 구성 요소들

- Convolutional Neural Network (CNN)
- Multipliers and adders
- Channel size, kernel size, activation function, etc



Fixed Functional 가속기의 이점

Fixed functional 가속기

- 자원 효율이 더 좋다.
- DRAM bandwidth 가 더 낮다.

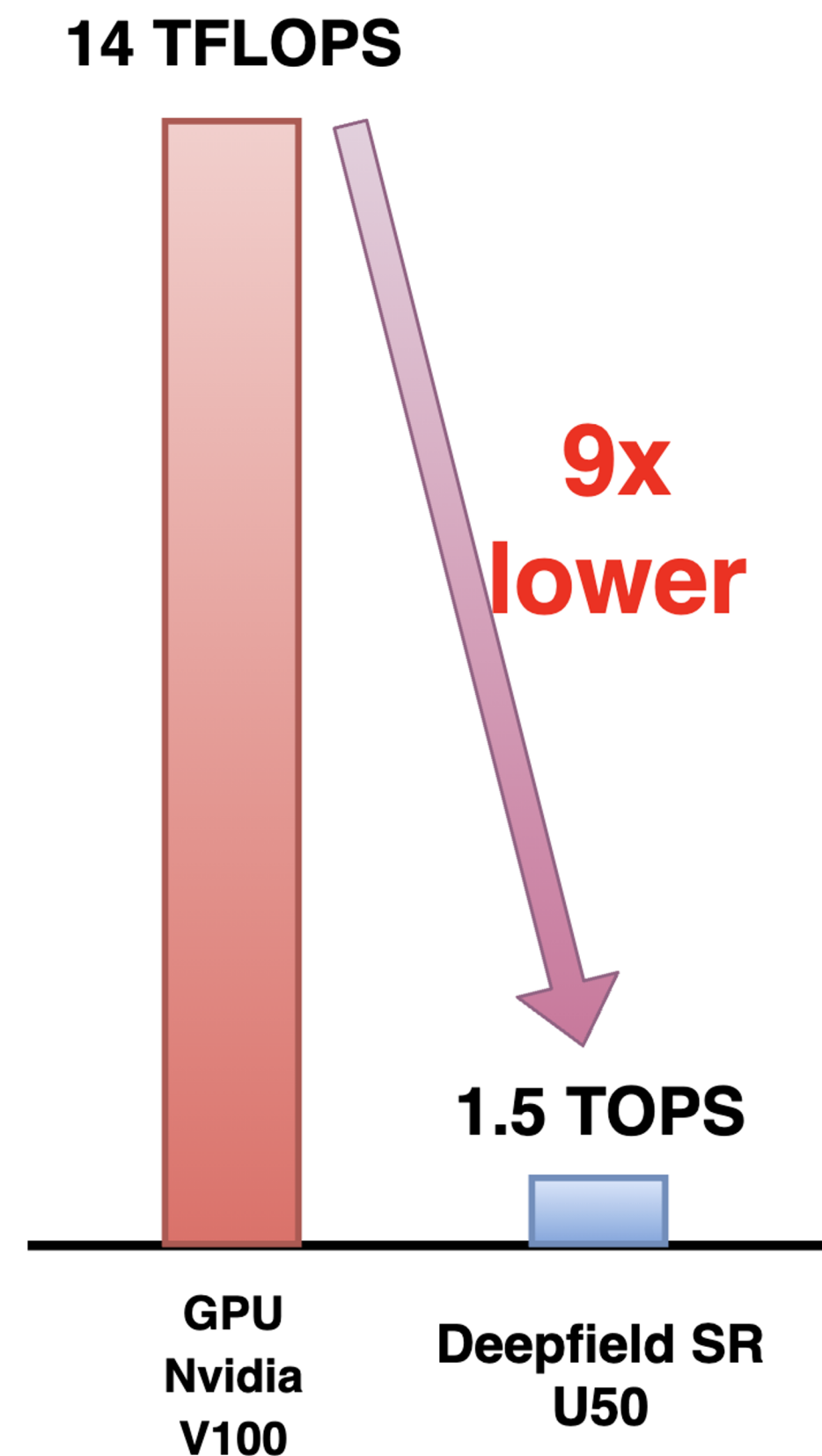
TOPS

- V100 : 14 FLOPS (FP32)
- Ours : 1.5 TOPS (INT13)

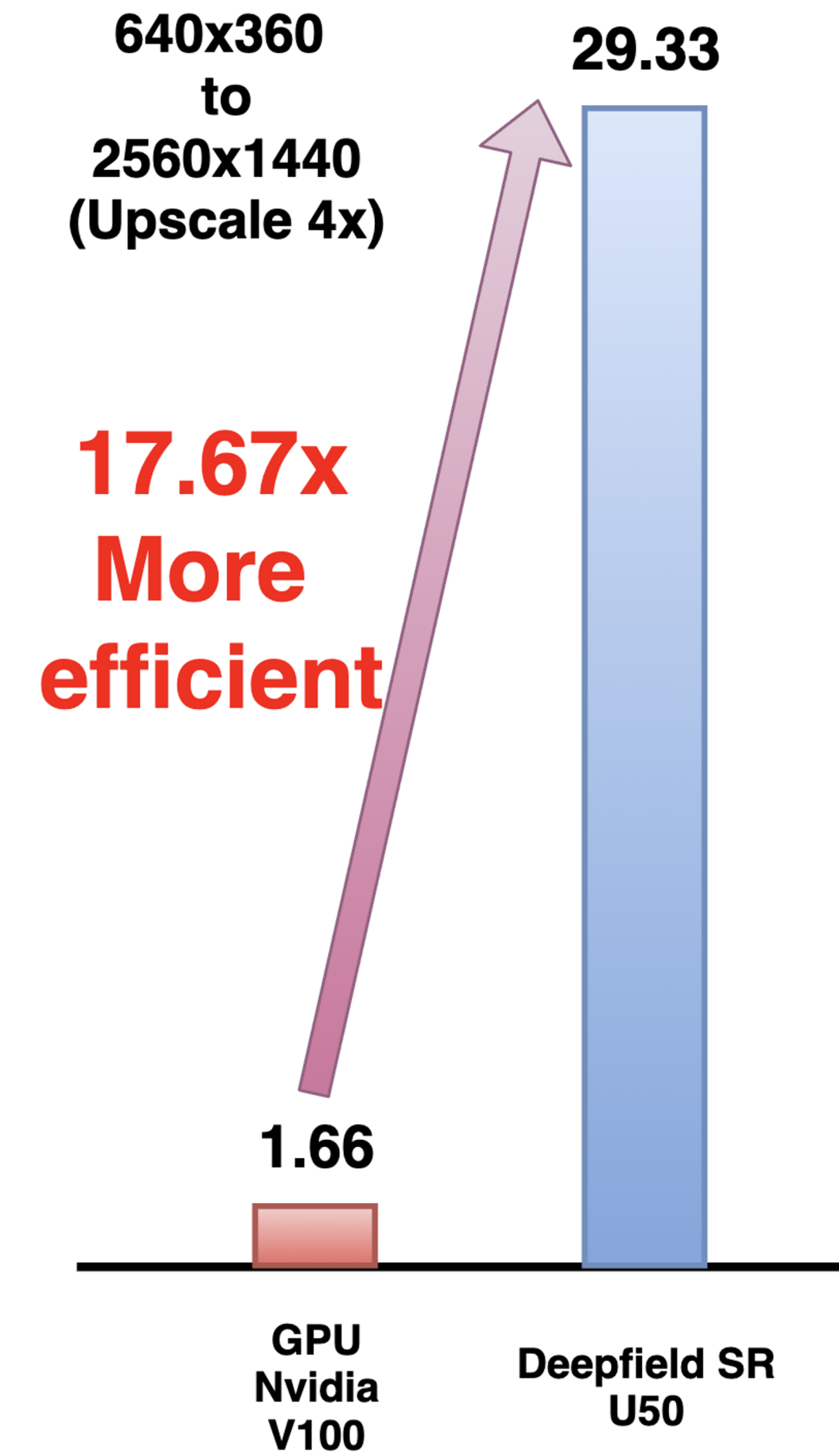
Frame / TOPS Ratio

- V100 : 1.66
- Ours : 29.33

< Number of TOPS >



< Frame / TOPS Rate >



HLS는 장점이 많습니다

Directive 조절

- 여러가지 구조, pipeline
- 동작 주파수 변경
- 성능/리소스 사용량 변경
- 다양한 target

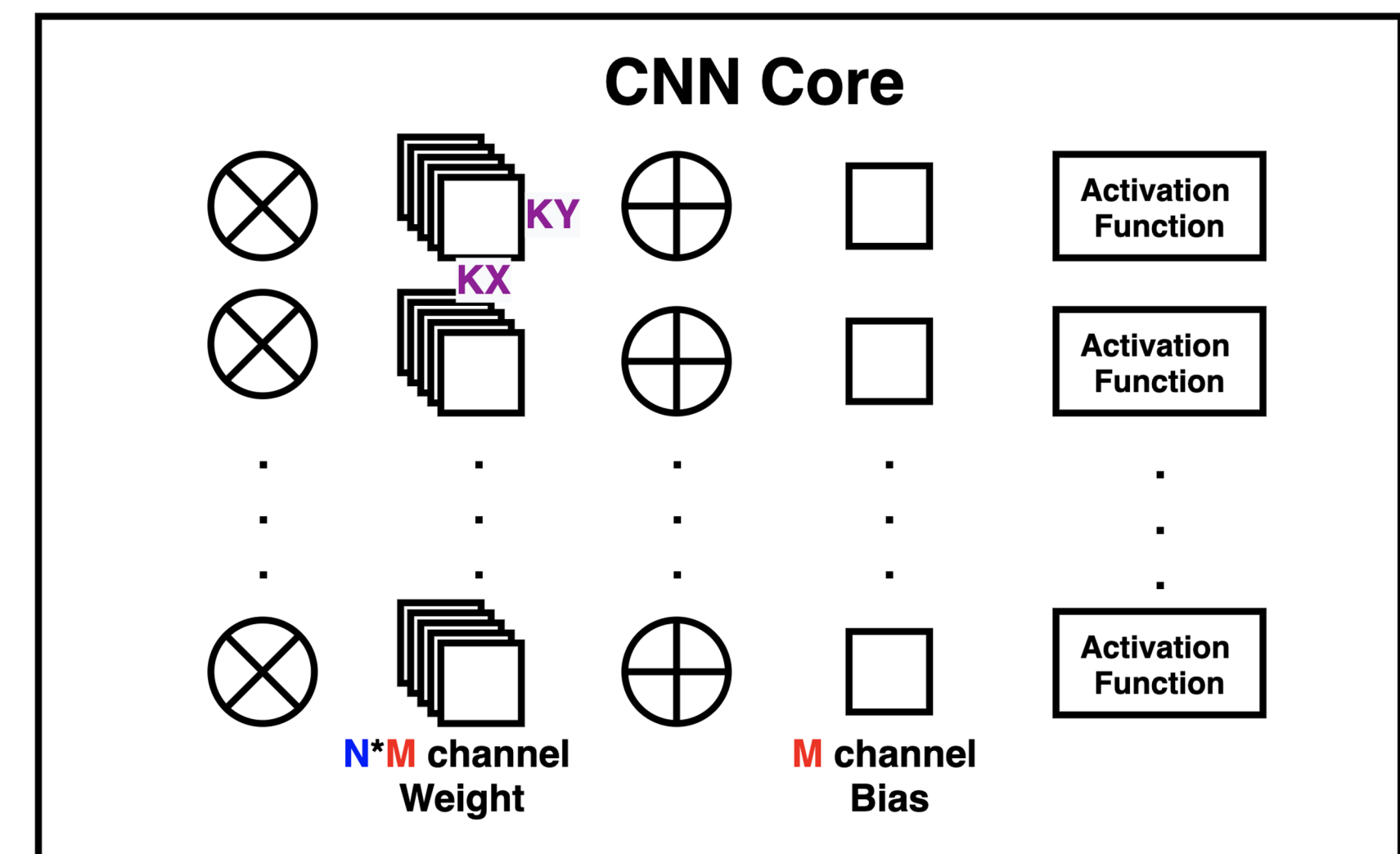
```
1 template<int L_X, int L_Y, int L_ZH, int L_ZOH>
2 void conv_T1
3 const ap_uint<W> w_in_weight(L_ZH)(L_X)(L_Y)(L_ZH),
4 const ap_uint<B> b_in_bias(L_ZOH)
5 {
6 // Parameter List
7 // BCh : output channel
8 // BCh : input channel
9 // KY : kernel Y
10 // KX : kernel X
11 for (och = 0; och < BCh; och++)
12 {
13     acc_result[och] = 0;
14     for (ich = 0; ich < L_ZH; ich++)
15     {
16         for (ky = 0; ky < KY; ky++)
17         {
18             for (kx = 0; kx < KX; kx++)
19             {
20                 mul = w_in_weight[ich][ky][kx] * acc_result[och];
21                 acc_result[och] += mul;
22             }
23         }
24     }
25 // added bias
26 wt_local[och] = acc_result[och] + bias[och];
27 //activation function (ReLu)
28 if (wt_local[och] < 0)
29 {
30     wt_local[och] = 0;
31 }
32 } // done
```

C/C++

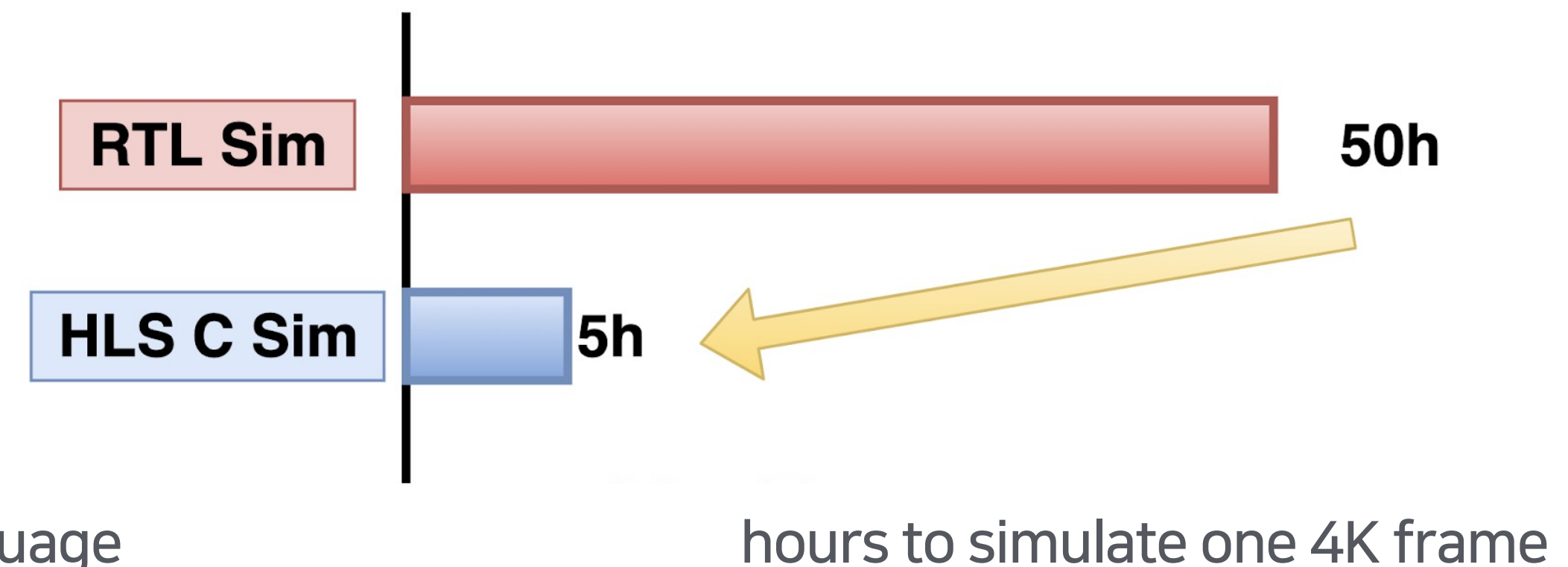
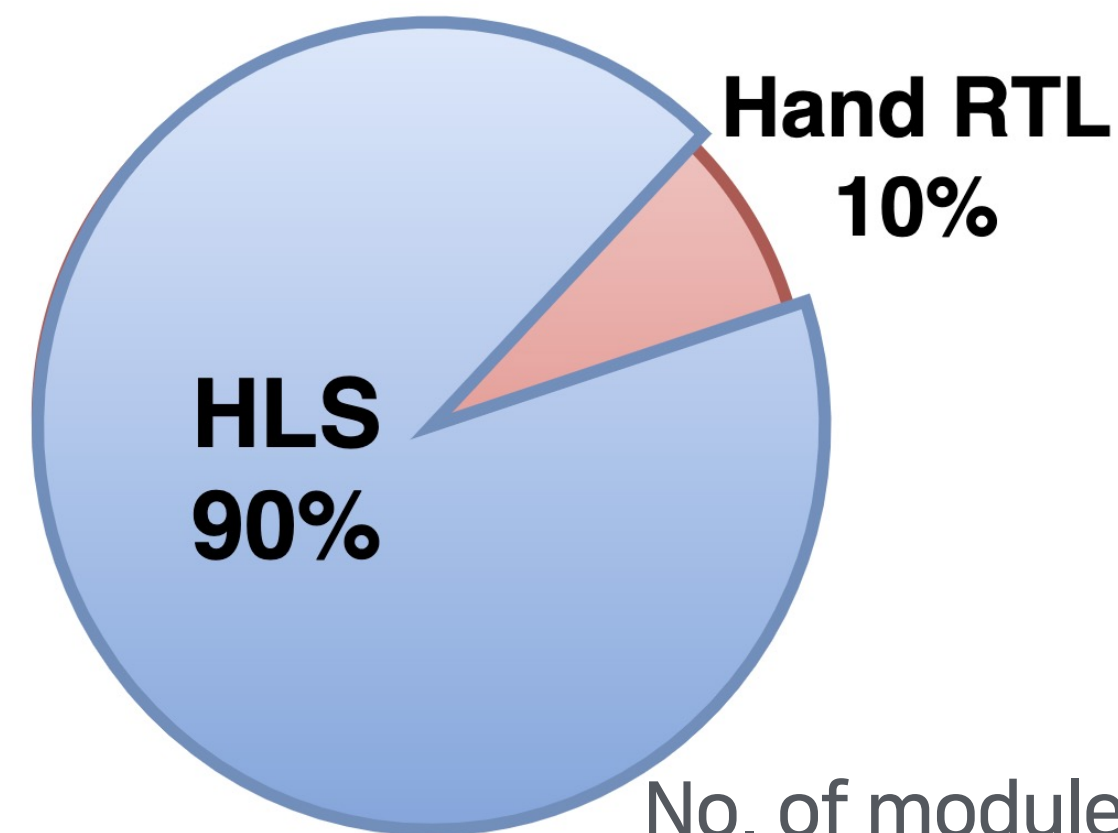
Directive



RTL 생성



대부분 블록 디자인에 HLS 사용



9x more modules



10x faster simulation

Timing closure가 쉬운 것이 많은 도움이 되었음!

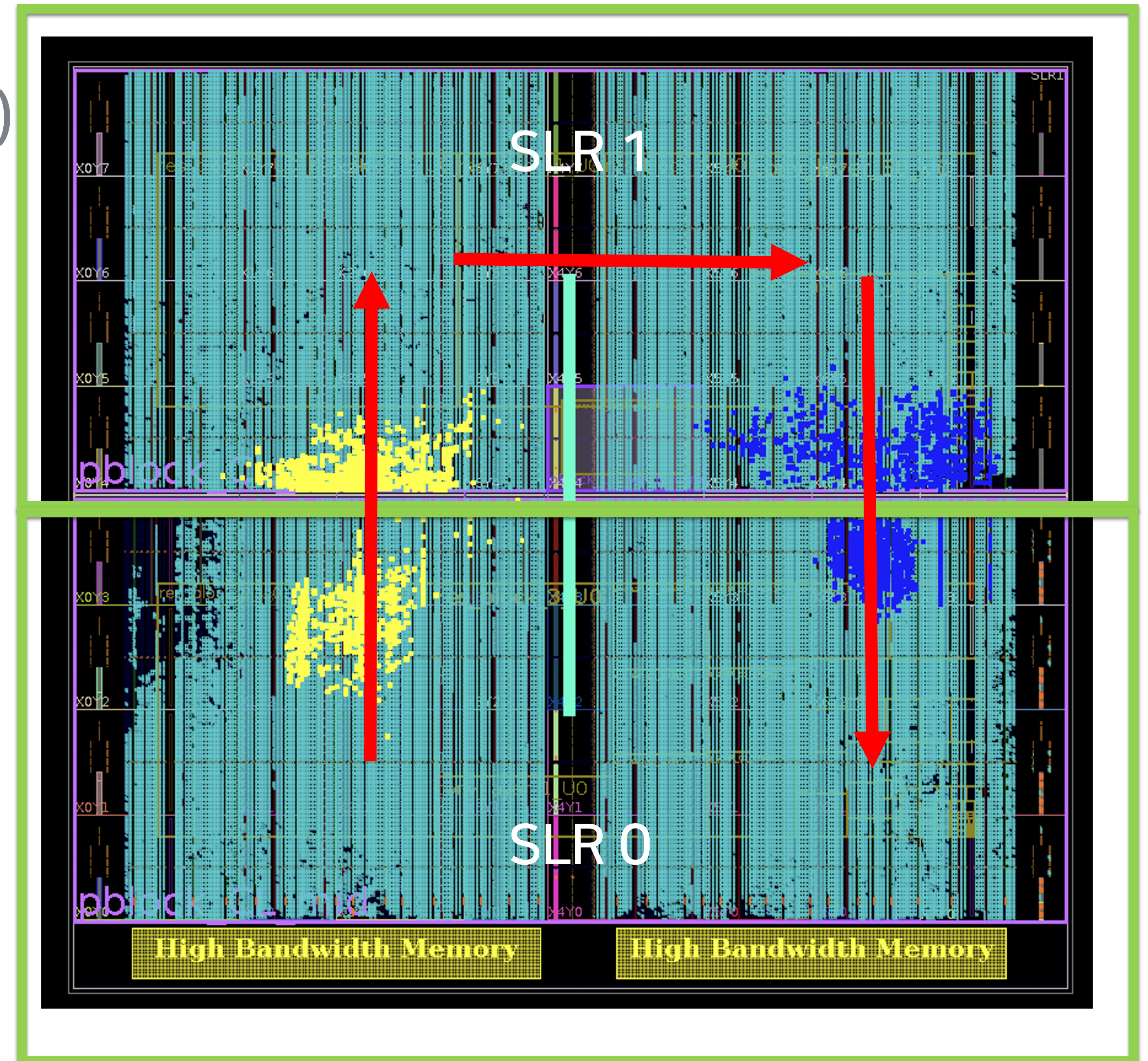
Deep Dive: Implementation

250MHz 를 향하여

- U50은 SLR0, 1 (Super Logic Region)로 구성
- Logic은 SLR0, 1에 반반씩
- Signal의 SLR crossing은 최소로
 - Circular placement
- SLR crossing에는 AXI register slices를 사용해서 timing closure

AXI register slices

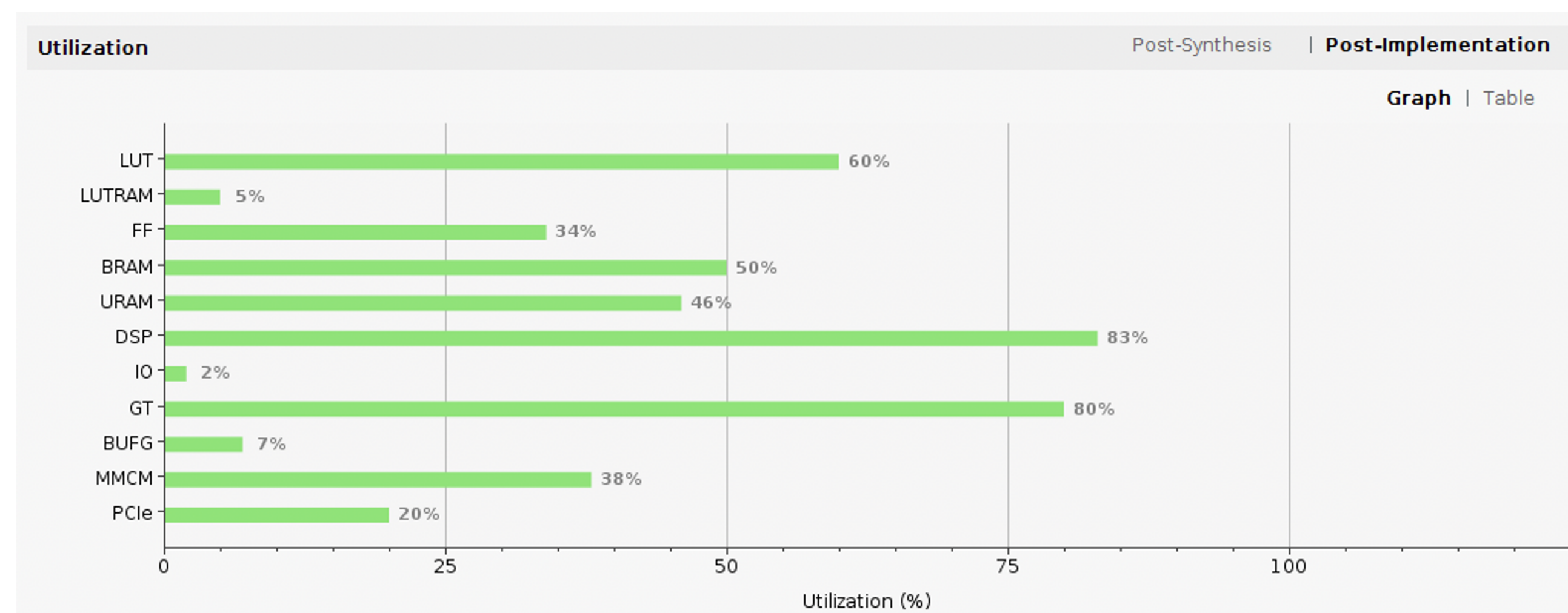
- **Yellow** : SLR0 to SLR1
- **Blue** : SLR 1 to SLR0



Utilization and Timing

• Utilization

Resource	Usage
LUT	60.16%
CLB Registers	34.03%
DSP	82.75%
BRAM	50.07%
URAM	45.63%



• 250MHz Implementation 성공!!!

Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 0.000 ns	Worst Hold Slack (WHS): 0.000 ns	Worst Pulse Width Slack (WPWS): 0.000 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 1942716	Total Number of Endpoints: 1942110	Total Number of Endpoints: 643223

All user specified timing constraints are met.

소모 전력

Total On-Chip Power: 50.015W

Results name: power_1

Environment | **Power Supply** | **Switching** | **Output**

Device Settings

Temp grade: extended

Process: typical

Environment Settings

Output Load: 0 pF [0 - 10000]

☐ Junction temperature: 57.671 °C

Ambient temperature: 55 °C

☒ Effective θ_{JA} : 0.75 °C/W [0 - 100]

Airflow: 250 LFM

Heat sink: medium (Medium Profile)

θ_{SA} : 0.655 °C/W [0 - 100]

Board selection: medium (10"x10")

Number of board layers: 12to15 (12 to 15 Layers)

θ_{JB} : 1.78 °C/W [0 - 100]

Board temperature: 25 °C [-55 - 100]

Option

Summary

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power: 50.015 W

FPGA Power: 43.419 W

HBM Power: 6.597 W

Design Power Budget: 60 W

Power Budget Margin: 9.985 W

Junction Temperature: 92.5°C

Thermal Margin: 7.5°C (9.1 W)

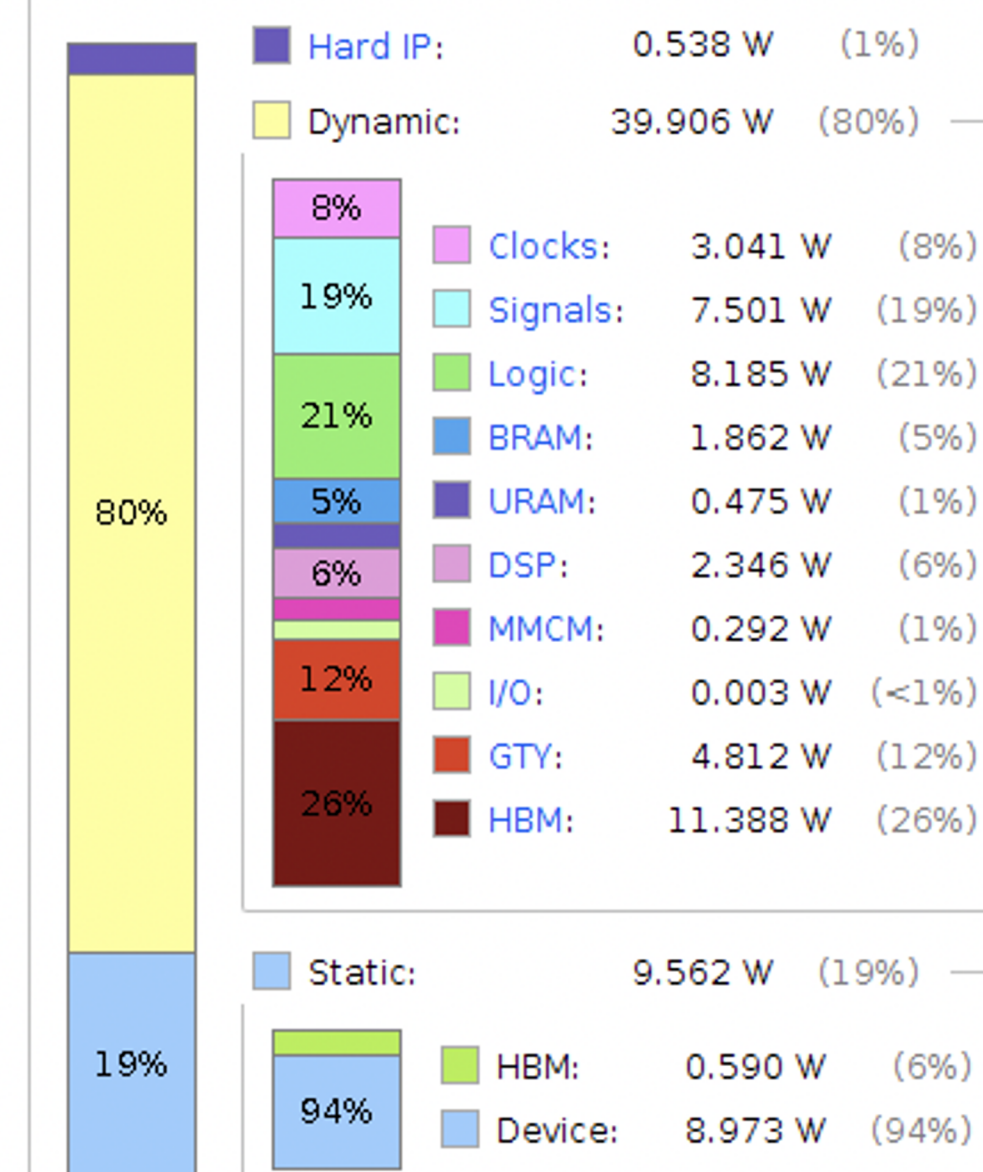
Effective θ_{JA} : 0.8°C/W

Power supplied to off-chip devices: 0 W

Confidence level: Medium

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

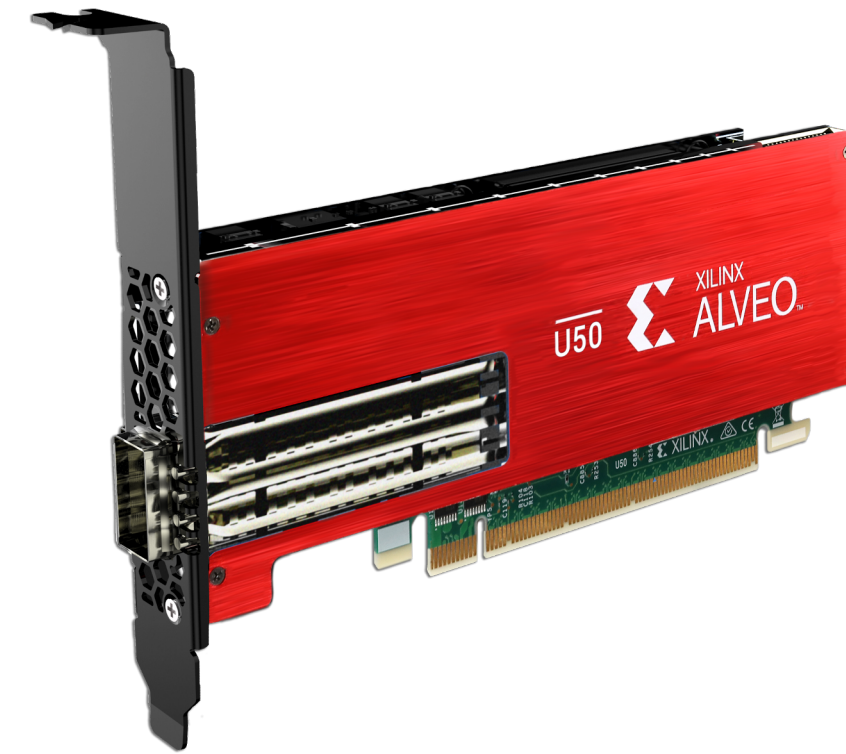
On-Chip Power



Summary

메모리 대역폭 사용량

720p => 2160p
@11FPS 처리



FPGA 내부

- U50 HBM total bandwidth: 316 GB/s
- 2 Gbps between the IP and HBM

1.1 Gbps

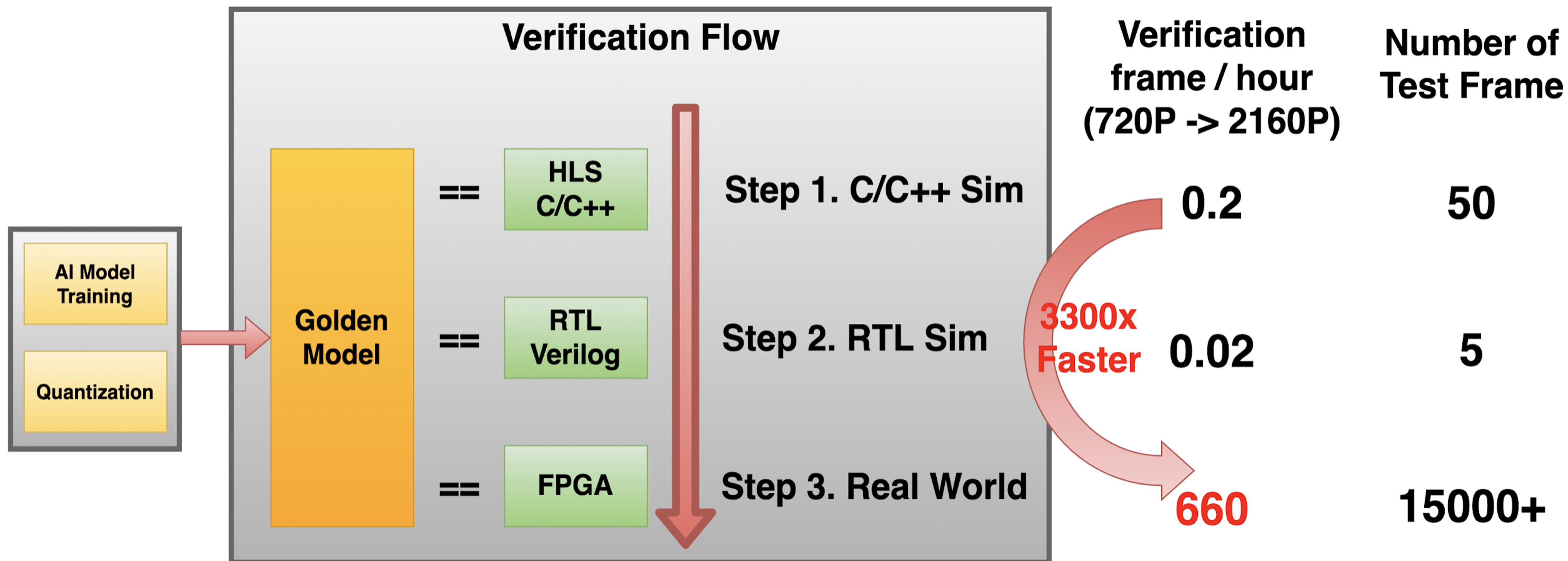
0.13 Gbps



외부 시스템 메모리

Deep Dive: 테스트

테스트 - Hardware



테스트 - Quality

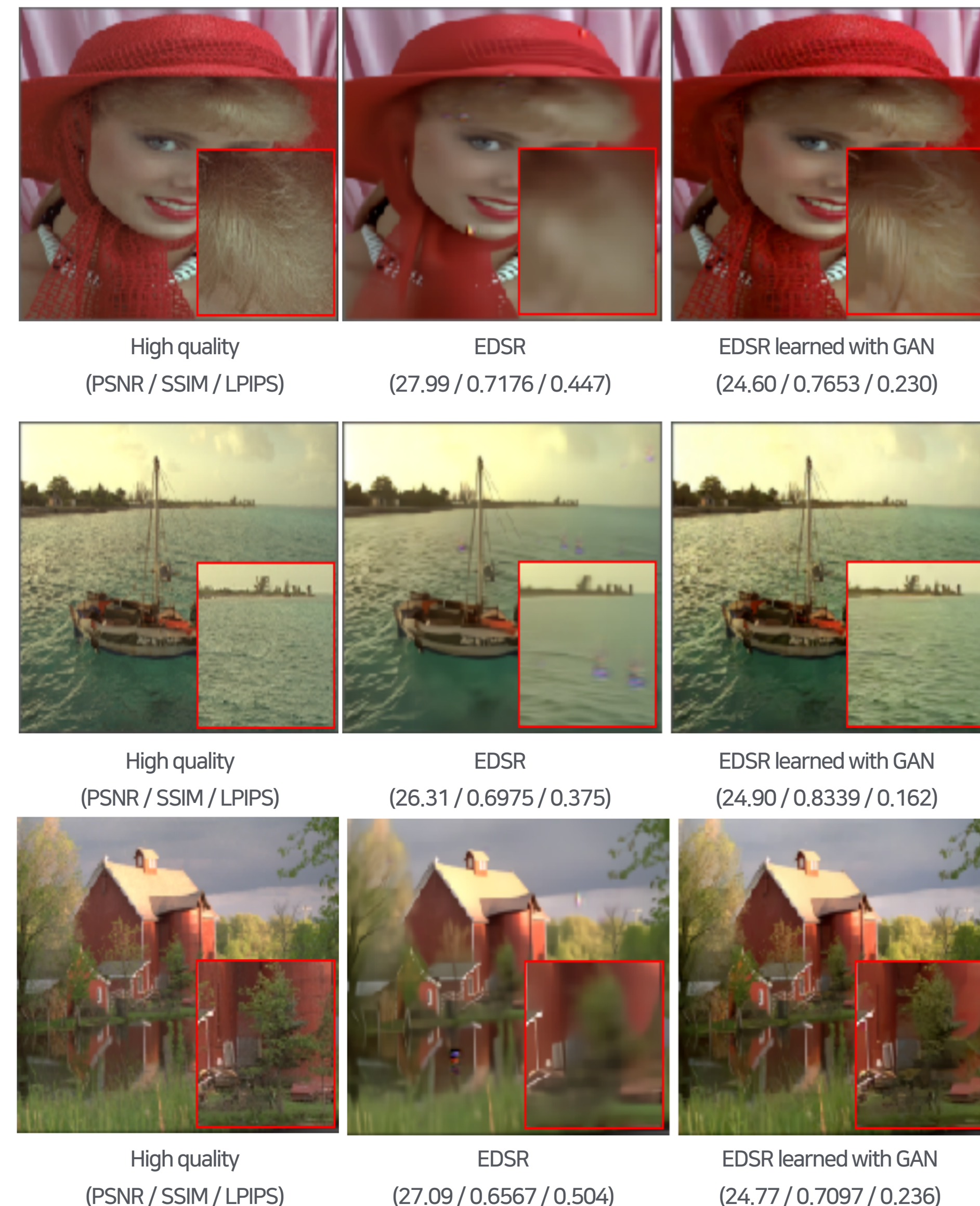
화질 평가 지표

- 객관적 평가 : PSNR, SSIM
- 주관적 평가 : LPIPS

PSNR, SSIM이 높다고 화질이 좋은 것은 아님.

주관적 평가 항목도 중요 지표로 사용.

LPIPS: Zhang et al., "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric," CVPR, 2018.



Deep Dive: 성능 지표

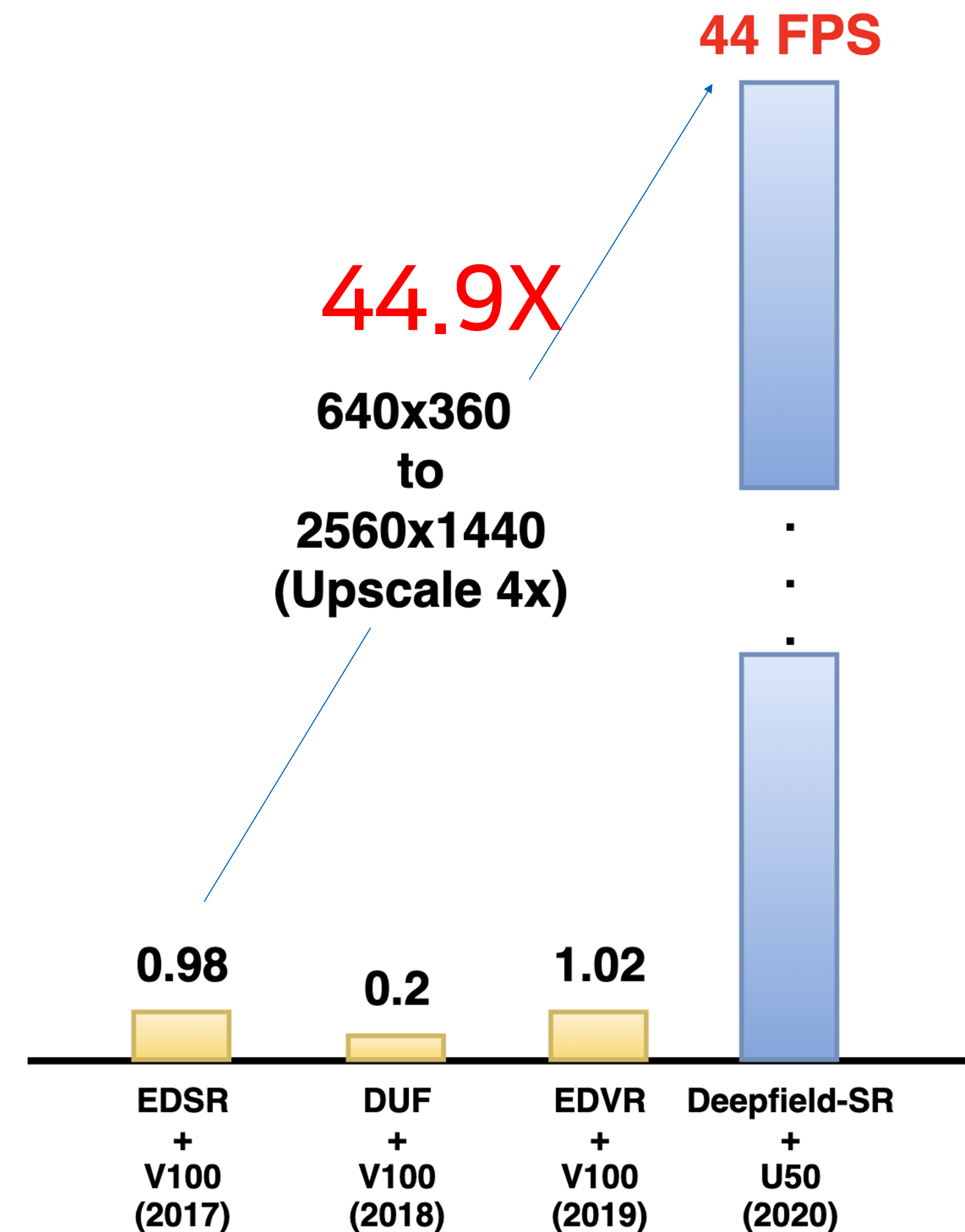


Frame Rate

360P to 1440P (4x) 수행시간

- 44 fps at a single U50 card.

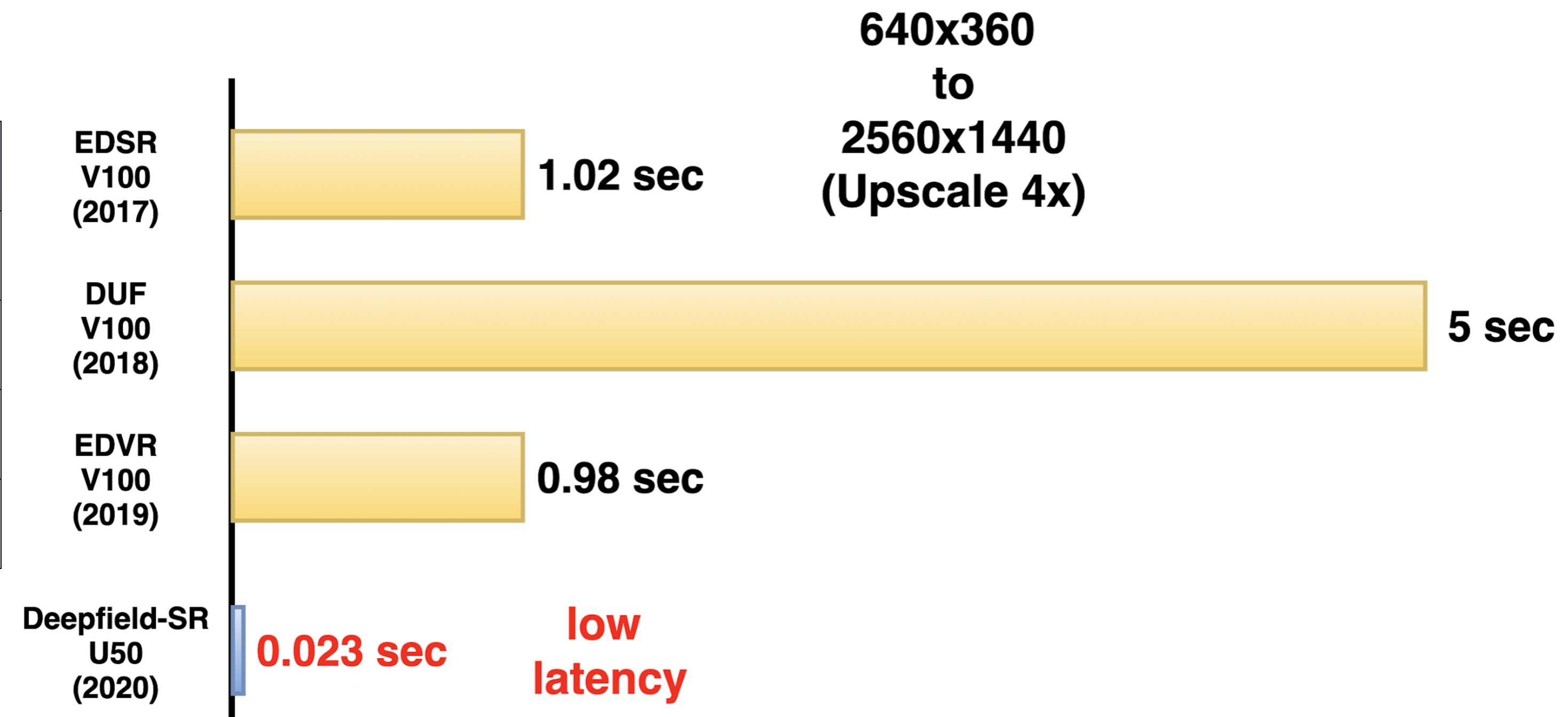
year	Network	Parameters#	FPS at NVidia V100	Longer(slower) than ours
2017	EDSR	43000 K	0.98 FPS	44.9x
2018	DUF	5800 K	0.2 FPS	220x
2019	EDVR	20600 K	1.02 FPS	43x



Latency

첫번째 출력이 나오는데까지 걸리는 시간 (360p to 1440p)

year	Network @ Device	Latency (sec)
2017	EDSR @ V100	1.02
2018	DUF @ V100	5
2019	EDVR @ V100	0.98
2020	Deepfield-SR @ U50	0.023



빠른 응답이 필요한 응용에 사용가능 (영상통화, 게임, 등)

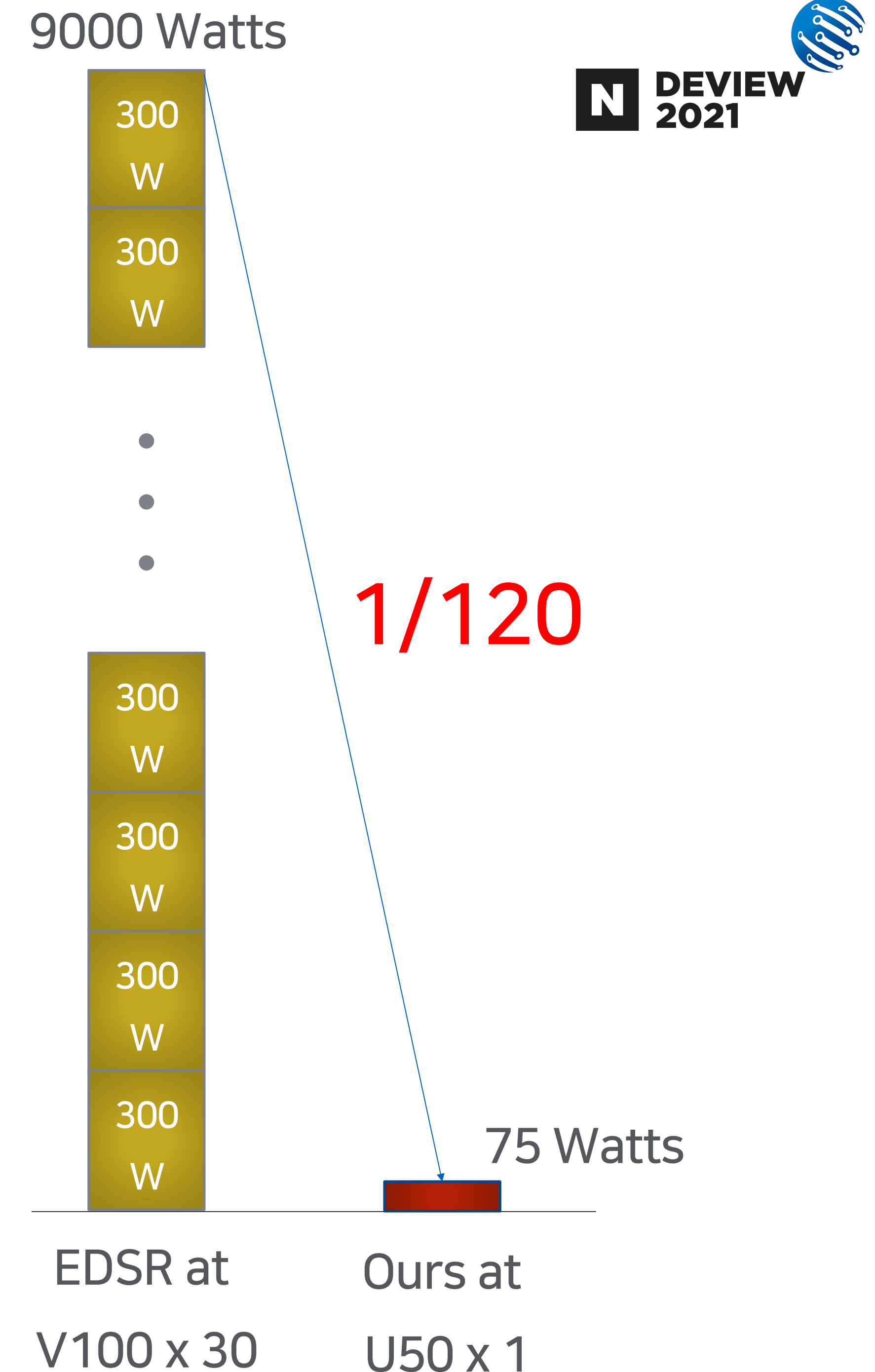
전력 소모

카드당 소모 전력

- NVIDIA V100: 300 Watts
- Alveo U50: 75 Watts

360p to 1440p @ 30 FPS 예상 소모 전력

- EDSR at V100 x 30: 9000 Watts
- DeepField-SR at U50 x 1: 75 Watts

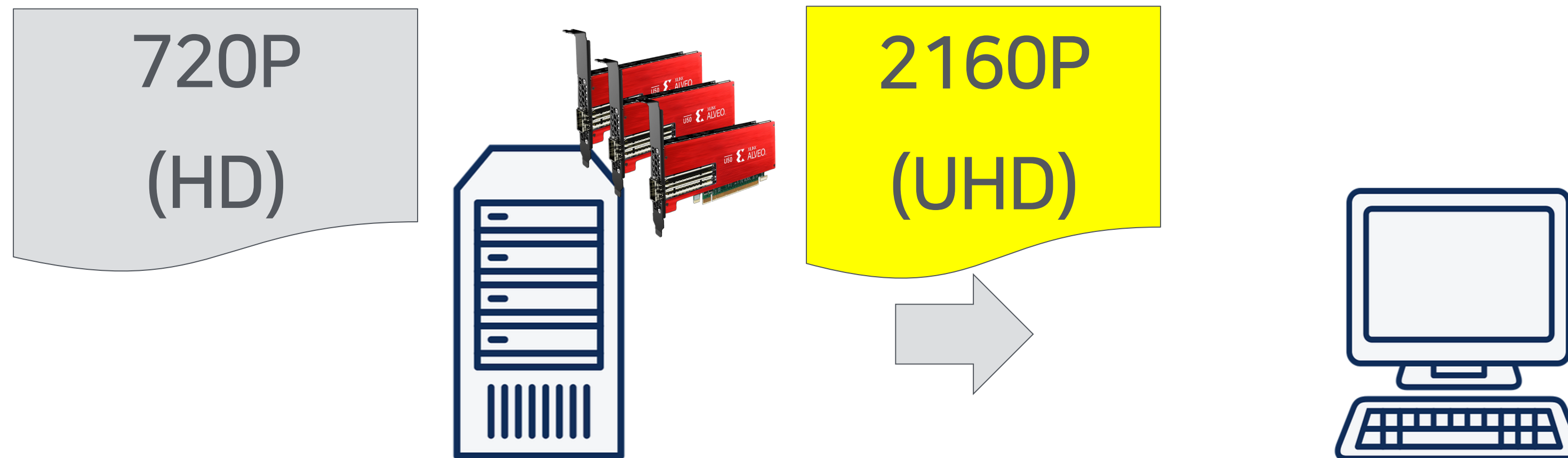


Demo – 비디오 스트리밍

Demo 1. 비디오 스트리밍

Demo environment

Data flow : Server → Client



Demo server
(U50 x3 with GPU)

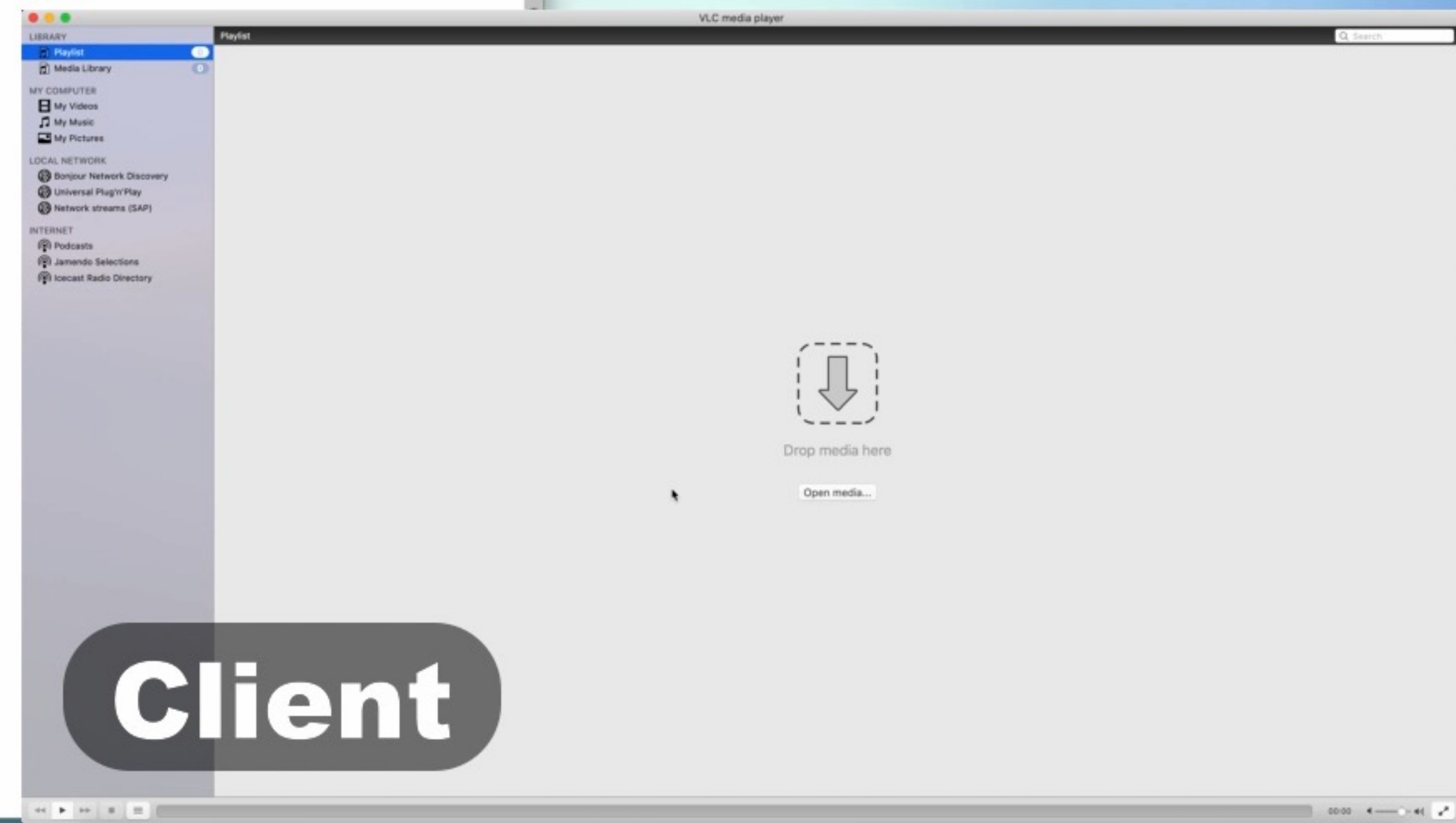
Client

root@demo01:/user2/deepfield

File Edit View Search Terminal Help

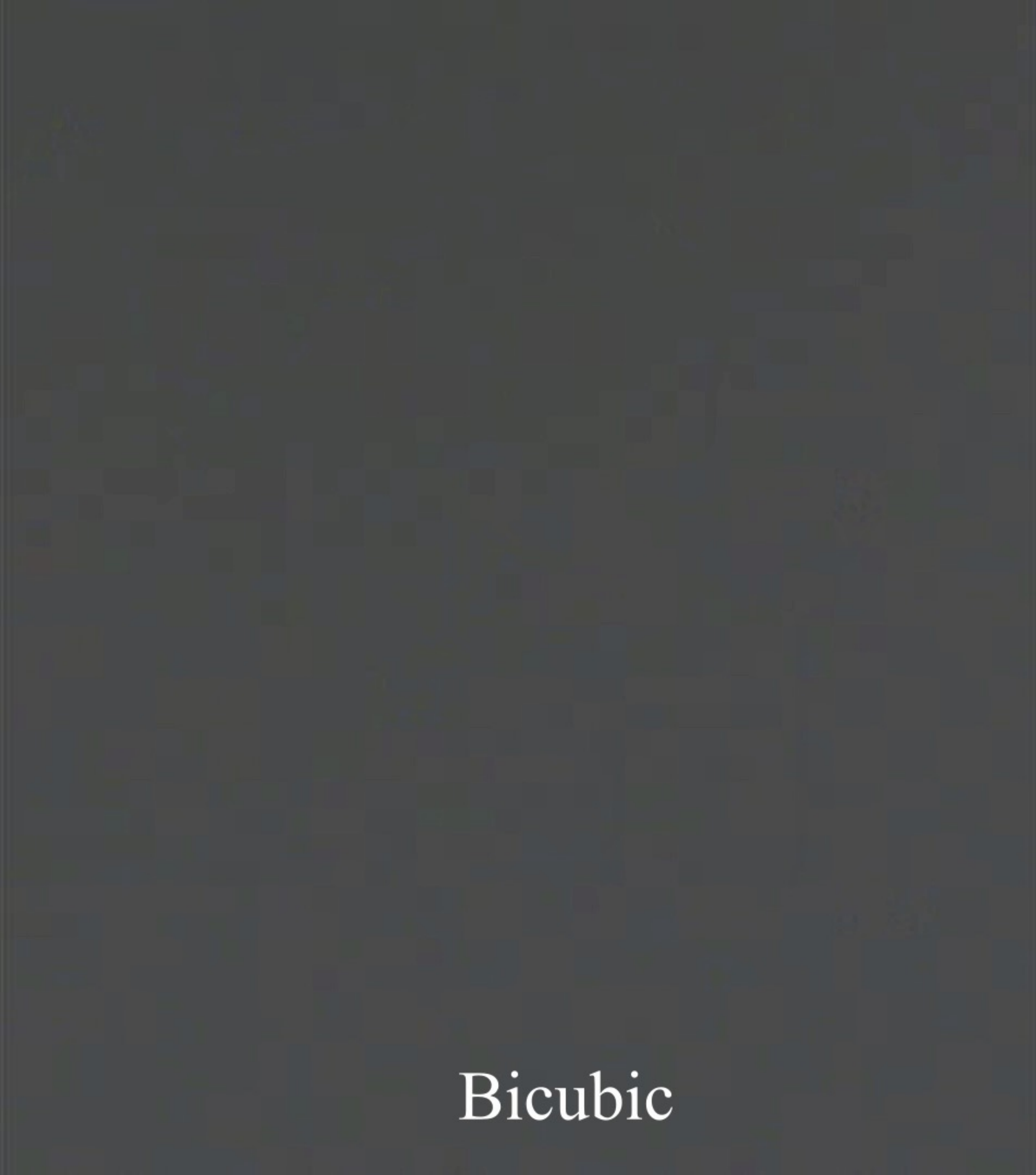
[root@demo01 deepfield]#

Host



Client

Demo - 화질 비교



Bicubic



DeepField-SR

Moving Forward



DeepField-SR 성능 개선

- De-blur 기능 추가



LR

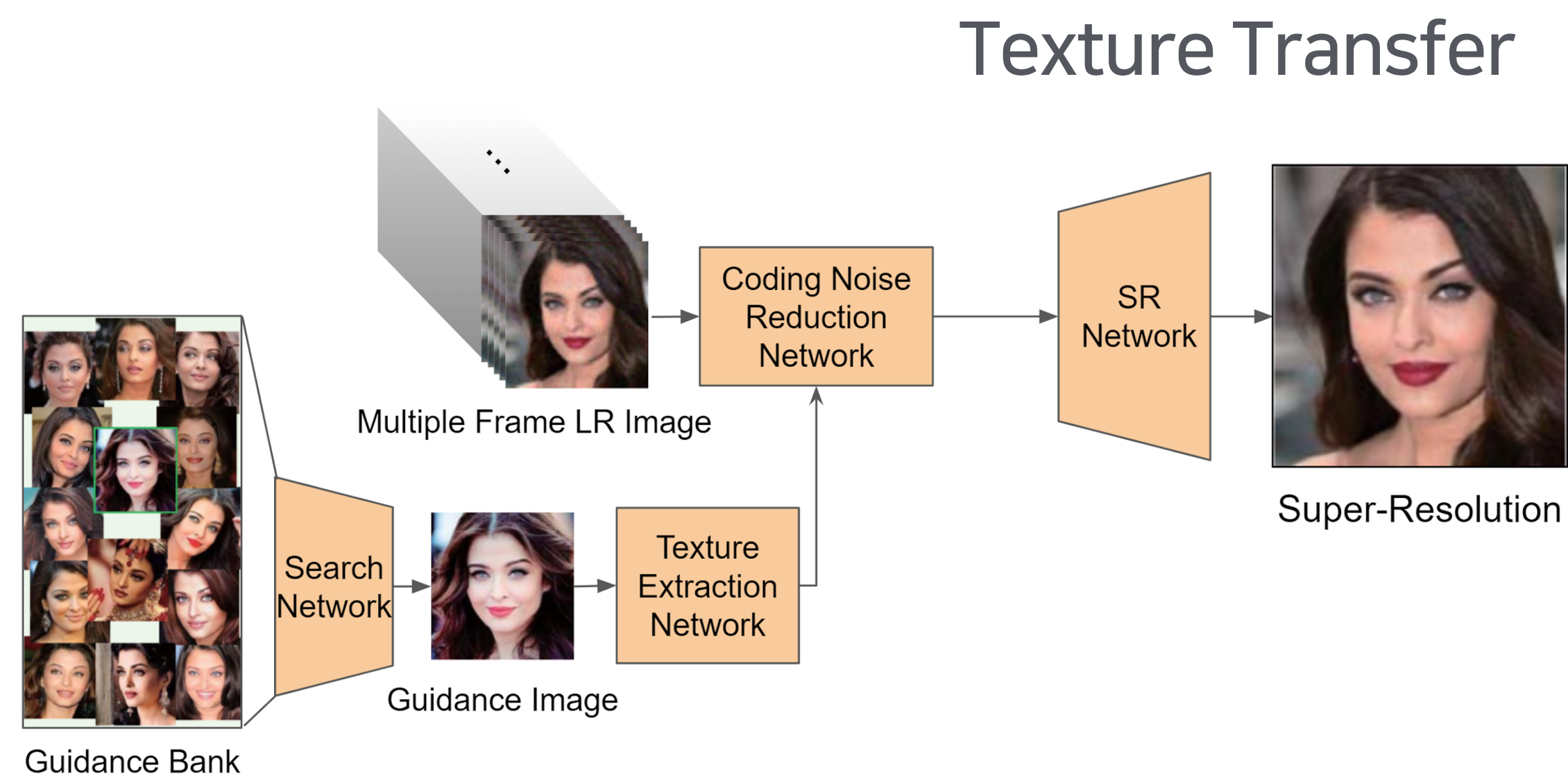
DeepField-SR



Enhanced SR
with de-blur

HR

- 좀 더 생생한 Texture



- 1080p 입력 지원

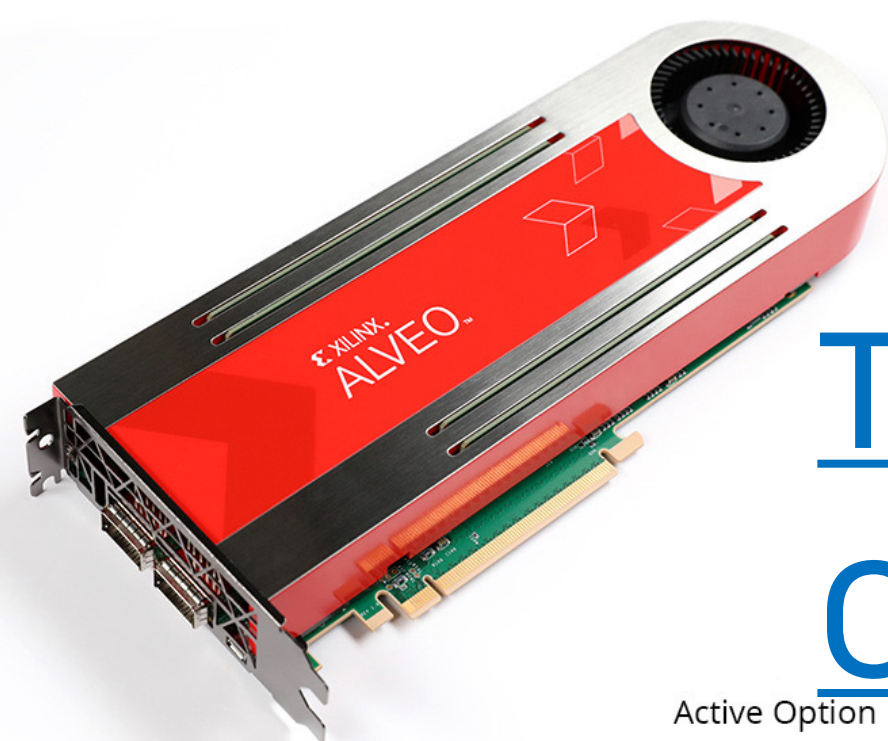
AV1 Encoder



Founding members

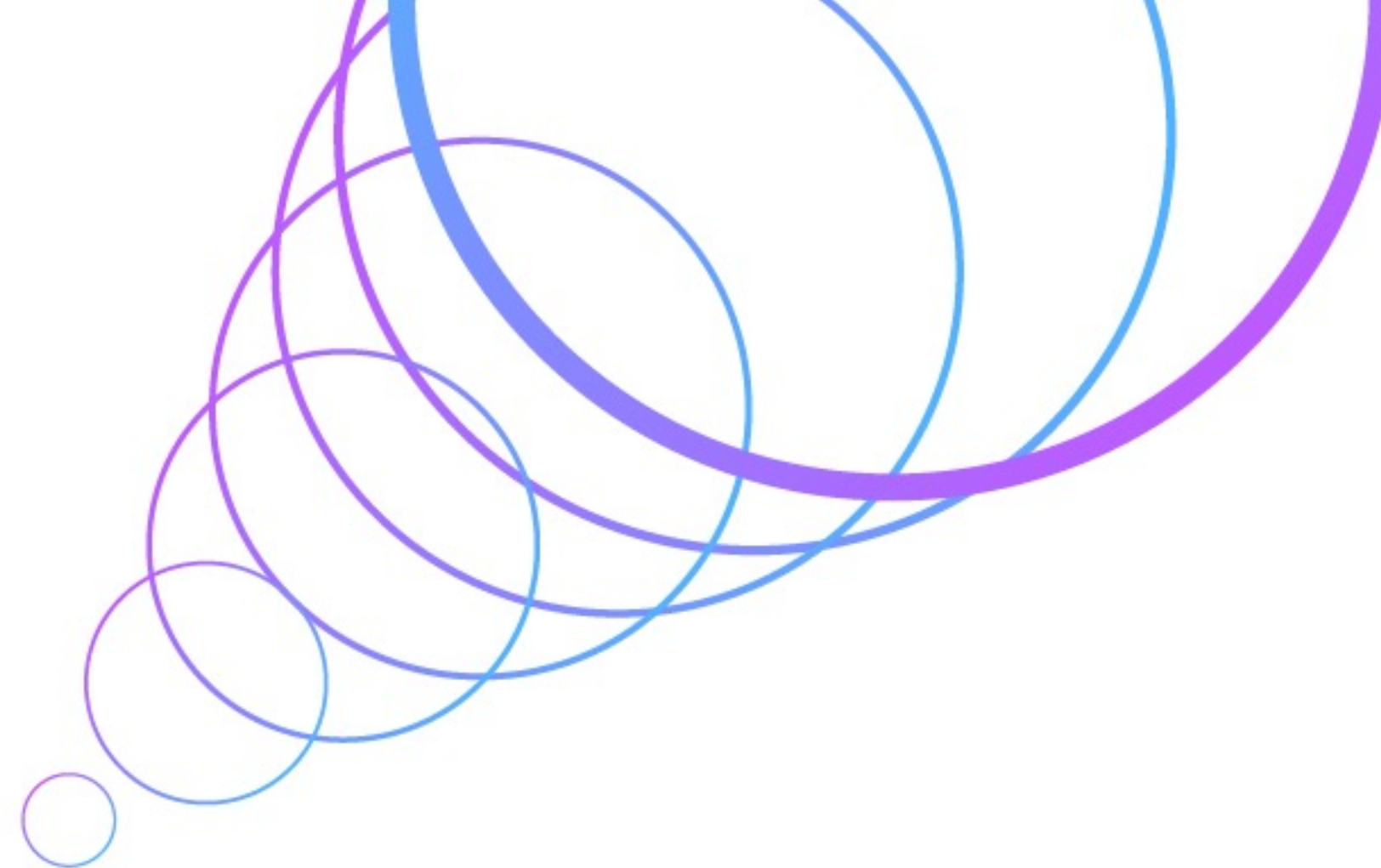
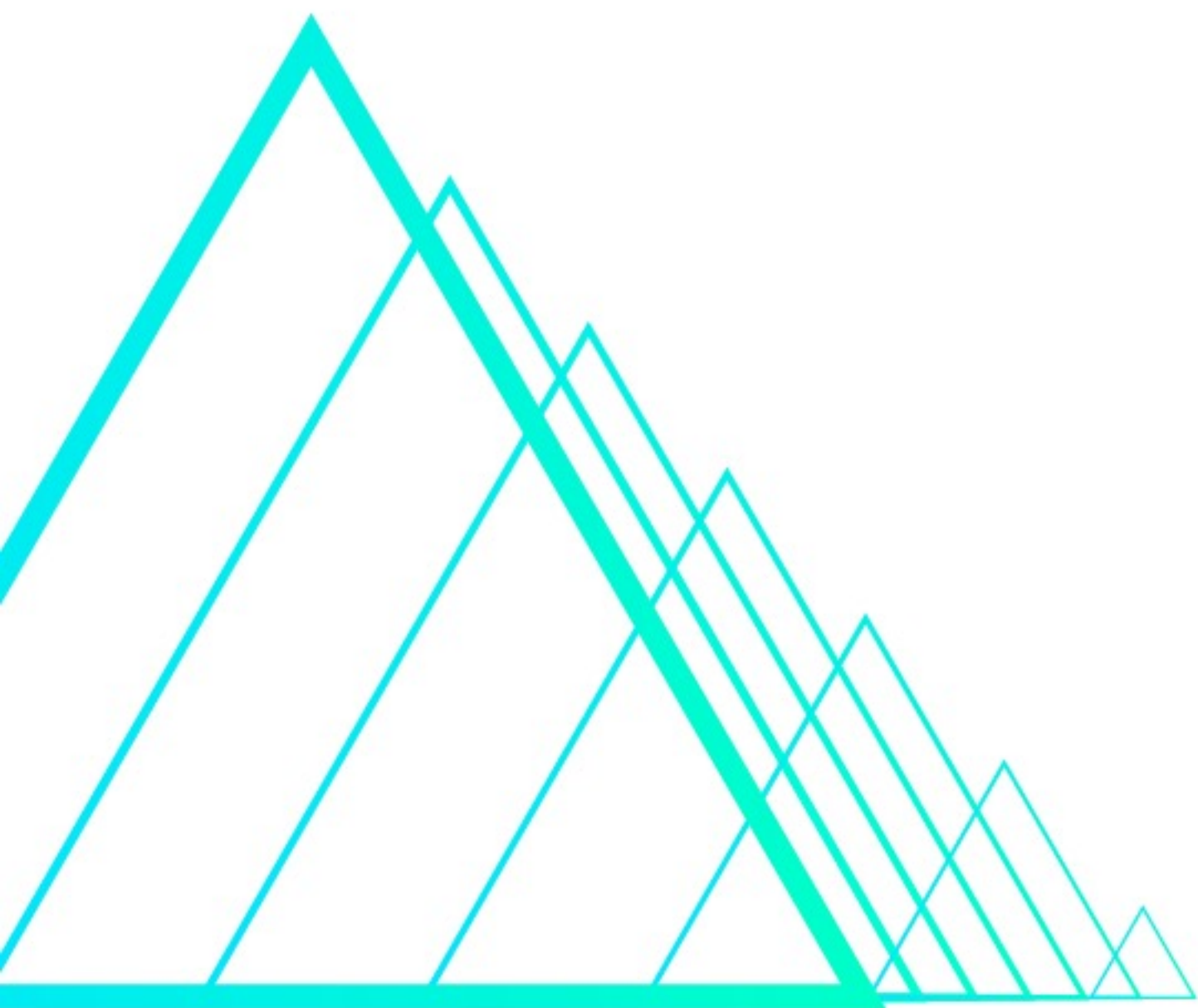


VS.



Active Option

Targeting 1920x1080p 60fps
On Xilinx U250 board



Thank You

