# Real Time Video Upscaling

Joshua Lau, James Garcia, Kunal Barde (B0)

# Problem Statement and Solution

- High definition video streaming often dependent on high bandwidth internet connections
    - 5Mbps or more
- Users with poor internet connections are forced to deal with low definition video streams to avoid lag or buffering
    - Lowest definition streams only consume 1.5Mbps << 5Mbps
- Upscaling a low definition video stream on the user’s end sidesteps the issue of a high bandwidth requirement for high quality video
-

# Requirements

- Be able to reconstruct a low-resolution video to a high-resolution video
  - Low-Definition Television Widescreen (426×240) to Full HD Widescreen (1920 x 1080)
  - Scaling factor of 4.5
- Be able to faithfully reconstruct the video
  - Netflix developed a metric for video quality: VMAF
    - scores range from 0 to 100
  - Our reconstruction will have a score of at least 80
    - 80 roughly corresponds to a “good” rating

# Requirements (cont’d)

- Be able to maintain the native frame rate of the video
    - Using videos with framerate of 24fps (standard for video streaming and movies)
    - Must maintain a throughput of no more than 41.7ms computation per video frame
- Be able to reconstruct the video in real-time
    - Do not want sound-after-picture as an issue
    - Hence, must maintain a latent delay of no more than 60ms
        - Adheres to the EBU Technical Recommendation R37 – 2007

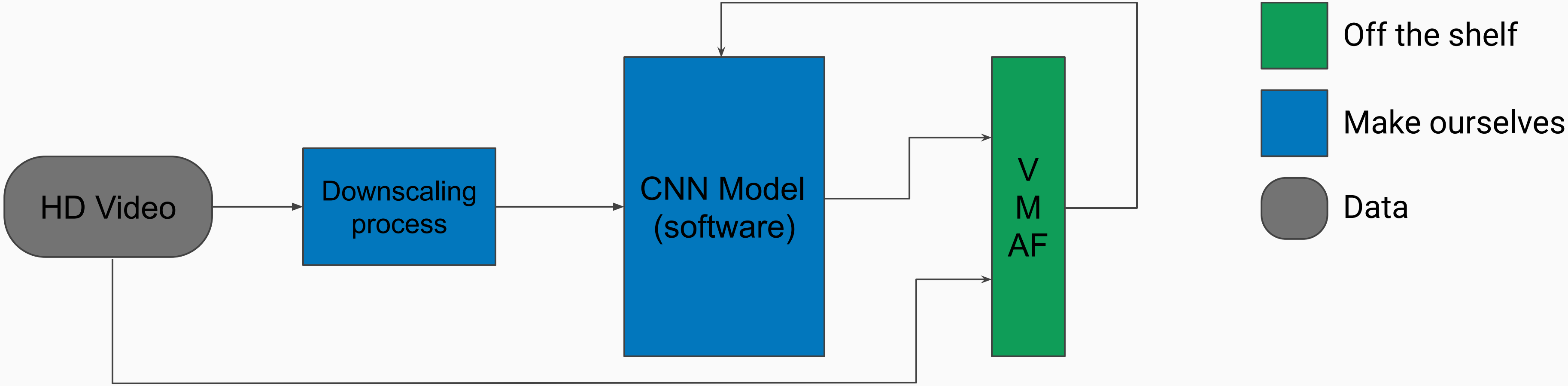
# Technical Challenges

- Training our network in software to faithfully reconstruct high resolution videos
- Sending video from the Ultra96's ARM core to the FPGA
- Maintaining throughput for video output from the FPGA
- Writing low-latency synthesizable Verilog to port our software model to the FPGA
- Finding an optimal pipeline architecture for our model
- Collecting user reviews for video reconstruction

# Solution Approach

- Software model of CNN to upscale videos
    - Using native high resolution videos which we downsample
    - Segments of video are used as batches to train the CNN
        - Using a negative VMAF score as a loss function
- Port software model onto FPGA to hardware accelerate it
    - Benchmark FPGA for throughput and latency
- Onboard ARM core sends low resolution videos to the FPGA
- FPGA applies forward pass of the CNN to output a reconstructed video

# Solution Approach - Training Graphic



HD Video

Downscaling process

CNN Model (software)

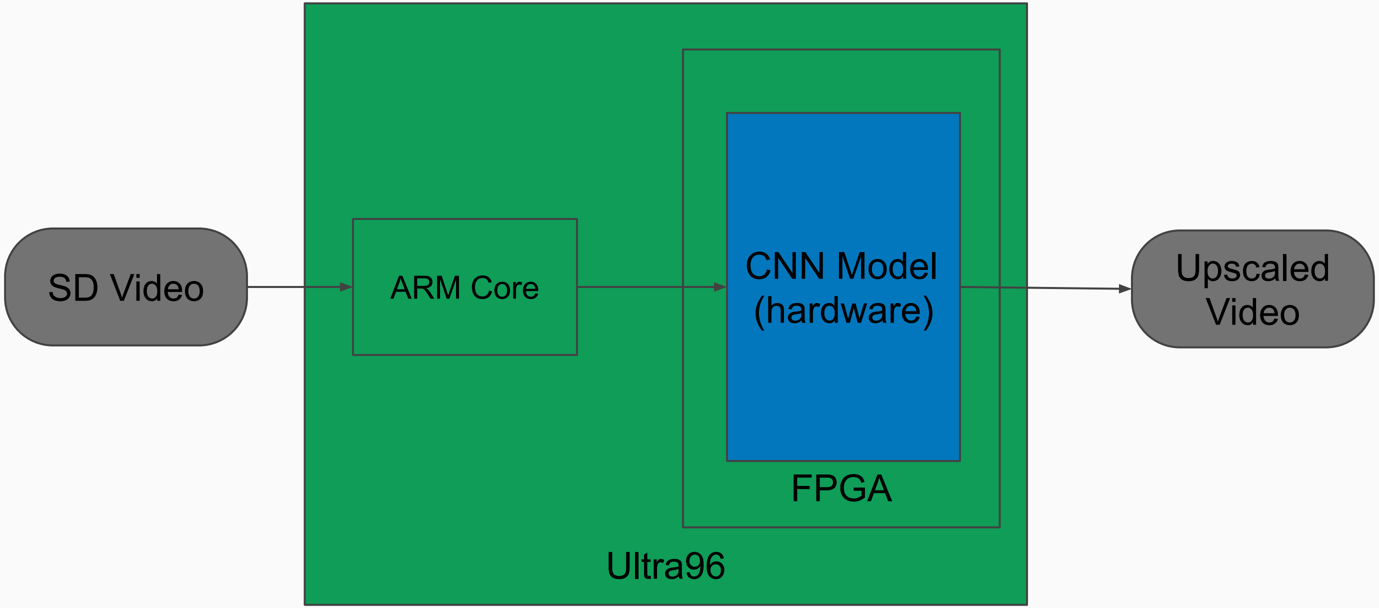
VMAF

🟩 Off the shelf

🟦 Make ourselves

● Data

# Solution Approach - Reconstruction Graphic



🟩 Off the shelf

🟦 Make ourselves

● Data

# Testing, Verification, and Metrics

- Downscale natively high-quality videos
  - Upscale downscaled version back to native quality using CNN
  - Compare original vs upscaled video using VMAF (Video Multimethod Assessment Fusion)
  - Form of full-reference method/model currently used by Netflix
- Combine objective with subjective test results
  - Use peer-reviewed/human feedback
  - Due to small dataset -> use as verification/sanity check, and not training
  - Use this as a heuristic to benchmark performance of VMAF

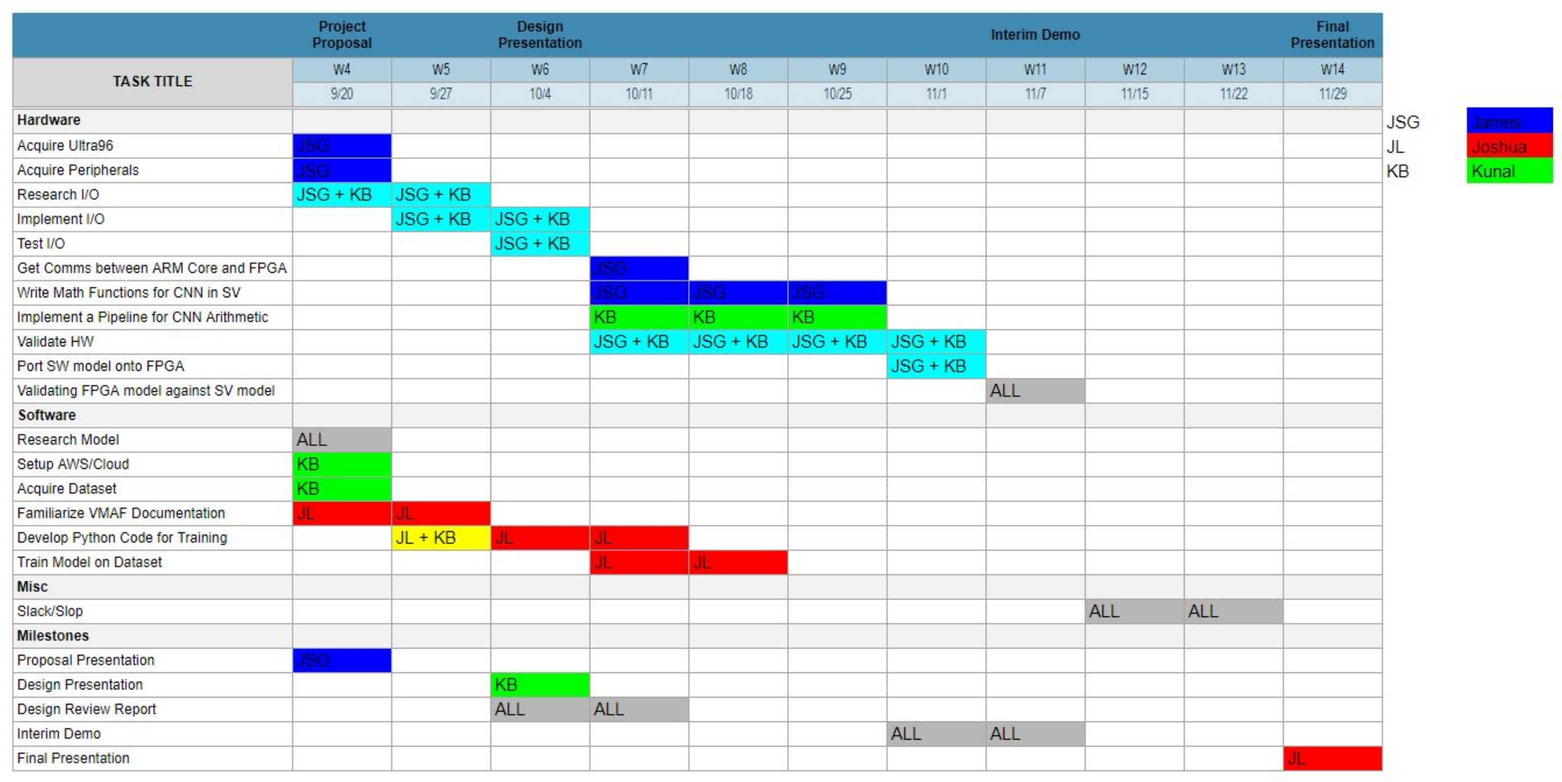
# Testing, Verification, and Metrics (cont’d)

- Benchmark the forward pass of the CNN on the FPGA for speed
  - Making sure to make the distinction between throughput and latency
  - How many cycles does it take?
  - How many ms of computation do we use per frame?
  - What is our latent delay?

# Tasks and Division of Labour

| **Tasks** | **Joshua** | **Kunal** | **James** |
|---|---|---|---|
| Algorithm development | X | X | X |
| Software model | X |  |  |
| Ramping on the Ultra96 |  | X | X |
| Hardware arithmetic |  |  | X |
| Hardware architecture |  | X |  |
| Porting & Integration | X | X | X |

# Schedule



| TASK TITLE | Project Proposal | | Design Presentation | | | | | Interim Demo | | | Final Presentation |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | W4 | W5 | W6 | W7 | W8 | W9 | W10 | W11 | W12 | W13 | W14 |
| | 9/20 | 9/27 | 10/4 | 10/11 | 10/18 | 10/25 | 11/1 | 11/7 | 11/15 | 11/22 | 11/29 |
| **Hardware** | | | | | | | | | | | |
| Acquire Ultra96 | JSG | | | | | | | | | | |
| Acquire Peripherals | JSG | | | | | | | | | | |
| Research I/O | JSG + KB | JSG + KB | | | | | | | | | |
| Implement I/O | | JSG + KB | JSG + KB | | | | | | | | |
| Test I/O | | | JSG + KB | | | | | | | | |
| Get Comms between ARM Core and FPGA | | | | JSG | | | | | | | |
| Write Math Functions for CNN in SV | | | | JSG | JSG | JSG | | | | | |
| Implement a Pipeline for CNN Arithmetic | | | | KB | KB | KB | | | | | |
| Validate HW | | | | JSG + KB | JSG + KB | JSG + KB | JSG + KB | | | | |
| Port SW model onto FPGA | | | | | | | JSG + KB | | | | |
| Validating FPGA model against SV model | | | | | | | | ALL | | | |
| **Software** | | | | | | | | | | | |
| Research Model | ALL | | | | | | | | | | |
| Setup AWS/Cloud | KB | | | | | | | | | | |
| Acquire Dataset | KB | | | | | | | | | | |
| Familiarize VMAF Documentation | JL | JL | | | | | | | | | |
| Develop Python Code for Training | | JL + KB | JL | JL | | | | | | | |
| Train Model on Dataset | | | | JL | JL | | | | | | |
| **Misc** | | | | | | | | | | | |
| Slack/Slop | | | | | | | | | ALL | ALL | |
| **Milestones** | | | | | | | | | | | |
| Proposal Presentation | JSG | | | | | | | | | | |
| Design Presentation | | | KB | | | | | | | | |
| Design Review Report | | | ALL | ALL | | | | | | | |
| Interim Demo | | | | | | | ALL | ALL | | | |
| Final Presentation | | | | | | | | | | | JL |

| | |
|---|---|
| JSG | James |
| JL | Joshua |
| KB | Kunal |