國立臺灣大學電機資訊學院電機工程學研究所

碩士論文

Graduate Institute of Electrical Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

於邊緣進行注重頻寬的推論：以實驗性方法分析影片超分析用於被壓縮影片之效果

Bandwidth-Efficient Inferencing at the Edge – An Experimental Approach to Analyze the Effect of VSR on Compressed Video

蔡予謙

Yu-Chien Tsai

指導教授: 黃寶儀 博士

Advisor: Polly Huang Ph.D.

中華民國 113 年 6 月

June, 2024

doi:10.5566/NTU2024XXXXX

國立臺灣大學碩士學位論文

口試委員會審定書

於邊緣進行注重頻寬的推論：以實驗性方法分析影片超分析用於被壓縮影片之效果

Bandwidth-Efficient Inferencing at the Edge – An Experimental Approach to Analyze the Effect of VSR on Compressed Video

本論文係蔡予謙君（R11921098）在國立臺灣大學電機工程學研究所完成之碩士學位論文，於民國 113 年 6 月 11 日承下列考試委員審查通過及口試及格，特此證明

口試委員：_____

（指導教授）

_____ _____

_____ _____

_____ _____

_____ _____

所　　長：_____

doi:10.5566/NTU2024XXXXX

doi:10.5566/NTU2024XXXXX

Acknowledgements

致謝

首先，我要感謝我的指導教授 Polly Huang 教授，她在我整個碩士學位期間給予了我無私的指導和支持。Polly 的專業知識、耐心和洞察力對於我的研究至關重要，沒有她的幫助，這篇論文將無法完成。不僅如此，Polly 也很注重做人處事的態度，包括做研究、與人相處、情緒管理和自我要求等等，讓我獲得比我的研究領域更寶貴的知識與經驗。另外，我也要感謝陳伶志博士和林崢茹教授，他們關鍵性的建議對於論文最後的完善有著很大的幫助。

再來，我要感謝實驗室的研究夥伴，Vincent、Kevin、Hao、GuoCheng，在這段時間裡，我們一起討論問題、分享想法，你們的幫助使得我的研究更加完善。

最後，我要感謝我的家人和朋友，特別是我的父母，他們一直以來的愛和支持給了我無盡的動力，對於任何經濟上的需求也是無條件的支持，讓我能夠堅持到底。

再次感謝所有在這段旅程中支持和幫助過我的人。

doi:10.5566/NTU2024XXXXX

doi:10.5566/NTU2024XXXXX

摘要

隨著深度學習的蓬勃發展，結合了強大深度神經網絡的行動裝置應用程式，特別是在擴增實境、混合實境和虛擬實境等等領域，正在蓬勃發展。然而，這些模型的計算需求往往超出了普通行動裝置的能力範圍，需要將神經網路運算移動至邊緣伺服器上。然而，這種方法會進一步衍生出延遲和網絡限制，這對邊緣運算輔助之移動擴增實境（Mobile Augmented Reality）中的實時性能尤其影響，特別是當上傳頻寬對於多數使用者來說是比較稀少的網路資源。

為了減輕頻寬的限制，本研究提出透過傳輸低畫質影片到邊緣伺服器後，再藉由超解析度（Super-Resolution）技術來增強影片品質與影像辨識（Object Detection）效能的方法。基於許多有關圖像質量對影像辨識的影響和超解析度在影片增強中之有效性的先前研究，我們探索了不同超解析度方法和影片壓縮率對使用邊緣運算做輔助之影像辨識的影響。我們的貢獻包括展現不同影片品質和超解析度水平上對於影片位元率與影像辨識性能之間的權衡，以及評估超解析度方法在低畫質影片上的效率。

關鍵字：超解析度、影片品質、影片編碼

doi:10.5566/NTU2024XXXXX

doi:10.5566/NTU2024XXXXX

Abstract

With the proliferation of deep learning, mobile applications integrating robust deep neural networks, particularly in Augmented Reality (AR), Mixed Reality (MR), and Virtual Reality (VR), are on the rise. However, the computational demands of these models often exceed the capabilities of commodity mobile devices, necessitating offloading to edge servers. Yet, this approach introduces latency and network challenges, crucial for real-time performance in edge-assisted Mobile Augmented Reality (MAR), especially when uplink bandwidth is a scarce resource for most users.

To mitigate bandwidth constraints, this study proposes transmitting low-quality (LQ) videos, augmented with super-resolution (SR) techniques to enhance object detection. Drawing upon prior research on image quality's impact on object detection and the efficacy of SR in video enhancement, we explore various SR methods and video compression rates' effects on object detection in a MAR scenario. Our contributions include delineating the trade-offs between video bitrate and object detection performance across different

doi:10.5566/NTU2024XXXXX

quality and resolution levels and assessing the effectiveness of SR methods on LQ videos.

Keywords: Super-Resolution, Video Quality, Video Coding

doi:10.5566/NTU2024XXXXX

Contents

doi:10.5566/NTU2024XXXXX

doi:10.5566/NTU2024XXXXX

List of Figures

doi:10.5566/NTU2024XXXXX

doi:10.5566/NTU2024XXXXX

List of Tables

doi:10.5566/NTU2024XXXXX

doi:10.5566/NTU2024XXXXX

Chapter 1 Introduction

With the rapid growth of deep learning, applications that rely on robust deep learning models are booming, particularly in domains like Augmented Reality (AR), Mixed Reality (MR), and Virtual Reality (VR). Naturally, there's a growing interest in deploying these applications on mobile devices. However, the resources of commodity mobile devices may not be sufficient to handle the heavy computational load of modern deep neural networks, even with proper optimization [16].

A common strategy in Mobile Augmented Reality (MAR) is to offload the intensive computation part of the system to edge servers [26]. However, the bottleneck in this method becomes evident in terms of latency and network conditions. Therefore, ensuring good downstream task performance with limited bandwidth and low delay is always critical for such edge-assisted MAR frameworks. One of the major downstream tasks in MAR is object detection. When offloading object detection computation, there's a significant demand for uplink bandwidth since we typically send images to the servers while only receiving labels or other critical information in return. Unfortunately, network traffic is often asymmetric, and uplink bandwidth tends to be particularly scarce.

So, we came up with one simple and naive idea that we can send low-quality (LQ) videos to save bandwidth at the cost of object detection performance. Additionally, by

doi:10.5566/NTU2024XXXXX

leveraging super-resolution, we can potentially enhance the video quality and recover object detection results. Some research has been conducted on how image and video quality affect object detection results [5, 7]. Other prior works [11, 17, 30] have indicated that machine learning super-resolution (SR) can improve user experience in a live-streaming system by enhancing video quality. In our research, we aim to provide a comparison of how different super-resolution methods and video compression rates specifically affect object detection results in edge-assisted MAR solutions. More specifically, we transcode videos into lower resolution and adjust the Constant Rate Factors (CRF) and Quantization Parameters (QP). Then, we directly apply SR to LQ videos and observe the quality difference and performance changes of object detection. In this study, the major contribution is: (1) we present a trade-off between video average bitrate and object detection performance across different video quality and resolution (2) we present the (in)effectiveness of applying well-performed SR methods on LQ videos.

The remaining chapters of this thesis are arranged as follows: background knowledge, including super-resolution, object detection, and video codec, as well as closely related prior works, will be covered in Chapter 2. Chapter 3 will detail the complete methodology, including data, tools, and experiment design. Chapter 4 will focus on the experiment results, along with additional analysis of the results. Finally, Chapter 5 will present a summary of the thesis and discuss future works.

doi:10.5566/NTU2024XXXXX

Chapter 2 Related Work

In our study, our objective is to apply super-resolution (SR) to videos before utilizing them as input for object detection inference, aiming to investigate the extent to which object detection can benefit from SR. Furthermore, the potential bandwidth savings achievable through the application of SR in video delivery scenarios remain unclear. To elucidate the primary contribution of this thesis, we will first delve into some background knowledge and related works in this chapter, focusing on two main aspects. Firstly, it is unsurprising that object detectors can be significantly influenced by image or video quality [5, 7]. As SR is essentially one of the many image enhancement methods, we will discuss prior studies [15, 21, 28] that address object detection tasks with the assistance of SR. Secondly, SR presents itself as a potential solution for video streaming services with limited uplink bandwidth. Therefore, it is crucial to explore efforts [11, 17, 30] that utilize SR to enhance the live streaming viewer experience.

2.1 Super-Resolution

Super-resolution (SR) generally refers to methods that recover high-resolution (HR) images from their corresponding low-resolution (LR) versions. Super-resolution has been applied to different targets such as satellite images, surveillance images, and medical im-

doi:10.5566/NTU2024XXXXX

ages. When the effectiveness of an SR algorithm is being tested, the process of generating LR samples from HR images is often formulated as follows:

$$\mathbf{I}^L = f_s(\mathbf{I}^H * \mathbf{k}) + \mathbf{n},$$

where HR images $\mathbf{I}^H$ undergo convolution with some blurry kernels $\mathbf{k}$ and are scale down using interpolation f_s with a specified scaling ratio s, resulting in LR samples $\mathbf{I}^L$. Some random sampled noise $\mathbf{n}$ might also be added in the process. Conversely, in reality, videos are encoded into various resolutions within video codecs, which is how we naturally generate LR videos. This will be discussed more in a later section.

Plenty of different super-resolution techniques have been developed throughout the past decade. For example, probability-based SR, reconstruction-based SR, and learning-based SR. In particular, neural networks in learning-based SR have consistently shown supreme results in the past few years. Therefore, we are interested in what these SR neural network models can deliver in terms of video quality. Super-resolution methods can be further categorized based on the number of LR images used to produce one HR image into Single Image Super-Resolution (SISR) and Video Super-Resolution (VSR).

2.1.1 Single Image Super-Resolution

SISR leverages various algorithms and models to learn the mapping between LR and HR image pairs. Deep learning-based approaches, particularly convolutional neural networks (CNNs), have gained prominence in recent years for their ability to effectively learn complex mappings and produce high-quality super-resolved images.

SRCNN [12, 13] is one of the earlier and more successful CNNs that learns the

doi:10.5566/NTU2024XXXXX

mapping between HR and LR images end-to-end with minimal preprocessing and post-processing. FSRCNN [14] was later proposed to expedite the process by eliminating the interpolation at the beginning of SRCNN and incorporating a deconvolution layer to learn the upsampling mapping at the end. Shi et al. [25] also proposed a famous sub-pixel convolutional neural network (i.e. pixel-shuffle) that efficiently upscales the image in the last layer. The sub-pixel convolutional neural network is then widely adopted by later SISR and VSR models.

2.1.2 Video Super-Resolution

Unlike SISR, VSR utilizes both temporal and spatial information as input. Existing deep-learning-based VSR approaches can be broadly categorized into two frameworks: sliding-window and recurrent. In the sliding-window framework, a fixed-size temporal window is usually used to select neighboring frames. Each frame undergoes multiple processing iterations, resulting in inefficient utilization of features and heightened computational costs. In the recurrent approach, models primarily leverage previously reconstructed high-quality frames to aid in the reconstruction of subsequent frames. Recurrent models usually adopt more sophisticated alignment methods to efficiently utilize the benefits from prior high-quality frames.

Chan et al. [8] compared different components in modern VSR approaches and proposed BasicVSR by leveraging common existing components (propagation, alignment, aggregation, and upsampling) and incorporating minor redesigns as needed. BasicVSR++ [9] was later proposed to further improve performance by implementing second-order grid propagation and flow-guided deformable alignment, allowing for more efficient and robust feature alignment.

doi:10.5566/NTU2024XXXXX

2.1.3 Super-Resolution with Object Detection as Downstream Task

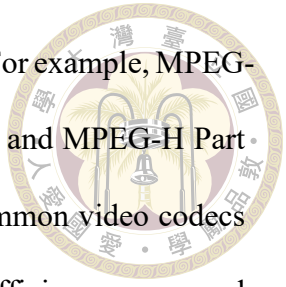
While super-resolution is primarily used to enhance user experience in terms of visual quality, cascading super-resolution before object detection isn't uncommon. The main idea is to compensate for the drop in object detection performance resulting from less-than-ideal image quality.

Haris et al. [15] proposed Task-Driven Super Resolution and trained SR explicitly together with object detection. They trained the entire system end-to-end with compound loss of detection loss and reconstruction loss to optimize the SR network, further fitting SR specifically to object detection. Other works [21, 28] implement SR explicitly prior to smaller object detection such as satellite images in remote sensing to resolve the multi-scale nature of those tasks. These models fit perfectly into specific image datasets; however, we consider these works to be too specialized for a mobile-AR case. In our work, SR models are tested without fine-tuning and optimization on the targeted dataset.

2.2 Video Streaming

Video streaming is essential for an edge-assisted mobile-AR solution. In this process, video codecs are utilized to encode and decode videos, preventing the waste of network resources. In addition to examining bandwidth consumption with video codecs, as mentioned in Section 2.1, we also utilize video codecs to efficiently and naturally generate low-quality videos. In reality, video streaming protocols and container formats can also impact network resource consumption due to overhead and other internal mechanisms. For simplicity, we ignore them in our study.

doi:10.5566/NTU2024XXXXX



Several major video codecs dominate the video delivery market. For example, MPEG-4 Part 10, also known as Advanced Video Coding or H.264 [23, 31], and MPEG-H Part 2, also known as High Efficiency Video Coding or H.265. Other common video codecs include VP9 and AV1. While H.264 may have lower compression efficiency compared to the other codecs mentioned, it remains the most popular due to its widespread compatibility. Most widely used desktop and mobile operating systems, along with the majority of popular content delivery networks and web browsers, offer support for H.264.

2.2.1 H.264/AVC Coding Scheme

H.264 is implemented with hybrid video coding, in which each input image is divided into macroblocks. There are two modes to encode these macroblocks, intra prediction mode and inter prediction mode. In intra-prediction, the prediction of target macroblocks relies solely on information from previously transmitted macroblocks within the same image. In inter-prediction, macroblocks are predicted based on the previously transmitted reference images. For each block (macroblocks' partition), a displacement vector is estimated and transmitted, indicating each block's corresponding position within the previously transmitted reference images.

2.2.2 Quantization Parameter

For further compression during encoding, quantization is performed in H.264 [19][10]. To be more specific, Quantization Parameter (QP) is utilized to determine the level of compression applied to each video frame. FFmpeg [1] offers two adjustable parameters for utilizing QP to compress an entire video: Constant Rate Factor (CRF) and Constant

doi:10.5566/NTU2024XXXXX

Quantization Parameter (CQP).

CRF adjusts QP to fluctuate among frames based on the level of motion within each frame while ensuring consistent overall quality across different videos. On the other hand, CQP maintains an identical QP value across all frames in a video. To clarify, both higher CRF and CQP values lead to increased compression rates, resulting in lower video quality and bitrate. It's important to note that these parameters cannot be adjusted simultaneously.

Other works [5] have discussed the influence of CRF on images' object detection results. They have shown that compressing videos with CRF can severely influence object detection accuracy regardless of the object detection model used. In this thesis, we aim to further investigate the influence of CRF and CQP by combining them with SR to evaluate the performance of SR methods across various quality levels of the same videos. We also aim to compare the bandwidth efficiency between CRF compression and resizing by resolution which have not yet been discussed in any other works.

2.2.3 Video Delivery System with Super Resolution

The idea of introducing SR to assist video delivery systems, especially live streaming, has sparked much discussion in recent years. Many prior works aim to deliver high-quality video streams with low latency by utilizing super-resolution neural networks, referred to as SR-enhanced live streaming.

Kim et al. [17] proposed LiveNAS that addresses limited uplink bandwidth by leveraging pre-trained SR deep neural networks (DNNs) on ingest servers with online training to improve performance. Chen et al. [11] proposed LiveSRVC, which addresses similar issues by applying SR alternately only on key frames. This approach greatly decreases the

doi:10.5566/NTU2024XXXXX

computational overhead of SR DNNs, while still allowing most frames to benefit from the SR-enhanced key frames. Wang et al. [30] investigated the bandwidth and quality gain in an SR-enhanced live streaming system across three aspects: different resolution pairs, model update frequency, and patch selection method where patches are used for training.

These works perform well in optimizing the trade-off among parameters such as the scaling ratio of SR, frequency of applying SR, uplink bandwidth conditions, and server computing power. However, they usually overlook video quality in terms of compression rate. In our study, our primary focus is on presenting the trade-off between video compression rate, resolution, and object detection performance, as both video compression rate and resolution significantly affect video quality and size.

2.3 Object Detection

Object detection is a computer vision task that involves identifying and localizing objects within an image or a video frame. The primary goal of object detection algorithms is to not only classify the objects present in the scene but also provide their precise locations through bounding boxes. Object detection finds applications in various fields, including autonomous driving, surveillance, image retrieval, and medical imaging.

In recent years, machine learning has achieved significant milestones in this field with the help of deep neural networks (DNNs), primarily convolutional neural networks (CNNs). Redmon et al. [24] proposed the famous object detector YOLO. YOLO is a single-stage detector that views detection problems as regression problems. It is extremely fast and proficient at learning generalizable representations of objects. Even today, YOLO and its successors are still considered state-of-the-art in object detection tasks.

doi:10.5566/NTU2024XXXXX

doi:10.5566/NTU2024XXXXX

Chapter 3 Experiment Design

The primary goal of the experiment is to investigate the extent to which object detection benefits from different qualities of SR-enhanced videos. As mentioned in Chapter 2, many SR problems are currently formulated with low-resolution images as blurry, interpolated versions of high-resolution images, which differs from how videos are transmitted over the internet. In this chapter, we will discuss the necessary tools and provide a detailed overview of the experiment process.

The process pipeline is briefly depicted in **Figure 4.1**. We utilize FFmpeg [1] to pre-process the raw videos into videos with the desired resolution and CQP/CRF settings. Next, the low-quality (LQ) video frames undergo processing through our selected SR networks. Depending on the SR networks chosen, the video frames may be processed into RGB or YCbCr channels beforehand. Subsequently, we provide YOLO [24] with both origianl high-quality (HQ) frames (set as groundtruth) and SR-enhanced HQ frames and calculate mAP@50 based on the bounding box results of the object detector.

3.1 Video Codec

To simulate the scenario of sending a video, we make use of a multimedia framework called FFmpeg [1]. FFmpeg provides libraries for encoding, decoding, transcoding,

doi:10.5566/NTU2024XXXXX

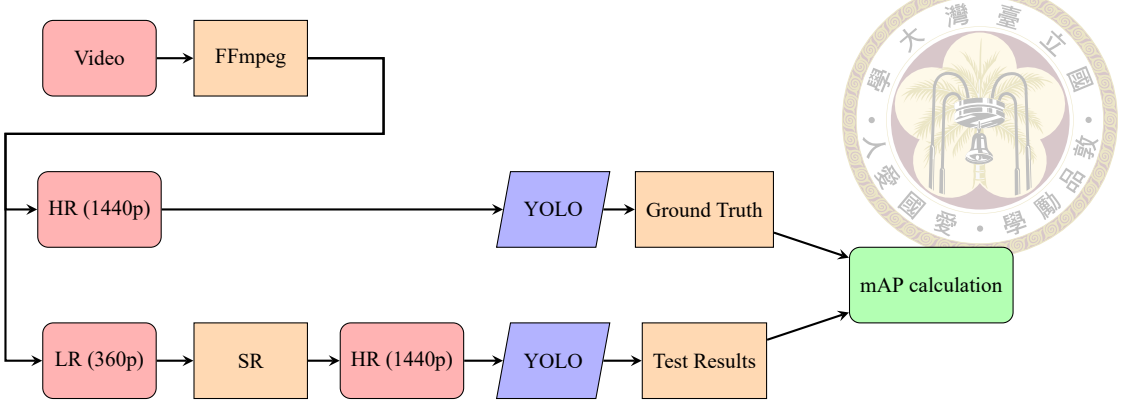


Figure 3.1: Experiment process pipeline

Flags	Values
i	<path to input videos>
c:v	libx264
preset	medium
vf	scale=<width>:<height>
crf	<a desired CRF value>
qp	<a desired CQP value>

Table 3.1: FFmpeg flags with corresponding input values.

streaming, and playing videos. It is also a command-line tool, making it highly flexible and scriptable for various multimedia processing tasks. In particular, we set FFmpeg Linux command flags as shown in **Table 4.1** to transcode original 4K videos.

The flag ”preset” controls the trade-off between the speed of encoding and compression efficiency. For example, setting ”preset” to ”slow” results in slower encoding but better quality at the same bitrate compared to setting ”preset” to ”fast”. In other words, ”preset” also affects quality while fixing CRF, CQR, or bitrate; therefore, we fix it as ” medium”. For other flags, we select an exhaustive combination of CRF and CQP values (0, 2, 5, 7, 10, 15, 20, 25, 30, 35, 40) to thoroughly investigate the results of the experiment. Regarding resolution pairs, we choose 1440p (1440x960) and 360p (360x240) to evaluate 4x super-resolution methods.

As mentioned above, the ”preset” flag is another trade-off you can make during encoding. Although we used the default setting and did not explore this further, Table 3.2

doi:10.5566/NTU2024XXXXX

Preset	Quality	Runtime (ms)	Bitrate (Mb/s)
ultrafast	CRF=0, 1440p	4.68	520
fast	CRF=0, 1440p	13.91	393
medium	CRF=0, 1440p	15.96	390
slow	CRF=0, 1440p	20.32	388

Table 3.2: Encoding runtime and bitrate for different ”preset” options.

Quality	Runtime (ms)	Bitrate (Mb/s)
CRF=0, 1440p	15.96	390
CRF=0, 1080p	9.92	236
CRF=0, 720p	5.77	122
CRF=0, 540p	4.18	74
CRF=23, 1440p	9.6	12.52
CRF=23, 1080p	6.23	8.23
CRF=23, 720p	4.3	4.25
CRF=23, 540p	3.34	2.66

Table 3.3: Encoding runtime and bitrate for different video qualities.

shows the average execution time per frame and the average bitrate per video using different ”preset” options. The encoding was performed using a standard 6-core Intel i5-12400 CPU. About the complexity of video transcoding, I also provide a simple comparison of the time required to transcode videos into different qualities, shown in Table 3.3. Unsurprisingly, transcoding videos into both higher resolution and lower compression levels results in longer processing times. Since delay is not the focus of my thesis, the details will not be discussed here. However, it is important to note the significant time differences when encoding videos into different qualities with my current overall settings.

3.2 Evaluation Metrics

Since our goal is to understand the trade-off between bandwidth and object detection results, mean Average Precision (mAP) is calculated for the evaluation of object detection.

Average Precision (AP) is a measure used to evaluate the precision-recall curve of a

doi:10.5566/NTU2024XXXXX

model. It is calculated by taking the average precision at different recall levels. The area under the precision-recall curve (PR curve) represents the AP. mAP extends the concept of AP to multiple queries or categories. It computes the AP for each query or category and then takes the mean of these individual APs. This provides an overall assessment of the model's performance across all queries or categories. mAP is a widely used performance metric in information retrieval and object detection tasks. To be more precise, in our study, we use mAP@50, where a true positive case (correct label prediction) is counted if the intersection over union (IOU) of a predicted bounding box and the ground truth bounding box exceeds 50 percent.

We also evaluate the Peak signal-to-noise ratio (PSNR) of SR-enhanced HR frames to assess whether the results of PSNR align with mAP. PSNR is calculated using the following formula:

$$\textbf{PSNR} = 10 * log(\textbf{MAX}^2/\textbf{MSE}),$$

where **MSE** stands for Mean Square Error and **MAX** is set to 255 for 24 bits RGB.

3.3 Video Enhancement

As mentioned earlier, we utilize FFmpeg to obtain LQ videos. These videos are then extracted into individual frames using OpenCV [4] before proceeding to the SR methods. In **Chapter 2.1**, we briefly introduce different types of SR methods. For this experiment, we select FSRCNN [14] to represent single-image super-resolution (SISR) and BasicVSR++ [9] to represent video super-resolution (VSR). Additionally, bicubic interpolation is tested for comparison with more advanced SR methods.

doi:10.5566/NTU2024XXXXX

3.3.1 Video Super-Resolution

For video super-resolution, we utilize the official BasicVSR++ pre-trained weights due to the complexity and lack of resources to train the model ourselves. The model has been trained and tested on the REDS [22] dataset, achieving a testing PSNR of approximately 32.4. It is fully open source and available on MMEditing [20]. While our experiment primarily focuses on other aspects rather than latency, it's essential to note that we are not pushing the model to its maximum capacity without any time limitation. Given that BasicVSR++ is not designed for real-time VSR solutions, we limit the maximum sequence length to 5 frames. This ensures that the model cannot utilize information much beyond the neighboring frames. For FSRCNN, the model is trained on the T91 image dataset and tested on Set5 [6], achieving a PSNR of approximately 30.55. The training settings closely follow those described in the original paper.

3.3.2 Video Interpolation

Bicubic interpolation is often used to enlarge images and usually performs better among other common interpolation methods such as bilinear or nearest-neighbor interpolation. In bicubic interpolation, a bicubic polynomial is fitted to a 4x4 grid of neighboring points. This allows for smoother and more accurate interpolation, particularly useful for scaling images.

The complexity of bicubic interpolation compared to previous SR methods is shown in Table 3.4. Since these three video enhancements differ significantly in complexity, we believe they provide a good scope for roughly understanding the time and computational resources required, in addition to considering the trade-offs between bandwidth and object

doi:10.5566/NTU2024XXXXX

Methods	Parameters	Runtime (ms)
Bicubic		0.05
FSRCNN	12809	5
BasicVSR++	7.7M	180

Table 3.4: The time it takes to process one 480x270 video frame with a single Nvidia [3] RTX 3080 GPU and the parameter count of the models.

detection performance.

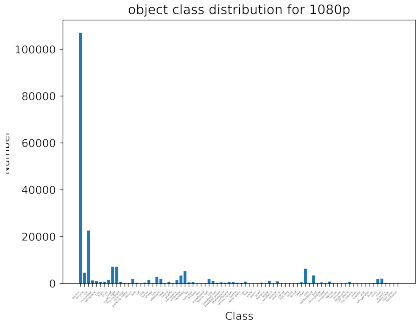
3.4 Object Detection

After obtaining SR-enhanced HQ frames, both the original HQ frames and SR-enhanced HQ frames are directly fed into YOLO [24] for object detection. Specifically, YOLOv5x6 is utilized for maximum object detection accuracy. The results of the original HQ frames transcoded with CRF set to 0 (no compression, maximum quality) are considered as ground truth. Conversely, the results of SR-enhanced HQ frames are used in mean Average Precision (mAP) calculation, based on the ground truth labels from the zero compression original HQ frames.

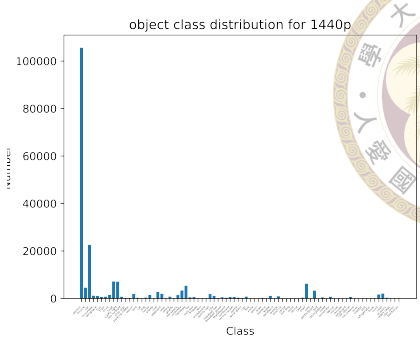
3.5 Dataset

The purpose of selecting an appropriate dataset for our experiment is to simulate a mobile AR use case. We chose the Inter4K dataset [27], which consists of 1000 ultra-high-resolution (4K) 60 fps clips, each lasting 5 seconds, as our primary dataset. Inter4K is primarily utilized for video super-resolution, offering 4K resolution, which enables users to generate various lower-resolution videos. Additionally, Inter4K categorizes its videos into six different categories, providing a diverse range of scenes. Furthermore, part of the dataset is captured with mobile phones across different locations, aligning with the

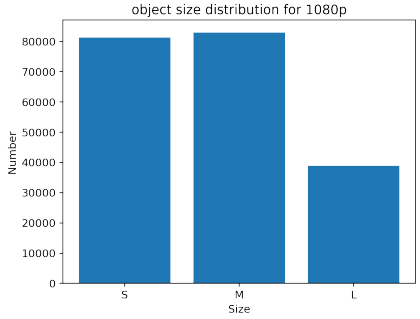
doi:10.5566/NTU2024XXXXX



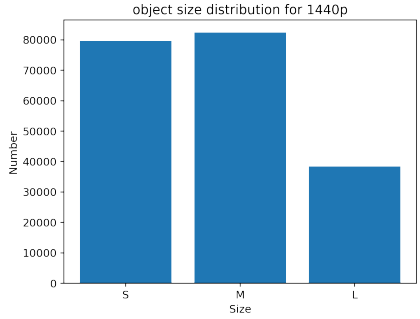
(a) Object class for 1080p videos



(b) Object class for 1440p videos



(c) Object size for 1080p videos



(d) Object size for 1440p videos

Figure 3.2: Object class and size distribution for tested videos in two different resolutions.

scenario of a mobile AR system. For our experiment, we tested 100 videos from the Inter4K dataset. The videos are considered dynamic, where the objects and backgrounds move along with the camera. Conversely, static videos are devoid of camera movement.

From the videos we tested, Figure 3.2(c, d) shows the frequency distribution of the sizes of detected objects. The standard for differentiating the size of an object is referenced from the MS COCO dataset [18], where small objects are defined as those occupying less than $0.\bar{3}\%$ of the entire image, and medium objects occupy less than 3%. The objects are quite balanced in terms of size, with large objects being the least common, appearing about 20% of the time. On the other hand, as shown in Figure 3.2(a, b), the ”person” class appears significantly more often than others. Other frequently appearing classes include ”car,” ”boat,” and ”chair.” While the class distribution isn’t balanced, it is considered reasonable and natural. Notably, the classes detected by the object detector are the same

doi:10.5566/NTU2024XXXXX

as those in the MS COCO dataset.

doi:10.5566/NTU2024XXXXX

Chapter 4 Experiment Results

In this chapter, we follow the experiment design and settings from the previous chapter and evaluate the results from different aspects. We will demonstrate the impact of SR methods compared to interpolation and further investigate the effectiveness of saving bandwidth by sending low-quality videos.

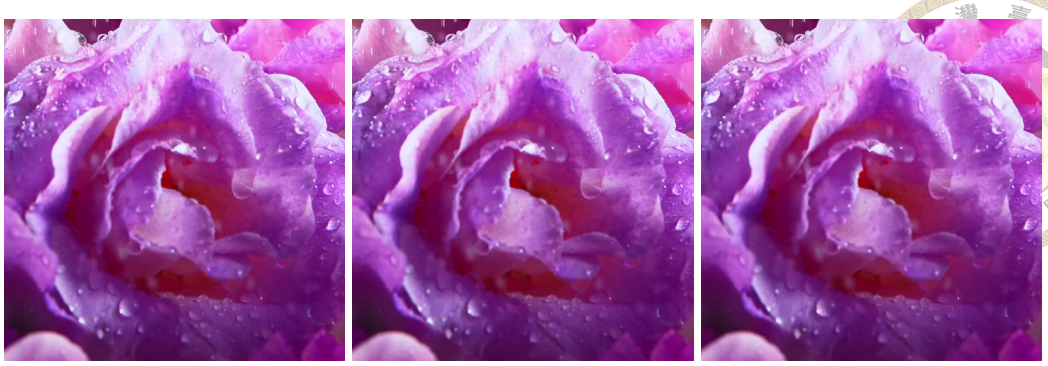
4.1 Resolution and bitrate

Our original intention is to reduce the video quality to decrease the bitrate. By reducing the resolution from 1440p to 360p and from 1080p to 270p, we shrink the height and width of the video by a factor of 4, leaving only 1/16 of the pixels. As shown in Table 5.1, despite having only 1/16 (6.25%) of the pixels, the bitrate remains between 8% and 12% under the same CRF values. Figure 5.1 illustrates an uncompressed video frame alongside its compressed variations.

CRF	1440p	360p	360p/ 1440p	1080p	270p	270p/ 1080p
0	390.07	36.08	9%	235.75	20.91	9%
20	18.33	2.02	11%	12.10	1.21	10%
40	1.81	0.14	8%	1.10	0.09	8%

Table 4.1: Video bitrate (Mb/s) of different qualities with percentages of low resolution compared to high resolution of the same CRF values.

doi:10.5566/NTU2024XXXXX



(a) CRF=0, 235.8Mb/s (b) CRF=10, 41.2Mb/s (c) CRF=20, 12.1Mb/s

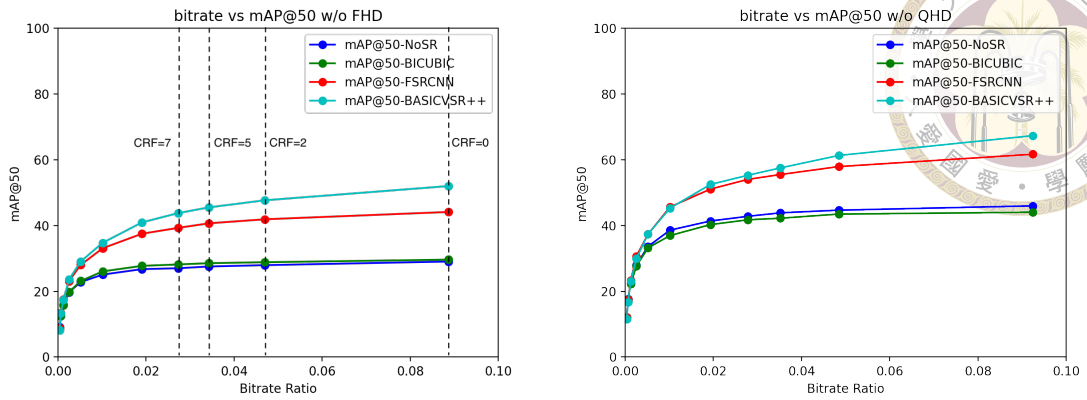
(d) CRF=30, 3.4Mb/2 (e) CRF=40, 1.1Mb/s

Figure 4.1: Image samples of different CRF and corresponding bitrates.

4.2 Results of Different Super-Resolution

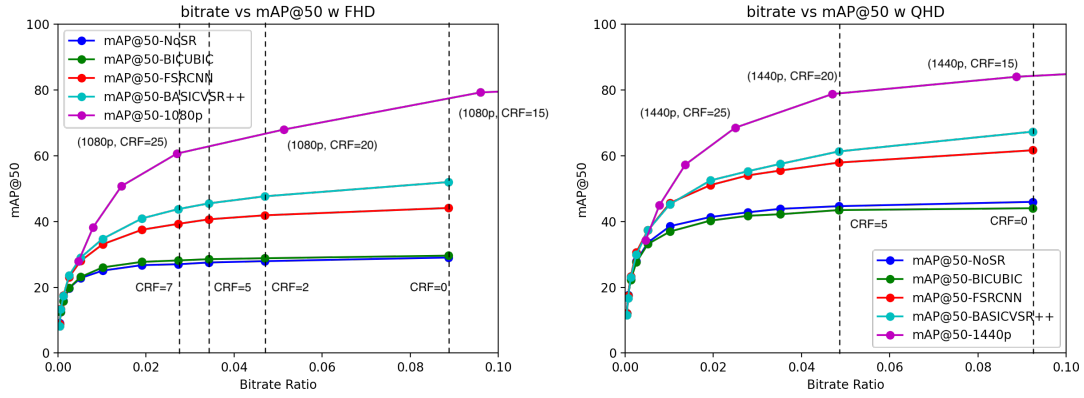
In this section, we evaluate the mAP@50 of different SR-enhanced high-quality videos. In Figure 5.2(a), four lines represent the results of low resolution (no enhancement), bicubic interpolation, FSRCNN (SISR), and BasicVSR++ (VSR) correspondingly. As mentioned in Chapter 4, we selected a set of CRF values (0, 2, 5, 7, 10, 15, 20, 25, 30, 35, 40). The rightmost point of each line represents mAP@50 when CRF=0. Moving from right to left, CRF gradually increases and the quality decreases accordingly. It's important to note that the bitrates are identical for all four methods at the same CRF, as the bitrate is the original transcoded bitrate of the low-quality videos. The enhancement does not affect the bitrate, as it is performed on the server side after the videos are uploaded. The x-axis represents the bitrate ratios compared to the ground truth, uncompressed high-resolution (1080p,1440p) videos. For example, the average bitrate of a 270p, CRF=2 video is only

doi:10.5566/NTU2024XXXXX



(a) 270p and 1080p (b) 360p and 1440p

Figure 4.2: mAP@50 of low-quality frames, SR-enhanced low-quality frames.



(a) 270p and 1080p (b) 360p and 1440p

Figure 4.3: mAP@50 of low-quality frames, SR-enhanced low-quality frames and high-quality frames.

about 4.8% compared to the ground truth (1080p, CRF=0) videos. This means that we can potentially save 95% of bandwidth by sending 270p, CRF=2 videos. The same way of interpretation applies to Figure 5.2(b).

Next, we evaluate the object detection performance boost of SR methods. In both figures of Figure 5.2, bicubic interpolation shows little to no benefit when applied to low-quality video frames. In contrast, FSRCNN achieves a performance increase of 7% to 15% when the bitrate ratio is above 1%, while BasicVSR++ performs slightly better with a performance increase of 7% to 20%.

doi:10.5566/NTU2024XXXXX

What if we simply compress the video without degrading the resolution? Figure 5.3(a) shows the same four lines as Figure 5.2(a), with an additional line representing high-resolution videos. Surprisingly, compressing the video with only CRF can be much more effective than using SR on low-quality videos, resulting in more than a 20% increase in mAP across most of the bitrate range. For example, as seen in Figure 5.3(a), videos of 360p with CRF=7 have a similar average bitrate to videos of 1080p with CRF=25, while the latter outperforms the second-best BasicVSR++ enhancement by over 20% in mAP.

Figure 5.4 presents visual examples corresponding to those depicted in Figure 5.3. In Figure 5.4(a), the ground truth is represented, while Figure 5.4(b) corresponds to the rightmost point of the blue line in Figure 5.3. Notably, Figure 5.4(b) displays the worst result, featuring multiple labels on the ”fire hydrant” and a misprediction of a yellow motorcycle in the background. Moving on, Figure 5.4(c) showcases the result of videos processed with FSRCNN, reflecting the rightmost point of the red line in Figure 5.3. While Figure 5.4(c) shows an improvement over Figure 5.4(b) with a correct single “fire hydrant” prediction, the misprediction of the yellow motorcycle persists in the background. Finally, Figure 5.4(d) corresponds to the rightmost point of the purple line in Figure 5.3, exhibiting a similar but slightly lower bitrate and better mAP. The visual examples align with the results, displaying higher confidence in the correct “fire hydrant” prediction and the correct pink bicycle prediction in the background.

4.3 CQP and CRF

Although the two modes utilize QP in different ways, Figure 5.5 indicates that when the overall bitrate is tuned to the same level, both PSNR and mAP show no significant

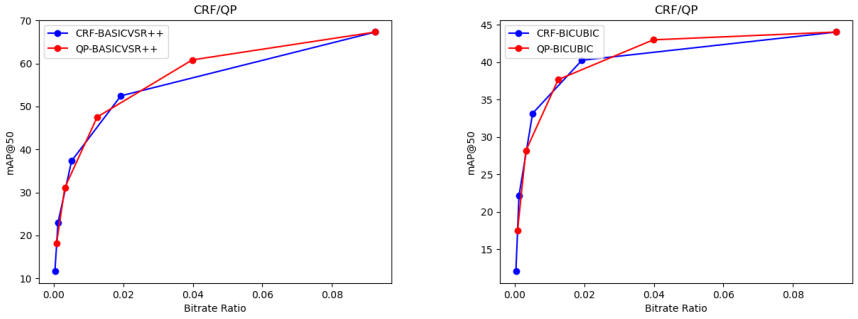
doi:10.5566/NTU2024XXXXX

(a) 1440p, CRF=0 (b) 360p, CRF=0

(c) 360p, CRF=0, SR=FSRCNN (d) 1440p, CRF=15

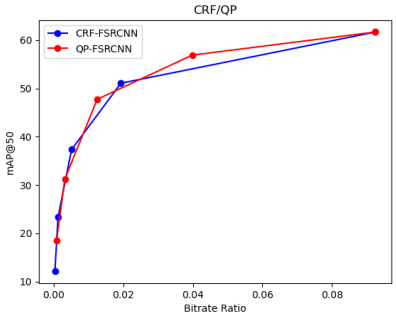
Figure 4.4: Sample bounding box results on video frames of different qualities.

doi:10.5566/NTU2024XXXXX

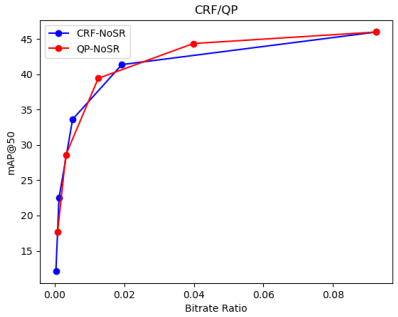


(a) BasicVSR++

(b) Bicubic



(c) FSRCNN



(d) NoSR

Figure 4.5: Object detection results of CRF compared to CQP.

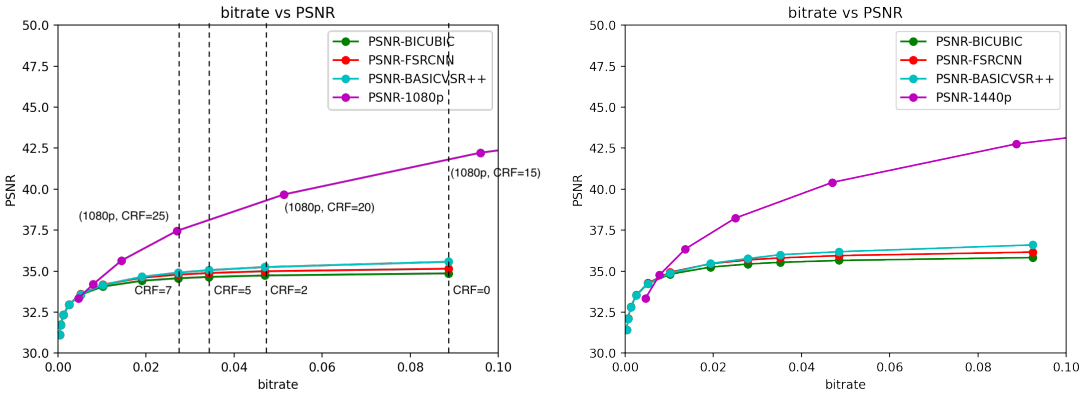
difference between the two modes. This is why we primarily use only CRF in most of the experiments.

4.4 Evaluation of PSNR

To examine the relationship between PSNR and mAP@50 for object detection, we also recorded the PSNR of the SR-enhanced high-quality frames. In Figure 5.6, we can observe that the PSNR of compressed high-resolution videos surpasses other SR methods by a large margin, which aligns with the results shown in Section 5.2: higher PSNR corresponds to higher mAP. However, the PSNR of both SR deep neural networks shows little difference, approximately 1 dB, compared to the PSNR of bicubic interpolation.

This indicates that while SR models might perform as poorly as interpolation in terms of video quality without specific fine-tuning or learning, they can still be beneficial in cases

doi:10.5566/NTU2024XXXXX



(a) 270p and 1080p (b) 360p and 1440p

Figure 4.6: PSNR of low-quality frames, SR-enhanced low-quality frames and high-quality frames.

(a) Interpolation (b) FSRCNN (c) BasicVSR++ (d) High-resolution

Figure 4.7: Examples of different high-quality and low-quality frames.

doi:10.5566/NTU2024XXXXX

where object detection is the downstream task. However, in our experiment, compressing the video without degrading the resolution still appears to be more effective than using SR methods.

Figure 5.7 displays sample frames at 270p, CRF=7 after interpolation and SR, alongside a frame at 1440p, CRF=25, which has nearly the same bitrate as the previous three. It's evident that frame (d) exhibits the best results, while frames (c) and (b) are slightly better than (a).

doi:10.5566/NTU2024XXXXX

Chapter 5 Conclusions

5.1 Conclusions

The demand for high-quality video delivery, coupled with applications requiring high computation, is growing and will continue to do so in the foreseeable future. Therefore, smoothly transferring videos with limited bandwidth remains a critical factor. In this research, we conducted a quantitative analysis to gain a better understanding of how VSR performs in an edge-assisted object detection scenario. Both SISR and VSR demonstrate a non-negligible performance boost in terms of mAP. However, the results also indicate that scaling down the resolution solely to save bandwidth might not be a great tradeoff, especially considering the additional processing time required for applying SR methods. Nevertheless, in scenarios where high-quality frames are unattainable, SR could prove useful.

Another critical result highlighted in the thesis is the significant role of quantitative parameters in video encoding and rate control. We often become distracted by the notion that resolution is the most critical factor when compressing videos, as many major video delivery platforms offer multiple options only for resolution. However, comparing the impact of CRF with SR, it's evident that working with these built-in functionalities of video codecs can be as impactful, if not more so, than directly using state-of-the-art SR

doi:10.5566/NTU2024XXXXX

neural networks.

The results raise questions about why state-of-the-art SR techniques, while highly prominent, underperformed in our experiments. One evident issue in our study is reflected in the PSNR of SR-enhanced frames. These models did not perform well, possibly due to differences between our dataset and how low-resolution videos are typically generated for their training data. Typically, SR models are trained on low-resolution images originally derived from high-resolution images degraded with blurry filters and downsampled using methods like MATLAB's *imresize* [2, 32] function.

This discrepancy may explain why in related works [11, 17, 30] focusing on SR-enhanced live streaming, researchers heavily optimize SR model performance. For example, they might send low-resolution videos with occasional high-resolution segments for online training or seek a balance between using SR and directly transmitting high-resolution content. Moreover, in live streaming or mobile AR scenarios, video super-resolution (VSR) models must operate in real-time and often cannot utilize future frames for SR, imposing significant constraints on many state-of-the-art VSR models.

5.2 Future Works

We conducted tests across different levels of compression and complexities of SR methods, with the SR ratio fixed at 4 in this study, ignoring other common ratios like 2 and 3. We speculate that the results will remain consistent with ratios like 2 and 3, as SR deep neural models, interpolation, and quantization in video codecs work on the same principle despite having different scaling ratios. However, this is not something we can immediately confirm without further experimentation. In the future, we aim to incorporate

doi:10.5566/NTU2024XXXXX

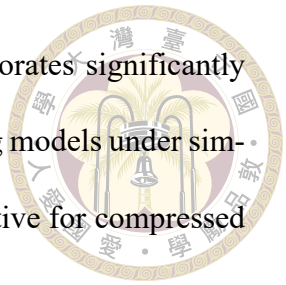
scaling ratios into our measurements to extend the generality of this study.

Furthermore, it's worth noting that the SR methods perform poorly considering PSNR for approximately a 1 dB increase, far from the capacity shown by these state-of-the-art models. Since we tested the models with datasets and videos that they were not trained on, as mentioned in Chapter 2, and the way we generate low-resolution videos differs from how they are trained also, the performance degradation is much as expected. However, other works [11, 17, 30] have shown performance increases by methods like utilizing online training to provide the model with more information on the target dataset. In the future, we also aim to focus on optimizing SR models to potentially improve their ability without directly fitting the model to specific datasets, which is not what we aim to do due to the variety of scenes that might appear on users' mobile devices.

In addition, since this thesis focuses purely on experimental analysis, it is important to discuss future directions that could extend the implications of this work. Based on the discussions and results above, future implications fall into two directions: (a) Developing algorithms or protocols to optimize compression decisions based on current network conditions and SR models on edge servers. While not a new idea, working with Constant Rate Factor (CRF) introduces challenges because CRF aims for consistent quality, yet resulting bitrates vary significantly with video content. Selecting the CRF value without adequate content knowledge can lead to suboptimal compression. Algorithms must consider both bandwidth consumption and video content to make informed decisions. (b) One of the distinguishing features of this thesis is the integration of compression and SR for analysis. Recent work has explored developing Video Super-Resolution (VSR) models specifically for compressed videos [29], as traditional VSR models are often trained on ideal low-resolution images or videos, which may not perform well with highly com-

doi:10.5566/NTU2024XXXXX

pressed content. As shown in my thesis, model performance deteriorates significantly with higher compression levels. Further research could involve testing models under similar frameworks to identify which VSR characteristics are most effective for compressed videos.



doi:10.5566/NTU2024XXXXX

References

[1] Ffmpeg official website. https://www.ffmpeg.org/.

[2] Matlab official website. https://www.mathworks.com/products/matlab.html.

[3] Nvidia official website. https://www.nvidia.com/.

[4] Opencv official website. https://www.opencv.org/.

[5] M. Aqqa, P. Mantini, and S. K. Shah. Understanding how video quality affects object detection algorithms. In VISIGRAPP (5: VISAPP), pages 96–104, 2019.

[6] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012.

[7] C. Borel-Donohue and S. S. Young. Image quality and super resolution effects on object recognition using deep neural networks. In Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications, volume 11006, pages 596–604. SPIE, 2019.

[8] K. C. Chan, X. Wang, K. Yu, C. Dong, and C. C. Loy. Basicvsr: The search for essential components in video super-resolution and beyond. In Proceedings of the

doi:10.5566/NTU2024XXXXX

IEEE/CVF conference on computer vision and pattern recognition, pages 4947–4956, 2021.

[9] K. C. Chan, S. Zhou, X. Xu, and C. C. Loy. Basicvsr++: Improving video super-resolution with enhanced propagation and alignment. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 5972–5981, 2022.

[10] J.-W. Chen, C.-Y. Kao, and Y.-L. Lin. Introduction to h.264 advanced video coding. In Asia and South Pacific Conference on Design Automation, 2006., pages 6 pp.–, 2006.

[11] Y. Chen, Q. Li, A. Zhang, L. Zou, Y. Jiang, Z. Xu, J. Li, and Z. Yuan. Higher quality live streaming under lower uplink bandwidth: an approach of super-resolution based video coding. In Proceedings of the 31st ACM Workshop on Network and Operating Systems Support for Digital Audio and Video, pages 74–81, 2021.

[12] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13, pages 184–199. Springer, 2014.

[13] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence, 38(2):295–307, 2015.

[14] C. Dong, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. In Computer Vision–ECCV 2016: 14th European Conference,

doi:10.5566/NTU2024XXXXX

Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14, pages 391–407. Springer, 2016.

[15] M. Haris, G. Shakhnarovich, and N. Ukita. Task-driven super resolution: Object detection in low-resolution images. In Neural Information Processing: 28th International Conference, ICONIP 2021, Sanur, Bali, Indonesia, December 8–12, 2021, Proceedings, Part V 28, pages 387–395. Springer, 2021.

[16] L. N. Huynh, Y. Lee, and R. K. Balan. Deepmon: Mobile gpu-based deep learning framework for continuous vision applications. In Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services, pages 82–95, 2017.

[17] J. Kim, Y. Jung, H. Yeo, J. Ye, and D. Han. Neural-enhanced live streaming: Improving live video ingest via online learning. In Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication, pages 107–125, 2020.

[18] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco: Common objects in context, 2015.

[19] H. S. Malvar, A. Hallapuro, M. Karczewicz, and L. Kerofsky. Low-complexity transform and quantization in h. 264/avc. IEEE Transactions on circuits and systems for video technology, 13(7):598–603, 2003.

[20] MMEditing Contributors. MMEditing: OpenMMLab image and video editing toolbox. https://github.com/open-mmlab/mmediting, 2022.

doi:10.5566/NTU2024XXXXX

[21] Y. R. Musunuri, O.-S. Kwon, and S.-Y. Kung. Srodnet: Object detection network based on super resolution for autonomous vehicles. Remote Sensing, 14(24):6270, 2022.

[22] S. Nah, S. Baik, S. Hong, G. Moon, S. Son, R. Timofte, and K. M. Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In CVPR Workshops, June 2019.

[23] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, and T. Wedi. Video coding with h. 264/avc: tools, performance, and complexity. IEEE Circuits and Systems magazine, 4(1):7–28, 2004.

[24] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[25] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1874–1883, 2016.

[26] Y. Siriwardhana, P. Porambage, M. Liyanage, and M. Ylianttila. A survey on mobile augmented reality with 5g mobile edge computing: Architectures, applications, and technical aspects. IEEE Communications Surveys & Tutorials, 23(2):1160–1192, 2021.

[27] A. Stergiou and R. Poppe. Adapool: Exponential adaptive pooling for information-retaining downsampling. IEEE Transactions on Image Processing, 32:251–266, 2022.

doi:10.5566/NTU2024XXXXX

[28] Y. Wang, S. M. A. Bashir, M. Khan, Q. Ullah, R. Wang, Y. Song, Z. Guo, and Y. Niu. Remote sensing image super-resolution and object detection: Benchmark and state of the art. Expert Systems with Applications, 197:116793, 2022.

[29] Y. Wang, T. Isobe, X. Jia, X. Tao, H. Lu, and Y.-W. Tai. Compression-aware video super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2012–2021, 2023.

[30] Z. Wang, Z. Luo, M. Hu, D. Wu, Y. Cao, and Y. Qin. Revisiting super-resolution for internet video streaming. In Proceedings of the 32nd Workshop on Network and Operating Systems Support for Digital Audio and Video, pages 8–14, 2022.

[31] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra. Overview of the h. 264/avc video coding standard. IEEE Transactions on circuits and systems for video technology, 13(7):560–576, 2003.

[32] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman. Video enhancement with task-oriented flow. International Journal of Computer Vision, 127:1106–1125, 2019.

doi:10.5566/NTU2024XXXXX