NeuriCam: Video Super-Resolution and Colorization Using Key Frames

Bandhav Veluri, Ali Saffari, Collin Pernu, Joshua Smith, Michael Taylor, Shyamnath Gollakota
Paul G. Allen School of Computer Science & Engineering
University of Washington

Abstract — We present NeuriCam, a key-frame video super-resolution and colorization based system, to achieve low-power video capture from dual-mode IoT cameras. Our idea is to design a dual-mode camera system where the first mode is low power (1.1 mW) but only outputs gray-scale, low resolution and noisy video and the second mode consumes much higher power (100 mW) but outputs color and higher resolution images. To reduce total energy consumption, we heavily duty cycle the high power mode to output an image only once every second. The data from this camera system is then wirelessly streamed to a nearby plugged-in gateway, where we run our real-time neural network decoder to reconstruct a higher resolution color video. To achieve this, we introduce an attention feature filter mechanism that assigns different weights to different features, based on the correlation between the feature map and contents of the input frame at each spatial location. We design a wireless hardware prototype using off-the-shelf cameras and address practical issues including packet loss and perspective mismatch. Our evaluation shows that our dual-camera hardware reduces camera energy consumption while achieving an average gray-scale PSNR gain of 3.7 dB over prior video super resolution methods and 5.6 dB RGB gain over existing color propagation methods.

Open-source code: https://github.com/vb000/NeuriCam

1 Introduction

The power consumption of video camera systems is a bottleneck for multiple applications like security cameras [26, 37, 55, 64], wearable cameras (e.g., Google glass), robotics and sensor deployments for wildlife monitoring and smart farms [13, 40]. While advances in sensor hardware have enabled microphones, temperature and pressure sensors that are low-power (10 to 100 uW) [72, 83], camera sensor hardware can generate significantly more data and thus have orders of magnitude higher power consumption [1, 2].

There is roughly a 100x difference in the power consumption of grey-scale, noisy, low resolution image sensors and higher resolution RGB sensors (Fig. 2) — the HM01B0 sensor supports grey-scale, noisy QQVGA (160x120) resolution and consumes only 1.1 mW, while the OV7735 color sensor consumes 100 mW at VGA (640x480) resolution [1, 2]. Three key factors contribute to this power difference.

• Video resolution. In contrast to a temperature sensor that outputs a single value, a VGA camera outputs 640x480 pixels per frame. At a frame rate of 15 fps, this is around 4.6 million pixels per second. Thus, increasing the video resolution requires both a higher clock rate and significantly more energy consumption to support the I/O operations [45, 60].

• Supporting color. RGB image sensors that can support color require increasing the number of channels per pixel which further increases the clock rate and I/O requirements.

• Noise and dynamic range. CMOS image sensors can reduce the power consumption at the cost of a higher read noise and fixed pattern noise. This however results in a lower signal to noise ratio (SNR) and dynamic range [4, 48].

This paper explores the following problem: can we design high resolution color video camera systems while minimiz-
ing camera energy consumption? Our idea is to design a dual-mode camera system where the first mode is low power (1.1 mW) but only outputs grey-scale, low resolution and noisy video and the second mode consumes much higher power (100 mW) but outputs color and higher resolution images. To reduce total energy consumption, we heavily duty cycle the high power camera to output an image only once every second. This data can then be wirelessly streamed to a nearby plugged-in access point or router that is not power constrained, where we run our real-time neural network decoder to reconstruct a higher resolution color video.

Our dual-mode\(^1\) approach to low-power video capture has three key advantages. First, heavily duty cycling the high power image sensor reduces its energy consumption. For example, duty-cycling it to capture one frame per second, can reduce its average power consumption by a factor of the target frame rate (e.g., a power reduction of roughly 15x for 15 fps video). Second, capturing lower resolution monochrome video for the remaining frames, can significantly reduce the amount of data captured by the camera system. For example, capturing greyscale 160x120 video at 15 fps and the key frame instead of RGB 640x480 video reduces the amount of data captured by a factor of 48. Finally, the router can combine the data across the two streams by super-resolving the low-resolution grey-scale video using the heavily duty-cycled high-resolution video frames, which we refer to as key-frames. We show that these key-frames act as ground-truth priors for reconstructing precise high-frequency details that is missing in the low-resolution frames. In addition, these key-frames are also a source of information for color, allowing us to generalize the super-resolution task to color.

Reconstructing high resolution color video from our low-power dual-camera system is challenging for three reasons: (i) since the input video from the always-ON camera is grey-scale, key frames from the heavily duty-cycled color camera are the only source of color. One naïve approach is to use the closest key frame to propagate color across the video. However, occlusions in the current frame may not always be present in the closest key frame due to motion, (ii) since we capture the higher resolution color key frames only once per second, there can be significant relative motion between the objects in the key frame and current frame. Standard convolutional operations mostly focus on local features and if there is a significant relative motion between these frames, they tend to fail to capture features that may have moved hundreds of pixels away, and (iii) since in our practical implementation, we use two cameras in slightly different locations, we can not assume that the transformation between the two views will preserve parallel lines. Thus, we need to account for a change in the perspective between the two cameras as well as different fields of view while combining the data.

We present a novel deep learning-based system to achieve video capture from low-power dual-mode IoT systems. To achieve this, we tackle key-frame based video super-resolution and colorization as a single problem. Our design uses a bidirectional recurrent network architecture that propagates information between the key-frames. The bidirectional nature of our network ensures that the two high-resolution key frames that appear before and after the low-resolution frames are both used to super-resolve and colorize the video. We introduce a novel attention feature filter mechanism that assigns different weights to different features, based on the correlation between the feature map and contents of the input frame at each spatial location. This ensures that correct features are used across the two key-frames to colorize different objects in the current frames (see §3.1.1). To address the problem of high mobility across frames, we use optical flow based alignment as an initial step to do long range motion alignment, and then refine it using deformable convolutions. We also aggregate temporal information to extract details from neighboring frames using grid propagation (see §3.1.2).

We first evaluate our neural network using standard datasets to emulate a dual-mode system with different resolutions, which we train using the Vimeo-90K dataset and perform evaluation on the Vid4, UDM10 and REDS4 datasets. The REDS4 dataset is a first-person point-of-view camera dataset with a lot of random motion. Our evaluation shows that:

- Compared to state-of-the-art video super resolution techniques (BasicVSR++ [11]), our key-frame based approach increases the grey-scale video PSNR by 1.7-4 dB across the three datasets\(^2\). These improvements in the video quality can help with more faithful reconstruction of finer details such as signs, letters, shapes and textures (Fig. 5) and are possible because we can learn from the finer details in the high-resolution key frames that are absent in the low-resolution video.

- We achieve a 4-6 dB gain in the colorization performance compared to prior color propagation methods (DEVC [25]). In comparison to reference based image super resolution methods (TTSR [85]), our bidirectional recurrent method with attention mechanism that use the key-frames achieves a 4-7 dB performance improvement across the three datasets.

We prototype a low-power dual-camera hardware using off-the-shelf components. Our prototype includes a Himax HM01B0 image sensor capturing greyscale video at 15 fps and QQVGA resolution and a OmniVision OV7692 color VGA sensor capturing key-frames once every second. Our prototype design includes microcontrollers and wireless ra-

\(^1\)Ideally, one could design a custom camera IC that can read out low resolution gray scale continuously and provide a higher resolution color frame periodically with minimal switching delay. In practice, given the switching delays and resolutions of commodity low-power cameras, we implement the dual-modes using two different cameras. In the rest of the paper, we reuse dual-mode and dual-camera interchangeable, given this context.

\(^2\)For fairness, our PSNR computations exclude the key-frames.
The runtime of our model on a Nvidia RTX 2080 Ti GPU is around 54 ms per frame. For context, at 15 fps, the receiver must process each frame in less than 66 ms to be real-time. By increasing the batch size and GPU utilization, we can reduce the per-frame processing time to 44 ms.

Our system can achieve color 640x480 wireless video streaming at 15 fps while consuming only 46 mA. The active power is much lower than existing wireless video streaming systems at comparable frame rates and resolution (Table 1).

Contributions. We address the problem of designing high resolution color video camera systems while minimizing camera energy consumption. To this end, 1) we introduce a dual-mode camera approach where the first mode is low power but outputs grey-scale, low resolution video while the second mode is high power but outputs heavily duty-cycled color and higher resolution images. 2) we design an efficient deep learning technique for key-frame based video super-resolution and colorization. 3) we address practical issues including perspective mismatch and wireless packet losses. 4) We provide extensive evaluations showing significant gains over state-of-the-art video super resolution methods as well as prior color propagation methods. 5) We built a prototype hardware that shows significant energy savings achieved by our low-power wireless camera system. 6) By making our algorithms and designs open-source at publication, we will help sustain progress in creating these low-power video systems.

2 Related Work

Energy-efficiency and cameras. There has been recent work on designing more power-efficient camera sensors [8, 35, 54], mixed-signal vision integrated circuits [43, 80] and processors [63, 70]. The general relation between power consumption versus resolution, color and SNR however still applies. [14] uses a combination of image compression and a simple pixel-difference based motion detection on the receiver to compress the input video. [6] on the other hand does not use image compression but implements a remote control mechanism that can trigger the transmission of a single image. [32, 61] shows wireless cameras for medical diagnosis.

Many commercial wireless cameras use on-device video compression algorithms to reduce the data transmission rate, including recent work that uses neural networks for on-device video compression [49, 62, 69, 79]. These neural networks, however, are typically designed for internet streaming, which assumed a complex encoder running on a compute-intensive device and a simple decoder for the compute-constrained device side. Low-power IoT cameras on the other hand have reversed complexity, where the encoder is extremely resource constrained, but the decoder can be relatively complex. While low power accelerators have been proposed for compression, they are limited to images [8, 26]. [38, 57, 77] use low-power image compression, and also serve as good references for limits on power consumption of compression-based techniques. We focus on the camera sensor power consumption and explore the use of key-frame based super resolution and colorization. Compression is complementary to sensor power and can also be applied to our low-power dual camera system.

More recently, BeetleCam [33] presents a Bluetooth-based video compression system for low-resolution video. [37, 55, 64] proposed backscatter systems for achieving low-power wireless video streaming (see Table 1). Backscatter is complementary to our approach and can be used instead of our off-the-shelf radio to reduce communication power consumption.

Neural networks for video processing. Super-resolution is a classic problem that has gained renewed popularity with the effectiveness of deep learning methods [10, 27, 29, 41, 47, 65, 67]. Variations to traditional convolution operations like sub-pixel convolutions [67], deformable convolutions [18, 18, 19, 36, 73, 75, 88] have been shown to work well for video super-resolution. However, more recently, it has been
shown that recurrent networks [11, 24, 31] can outperform feed-forward networks that operate on a small batch of frames, at much lower complexity. [11] performs super-resolution on these low-resolution frames and the rest of the frames are super-resolved using motion information and residual present in the encoded bit stream referring to super-resolved key-frames. In contrast to our use of high-resolution key-frames to super resolve and colorize low-resolution grey-scale noisy video, these works perform super-resolution using low-resolution frames and hence cannot extract high-frequency details that may be non-existent in the low-resolution input.

While reference based image super resolution has been explored [86, 87], key-frame based video super-resolution has not received much attention from the research community in recent years. Early work on key-frame based super-resolution considers this problem as a special case of more general example-based super-resolution problem [23], where a low-resolution image or video is super-resolved based on a pool of high resolution images. While generalized block matching methods have been proposed for the example-based super-resolution problem [23, 59], works on key-frame based super-resolution specialize those methods by performing motion compensated block matching. Those methods, however, only achieve a 2x scaling and do not consider color [9, 28, 71]. Recent related work on reference-based super-resolution (RefSR) [20, 68, 85–87] only consider images, not video, and do not transfer color. To the best of our knowledge, our work presents the first deep learning technique to tackle both key-frame based video super-resolution and colorization.

Early work on colorization focused on techniques to propagate user edits or suggestions to other parts of an image using optimization techniques [44, 50]. More recently, deep learning methods have been shown to be more effective by gaining a higher level understanding of image contents [17, 22, 30, 31, 66, 84]. These methods are primarily designed for image editing applications, and need color hints at precise locations. As a result, they tend to lose temporal coherence within a few frames [52]. Welsh et al. [32], suggested that color spaces that separate luminance and color channels work better than RGB color space. In recent years, deep learning methods have been proposed for color propagation to achieve reliable transfer of color for large number of frames [34, 52, 53]. These methods operate on input videos in original resolution and as a result do not perform scaling on luminance channel. Further, we demonstrate that our two-way propagation mechanism, achieves a much higher PSNR.

Prior work also designed neural networks for dual-camera settings. [76] improves the performance of stereo image super resolution by using the symmetry cues in stereo image pairs. [74] targets super resolution for the wide-angle and telephoto cameras on the phones that capture a scene with different focal lengths. More concurrently, [42] proposes a three-color camera system for super-resolving a low-resolution ultra-wide video utilizing wide-angle and tele-photo videos. [16] addresses the problem of super resolution using two cameras where one camera shoots a video with high spatial resolution and low frame rate while the second camera captures a low spatial resolution and high frame rate video. Their target application is high resolution video (4K) at high frame rate (240 fps). In contrast, in our low-power application scenario, our high resolution camera is heavily duty cycled and further, the low resolution camera has a very low resolution (160x120) and in contrast to all the above systems does not support color.

### 3 System Design

Video camera systems in the IoT context, typically involve a transmitter that is power constrained, and a receiver with a relatively high power budget. As seen in Table 1, low-power video streaming systems operating on the order of 10mW are only able to stream QQVGA (160x120) grayscale video at 1-5fps. On the other hand, high-resolution color video streaming systems consume 700-1800mW for streaming VGA (640x480) to HD resolutions. We observe that the sensor power is significant at high-resolutions due to the large number of pixels being activated for each frame as well as the I/O required to handle the data rates. To reduce the sensor power while also preserving the high-frequency details available in high-resolution frames, we propose a dual-camera system where the first camera is low power camera capturing 15fps grayscale video at 160x120 resolution, operating at 1.1mW, and the second high-resolution camera is heavily duty cycled for 1fps color video at 640x480 resolution, operating at 100mW, pre-duty cycling. The two video streams transmitted from this transmitter are then fed to a neural network decoder on the receiver to generate 15 fps color video at 640x480 resolution. We also note that the power numbers in Table 1 includes the image sensors as well as communication and on-device processing.

While, we reduce the amount of data captured by the sensor, one can optimize the energy consumption of communication by using backscatter communication, like prior approaches. This however is not in the scope of our paper.

In this section, we describe the design process of the neural
network model that works with the proposed sensing method as well as address practical issues such as perspective mismatch and packet losses. We then provide a description of the hardware prototype we designed to demonstrate our system.

3.1 Design of the neural network model

Specific to our task of key-frame based video super-resolution and colorization, since key-frames are received only once every second, the model should be capable of mapping objects in the key-frames to 15 low-resolution frames in the presence of large relative motion. This is especially challenging if an object in the current low-resolution frame is occluded in the key-frame. To address this, we implement a bidirectional recurrent network that propagates information between the key-frames, and introduce a novel attention mechanism to selectively choose forward or backward features from each region of the image.

Fig. 3 shows the architecture of our neural network model. We split the input video sequence into key-frame to key-frame sequences before they are fed into the network. So, in each iteration, the network operates on 15 low-resolution frames, \( \{f_i\} \), and two key-frames (previous and next), \( \{k_i\} \). Multiple key-frame to key-frame sequences can be stacked along the batch dimension to increase the throughput of the network. While the bidirectional recurrent network with forward and backward paths as well as the feature extractors are building blocks we use from prior work, the attention feature filters are novel blocks we introduce in this work for this task, applying the attention mechanism. In the rest of this section, we describe our attention mechanism and then provide a detailed description of the complete architecture.

3.1.1 Attention feature filters

**Problem.** Since our method is based on extracting and mapping high-resolution details and color from key-frames, it is vulnerable to scenarios where an object appears in a frame between the key-frames, but the closest key-frame does not have the object. Since we get color information primarily from the key-frames, consider a scenario where frame 2 has an object that appears in view, but is not present in previous (closest) key frame 0 but is present in next key frame. The recurrent forward blocks in Fig. 3 are heavily influenced by the previous key frame since it is closest key frame. The color information for such an object might only be present in the next (farther) key frame. While alignment and propagation are effective at aggregating features from the entire sequence, those blocks do not explicitly consider the content of the frame and its correlation to features at each level. This results in blurred and grayed out details, limiting our ability to effectively super-resolve and colorize.

**Our attention mechanism.** To solve this problem, we propose feature filter based on attention mechanism [7], that assign varying level of weightage to each filter level, based on the correlation between the feature map and contents of the input frame at each spatial location.

For time steps \( t \in \{0, 14\} \), let \( w \times h \) be the resolution of the low-resolution input frames, \( f_i \in \mathbb{R}^{w \times h \times 64} \) be the low-resolution input feature map computed by the low-resolution feature extractor, and \( f_{ij} \in \mathbb{R}^{w \times h \times 64} \) be aggregated features output by alignment and propagation blocks at level \( l \) in the grid shown in Fig. 3. 64 is the channel depth of the feature maps that our network operates on. Then, the features computed by the attention feature filter, \( \hat{f}_i \in \mathbb{R}^{w \times h \times 64} \), is given by:

\[
\hat{f}_i = \sum_{l=0}^{3} f_{ij} \odot A(f_{ij}, f_{il})
\]

where \( A(f_{ij}, f_{il}) \in \mathbb{R}^{w \times h} \) is the attention score value (which effectively is the weight) between low-resolution input feature map and the aggregated feature map at level \( t \) at each spatial location. And, \( \odot \) effectively denotes the element wise multiplication between the feature map and attention value computed for that feature map at that spatial location where all the 64 values/channels in the feature map are multiplied by the attention value. Thus, the attention score value is unique for each spatial location in the frame but is the same across all the 64 channels in the feature map. This allows us to uniquely sample the relevant feature maps at each spatial location, based on the contents of that location.

We use the following definition for attention score function at each spatial location \((i, j)\):

\[
A(f_{ij}, f_{il}) = \text{Softmax}(f_{il}^i, f_{ij}^l)
\]

Here, we first compute the dot product similarity between the low-resolution features, \( f_i \in \mathbb{R}^{64} \), and aggregated features \( f_{ij} \in \mathbb{R}^{64} \), at spatial location \( i, j \). This results in four similarity scores, one for each propagation level. We use these scores to choose the features map with maximum similarity. But since choosing the maximum would be a non-differentiable operation, we use the Softmax function to perform non-maximum suppression of the scores in a differentiable way.

Essentially, the summation in Eq. 1 is a weighted average of correlation between feature map and low-resolution input frame at each spatial location, over all propagation levels in the network. The importance of our attention mechanism to our task of key-frame based super-resolution and colorization is demonstrated in Fig. 4. Here, we can observe that the occluded region is better predicted by the model with attention using information from the next key-frame, despite frame 7 being closer in time, to the previous key-frame. The model without attention is able to predict the structure well from the low-resolution input, but fails to prioritize the second key-frame to infer the accurate color information.

3.1.2 Neural network details

We describe the remaining blocks in Fig. 3 for completeness.
**Figure 3:** Complete architecture of our model with aligned grid propagation and attention similarity filters. Residual blocks (shown in blue in the right figure) are the processing elements used in feature extraction and feature propagation. Recurrent element (green block in the right figure) align the neighboring features and compute aggregated features at each level in the grid.

**Feature extractors.** Both the low-resolution and high-resolution input frames are fed into feature extraction blocks. We use 5 residual blocks for the low-resolution feature extractor and 7 residual blocks for key-frame feature extractor. As shown in Fig. 3, each residual block is a sequence of two 2D convolution operations followed by a residual skip connection. The first two convolutional layers in the key-frame feature extractor subsample the input spatially while increasing the channel depth to generate key-frame features that have the same dimensions as low-resolution features. We use separate feature extraction blocks with independent parameter sets for low-resolution frames and key-frames. However, the feature extractors used for individual low-resolution frames, or individual key-frames, is the same.

**Deformable alignment.** In videos with high relative motion between subsequent frames, hidden states computed based on previous frames might align poorly with the current frame. Since convolution operations tend to fail at effectively considering global correlations [21,51], it is hard for residual blocks to transform previous frame features to align with the current frame. [73, 75] proposes the use of deformable convolutions to mitigate this problem. [12] further improves this by using optical flow based alignment as an initial step before applying deformable convolutions. Deformable convolutions are generally hard to train due to the instability caused by large filter offsets. [12] uses optical flow based alignment as an initial step to do long range motion alignment, and then refine it using deformable convolutions, which would now have reasonable offsets. Since we perform similar alignment between neighboring frames, we use the same alignment scheme.

**Grid propagation.** Aggregating temporal information is
important for extracting details from neighboring and key-frames. We use grid propagation to achieve feature aggregation over the entire input sequence refined over multiple levels. Each cell in the grid is shown in Fig. 3(b), contains an flow based deformable alignment module and residual feature propagation module. It has to be noted that alignment and propagation module at each level are different (with unique parameter sets), but same module is used across all time steps.

**Upsampler.** Once we compute aligned feature propagated from both directions, we apply the attention filter described in §3.1.1 to obtain the final feature map, that we can predict the high-resolution color frames with. We use two convolutional layers to generate 48 channel output, which is then followed by a pixel shuffle layer [67] to upsample the output to a 3-channel high-resolution output. We apply tanh activation to the final layer to convert the output to be in the \([-1, 1]\) range. The output we obtain is in \(L^\ast a^b\ast\) colorspace, so we normalize the output to \([0, 1]\) range, convert it to RGB colorspace. We normalize the resulting RGB output to \([0, 255]\) range, round the values and store them in 8-bit unsigned integer format.

**Color space and loss function.** Most prior super-resolution works operate in the RGB colorspace. However, since we are also performing colorization task in tandem, using RGB colorspace resulted in subpar colorization results in our experiments. So, we convert the key-frame, and during training ground-truths, to CIELAB colorspace, which has been shown to perform better for colorization [25, 84]. For loss function however, we found that Charbonnier loss [15] popular in video super-resolution works better compared to L1 or Huber loss functions used in colorization works. Given the high-resolution output frame \(HR_{\text{predicted}}\), and ground-truth high-resolution frame \(HR_{\text{gt}}\), the Charbonnier loss is given by,

\[
L = \sqrt{||HR_{\text{predicted}} - HR_{\text{gt}}||^2 + \epsilon^2},
\]

where \(\epsilon\) is the regularization factor which we set to \(1e-3\) for training.

### 3.2 Practical Issues

**Perspective correction.** Since in practice, the low-resolution and high-resolution sensors are spatially shifted relative to each other, the images captured by the low-resolution and high-resolution sensors would have a different perspectives. Having a same perspective however would be beneficial for alignment. Thus, we compute a fixed homographic transformation for low-resolution frames to transform them into the perspective of the high-resolution frames and use the transformed frames as inputs to the model.

Since the camera location with respect to each other are fixed, any transformation between them would be a fixed transformation that can also be applied as a prepossessing step to the model’s input. As the two cameras might be capturing different planes, we cannot assume that the transformation between them would be an affine transformation. But assuming planar views of both cameras, we can estimate the homographic transformation of the two views. Given four reference points in low-resolution frame and key-frame, we can transform an arbitrary point \((x, y)\) in low-resolution plane to key-frame plane \((x_k, y_k)\), by:

\[
\begin{bmatrix}
  x_k \\
  y_k \\
  1
\end{bmatrix} =
\begin{bmatrix}
  h_{11} & h_{12} & h_{13} \\
  h_{21} & h_{22} & h_{23} \\
  h_{31} & h_{32} & 1
\end{bmatrix}
\begin{bmatrix}
  x \\
  y \\
  1
\end{bmatrix}
\]

We estimate this fixed homographic matrix using four known reference points during a one-time calibration step. We then identify the reference points in the low-resolution frame and key-frame frame. Using these points in low-resolution frame as the start, we estimate the above parameters using opencv’s `getPerspectiveTransform` method. The resulting transformation matrix is then applied as a prepossessing step to the input of the model.

**Packet losses.** In practice, video data transmitted wirelessly can be lost due to packet losses on the wireless channel. We characterize these losses in our evaluation, and here, we describe our approach for handling these losses. In our experiments, we observe that packet loss translates to a loss of 1-2 image lines, on average. However in adverse cases, multiple packets might be lost at a time, and our packet loss handling mechanism should handle that gracefully. With these considerations, we implement a packet correction mechanism, that first detects the start index and count of the number of lines lost and stores it in a key-value map. Starting from the top index in the map, we consider extracting a 5 pixel tall snippet of the error free image, immediately on top of the error line and use bicubic interpolation to expand the snippet to cover the lost lines, while keeping the existing lines as is. This ensures that, starting from the top we progressively fill the lost

---

**Algorithm 1: Packet loss correction algorithm**

```plaintext
Function CorrectPacketLoss(image):
   lost_index_map ← {}
   curr_height ← 0
   while curr_height < Height(image) do
      if IsLineZero(image, curr_height) then
         lost_index_map[curr_height] ← 0
         while curr_height < Height(image) do
            lost_index_map[curr_height] ←
            curr_height ← curr_height + 1
         end
      end
      for each [idx, count] in lost_index_map do
         top ← GetLines(start = idx - 5, end = idx)
         top ← BicubicResized(top, height = count + 5)
         image(start = idx, end = idx + count) =
         top(start = 5, end = count + 5)
      end
   return image
```

---

7
lines all the way to the bottom index.

3.3 Low-power dual-camera hardware design

Low-resolution grayscale camera system. We use a low-power STM32L496 microcontroller (MCU) to initialize the Himax HM01B0 image sensor in QQVGA mode and record 15 frames of grayscale video in one second. Our MCU operating at 80 MHz, uses its PLL (Phase Locked Loop) module to provide a 10 MHz clock for the image sensor operation. Once the image sensor clock is ready, the MCU uses the I2C module to initialize the image sensor in streaming mode and send commands to the Himax camera. Next, the MCU uses the SPI module in slave mode to receive the 8-bit pixels value from the image sensor (SPI master). A DMA (Direct Memory Access) channel is enabled to transfer the data from the sensor (peripheral) to the MCU (memory). This allows us to leave the MCU central core out of the reading process and keep it in sleep mode, which results in lower power consumption. Finally, to transfer the collected data to the basestation, we can use a TI CC2640R2F radio in 8-FSK mode with a maximum throughput of 2.5 Mbps. The radios can be set to different frequencies across at 2.4 MHz band and generates RF signal at 0 dBm. Our STM32L496 MCU uses a DMA channel to transfer data to the TI radio’s memory via UART.

High-resolution color camera system. Our design uses a STM32U575 MCU plus an OmniVision OV7692 color VGA image sensor to record one key-frame per second. The MCU, operating at 160 MHz, generates a 12 MHz clock signal for the image sensor’s operation and initializes it in YUV 4:2:2 mode through an I2C connection. In this mode, the image sensor outputs 16 bits of data per pixel, resulting in a frame data size of 5 Mb, which is smaller than our MCU’s RAM capacity. After the initialization process, the sensor captures a picture and uses its digital video port (DVP) parallel output interface with 8 data lines and three synchronization signals (horizontal reference, vertical reference, and pixel clock) to transfer the image. A DMA channel of the microcontroller is activated to transmit 8 bits of data on the positive edge of the pixel clock signal until a frame is complete. The central MCU core is mainly kept in sleep mode during image reading to preserve energy. To reduce the energy consumption even more, the VGA image sensor is only active for 40 ms which is the duration of the initialization and reading processes. Since the radio’s maximum throughput is 2.5 Mbps, we can should divide the data into two sections and use two separate TI radios operating at two different center frequencies to transfer the data. Our STM32U575 MCU can enable two DMA channels and two UART modules (one for each radio) to send the video data to the radios’ memory.

To synchronize the frames across the two cameras, we add a specific three-byte footer at the end of every frame (13, 0, and 10) and transfer it alongside the image data. The basestation looks for the first footer in the coming data to find the start pixel and reconstruct the images. We implement a 32-bit timer on the low and high-resolution MCUs with a precision of 1 ms. After capturing a frame, the MCU reads the 32-bit timer value and appends it to the image data.

4 Evaluation

4.1 Benchmarking our neural network

Training setup. We use the Vimeo-90K [81] dataset as the standard training dataset for all the methods we compare with. The Vimeo-90K dataset is a large collection of 39K videos with 7 frames per video. Given the limited number of frames per video, we use a key frame interval size of 6 for training (but 15 for evaluation). The sequences are selected such that each sequence has sufficient motion. Following [31], we pad the input frames with 2 pixels, and equivalently pad the key-frames with 8 pixels, along the borders in reflect mode. This would result in an output padded with 8 pixels along the borders, with any edge artifacts induced by the model limited to those 8 pixels. Later, the padding on the output is removed as a post-processing step before evaluation. During training, we augment the dataset by sampling a random crop of size 256x256 from a video sample and then applying random flipping and rotation transformations. Each video sample is converted to the L*a*b* color space and normalized to $[-1, 1]$ range. We use Matlab’s imresize in bicubic downsampling mode to obtain low-resolution input stream for evaluations on standard datasets. We use Adam optimizer [39] with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. We train the model with a batch size of 8 for a total of 80 epochs, with an initial learning rate set to $10^{-4}$. The learning rate is scheduled using ReduceLROnPlateau learning rate scheduler with reduction factor and patience set to 0.1 and 5 epochs.

Evaluation setup. We use three widely used data sets for evaluation: Vid4 [46], UDM10 [82] and REDS4 [75]. Vid4 is a set of 4 videos with varying resolutions and frame count, that contain fine textures that are hard for models to predict. UDM10 is a set of 10 1280x720 videos, each containing 32 frames, that captures relatively high motion. REDS4 is a first-person point-of-view dataset, and thus has a lot of random motion spanning the entire video. REDS4 is set of 4 1280x720 videos each containing 100 frames. The key frame interval is set to 15 matching that of the hardware system and the models are evaluated on PSNR and SSIM [78] metrics computed between individual output frames and corresponding ground-truth frames. As key-frames are available as inputs to the model at original resolution, output frames corresponding to key-frames are of high quality. So, for a fair comparison, per frame metrics are averaged over all frames except the key-frames, to obtain average metrics on a video sample.

Compared schemes. Our model performs multiple vision tasks as a single task: (i) super-resolving the low-resolution
video stream using key-frames as references, and (ii) colorizing the output stream by propagating color from key-frames. We compare the following aspects of our model’s performance to evaluate the effectiveness of our approach:

- **BasicVSR++ and RSDN.** We first evaluate the usefulness of key-frames by comparing with state-of-the-art video super-resolution methods, BasicVSR++ [11] and RSDN [31] that improve the resolution by the same scale (4x) as our system. Since super-resolution methods do not perform colorization of the input low-resolution grayscale stream, we only use Y-channel output of our model for comparisons.

- **DEVC.** To understand how effectively our model can propagate color from key-frames, we compare the colorization performance of our method with the state-of-the-art color propagation method DEVC [25] that propagate color from a reference color frame to a sequence of grayscale frames. We convert the outputs of our method as well as DEVC, to CIELAB color space and evaluate the performance on channels, which contain all of the color information.

- **TTSR.** Finally, we compare our use of temporal information from multiple frames to existing reference based methods that are for images and do not leverage similarities across frames. Since RefSR methods are primarily trained to use images in the wild as reference, they might not be well tuned to super-resolve and colorize sequence of frames in a video. So we retrain the RefSR method TTSR [85] with the same dataset we use for a fair comparison.

**Effectiveness of key-frames for super-resolution.** One alternate solution to achieving high-resolution output video is to stream low-resolution video as is, and use recent advances in deep learning based video super-resolution methods to upscale the low-resolution video. We evaluate the effectiveness of key-frames, both qualitatively and quantitatively, by comparing the performance with video super-resolution methods: BasicVSR++ and RSDN. Since video super-resolution methods do not support colorization, we convert the outputs to YCbCr color space and compare the Y-channels.

In Fig. 5, we compare frame 7 of a video from each of the datasets. Since the key-frame interval is 15, frame 7 is farthest from forward and backward key-frame and hence serves as the worst case scenario for our model. The plots show that our key-frame approach can generate more accurate details compared to traditional super-resolution methods. Our method is able to more faithfully reconstruct the fine details such as signs, letters, shapes and textures.

Fig. 6 compares the Y-channel performance of our method with BasicVSR++ and RSDN. Each bar in the plot represents the distribution of average PSNR and SSIM over each video in the datasets. The results show a 1.7-4dB improvement in PSNR across the three datasets. We also examine the frame-by-frame PSNR progression and compare it to that of prior methods. Fig. 7 plots the PSNRs for each frame averaged across all videos in each dataset. As expected, key-frames improve the baseline quality of the output video, in addition to providing larger gains closer to the key-frames.

**Effectiveness of color propagation.** Obtaining color is an additional challenge in low-power video streaming as color sensors are more power hungry compared to monochrome sensors. To mitigate this problem, in our method, we use low-resolution grayscale sensor as our primary sensor and use key-frames captured from the high-resolution color sensor to colorize the output. While our method upscales the input to higher resolution in addition to colorization, the colorization task is similar to deep learning methods proposed for color propagation which colorize a grayscale video using a color frame as reference. To evaluate the colorization aspect of our model, we compare our model’s color output with the state-of-the-art color propagation method DEVC [25].

DEVC natively operates on 768x432 resolution, and as result we center crop the high-resolution test sets UDM10 and REDS4 to 768x384 to obtain UDM10_768x432 and
Figure 8: We compare our approach with DEVC which does not perform super resolution but can propagate color.

Figure 9: Qualitative comparisons with reference-based image super-resolution methods.

Figure 10: Quantitative comparisons with reference-based image super-resolution methods.

Figure 11: Qualitative comparison on real world data.

4.2 Evaluation with our prototype hardware

Simulating the downsampling process with bicubic downscaling, and generating key-frames from periodically sampled ground truths is effective for evaluating the neural network model architecture by providing a standard and controlled platform to compare with prior methods. However, real world resolution degradation process is much different from the standardized bicubic downsampling process and moreover, varies from one sensor to the other. Using a second camera for capturing key-frames results in image and color artifacts induced by the sensor. In addition to the sensor characteristics, low-resolution and high-resolution streams have a different perspective due to a small difference in their spatial locations. To address these concerns, we evaluate the system’s
we retrain our method as well are BasicVSR++ and TTSR for when there is lot of random motion between frames, for ex-
we observed this method achieves 3 synchronized timers for mark results we observed, our model shows least gains
to both low-resolution and high-resolution micro-controllers. We observed this method achieves 3 synchronized timers for each of the 3 video streams, separated by less than 10ms, which is sufficient to synchronize 15fps video streams.

Capturing the ground truth data. We use a low-power micro-controller for our hardware prototype, so it only supports the high resolution color video at 1 fps. Training the network however, requires 15 fps ground truth video stream that has one-to-one correspondence to the 15fps low-resolution stream. To overcome this problem, we use a third Raspberry Pi camera [5] that can operation at VGA resolution to collect the ground-truth data. For effective training of the model, the input streams have to synchronized with the ground truth stream, which can be challenging. We address the synchronization problem by measuring the relative timestamps. At the start of the data collection, the Raspberry Pi initiates a local counter and at the same time, sends a synchronization pulse to both low-resolution and high-resolution micro-controllers. We observed this method achieves 3 synchronized timers for each of the 3 video streams, separated by less than 10ms, which is sufficient to synchronize 15fps video streams.

Training and evaluation procedure. Based on the benchmarking results we observed, our model shows least gains when there is lot of random motion between frames, for example in the REDS4 dataset. While our system works when pointed towards a busy intersection, to evaluate our method against a similar level of high mobility as is seen in the first-person point-of-view REDS dataset, we capture the entire REDS dataset [58] projected on a screen using our three camera setup. With this setup, we capture a total of 11565 ground truth and low-resolution frames, and, 877 key-frames. We then split the dataset into 224 samples, with each sample approximately containing 50-60 frames. We split the samples making sure that each sample is a contiguous scene without jump cuts. The 224 samples are then split into non-overlapping train and evaluation sets containing 220 and 4 samples, respectively. Since the resulting dataset is significantly smaller than typical simulated datasets, in addition to all the augmentations we apply, in every epoch, we train on each video sample multiple times starting at a random start frame. Using these techniques, we retrain our method as well are BasicVSR++ and TTSR for 40k forward and backward passes with a batch size of 8.

Y-channel performance with hardware data. Similar to our approach for evaluating our model with standard datasets, we compare the Y-channel performance of the model with that of BasicVSR++. Fig. 13 shows an average gain of 3.7dB in Y-channel performance compared to BasicVSR++, more than the 1.7dB gain we observed for REDS4 dataset with simulated downsampling. This shows that our key-frame approach adapts well to the real-world data. Moreover, the improvement in performance gain shows that our method performs better when the amount of training data is limited.

Fig. 11 also shows that relative to the ground truth we trained with, our model generates much clearer details compared to BasicVSR++. The results with hardware data, show the overall effectiveness of key-frames in case of limited data. While traditional super-resolution methods achieve non-trivial quality improvements, compared to bicubic upsampling, the high-frequency details that are not present in the low-resolution input stream cannot be accurately reconstructed. This results in smoothing of details, that results in a loss of sharpness in the resulting video. Periodic snapshots of high-resolution information in the form of key-frames paired with improvements like attention filter enable our model to do faithful reproduction of details.

RGB-channel performance with hardware data. We evaluate the RGB performance of our model on the hardware data, by comparing it against TTSR. In Fig. 14, we observe an RGB performance gain of 5.6dB, correlating well with the 5.7dB gain observed on the standard datasets. Similar to qualitative results observed with standard datasets, Fig. 11 shows that our bidirectional recurrent model with attention feature filters, leverages temporal information to better map the color and high-frequency details from key-frames.
Addressing wireless packet losses. We use the testbed in Fig. 15a to understand the packet-error rate in our system. We fix the location of the transmitter and place the receiver at 10 different locations. The tested locations span a variety of line-of-sight, non-line-of-sight as well as locations that are not in the same room. At each receiver location, we transmit 1000 packets from the transmitter and compute the packet error rate (PER) at the receiver. We repeat this three times at each location to have a total of 30,000 transmitted packets. across all the locations. Fig. 15b plots the CDF of the PER across all the receiver locations. The plot shows that the worse case PER is around 7%. Note that since the packets come with a CRC, we know which of the packets have been incorrectly received. So an incorrect packet can be considered as missing data in the corresponding video frames. Fig. 16 shows that the bicubic interpolation method we implemented to correct the lines lost due to packet error is effective. For most cases, we observe that no noticeable artifacts remain in the corrected images. Even in adverse scenarios where multiple line losses occur together and interpolation method does not have the required information (as seen in the last example of Fig. 16), the artifacts are much less pronounced in the corrected image.

Model run-time analysis. At 15 fps, the receiver must process each frame in less than 66 ms. We measure the model runtime on a Nvidia RTX 2080 Ti GPU where it took less than 54 ms per frame. Since our model operates on key-frame to key-frame sequences, the minimum latency of our system is equal to the period of the key-frame, i.e., 1 s. Running inference on our model with multiple batches (key-frame to key-frame sequences) at a time increases the GPU utilization and as a result, could improve the overall throughput of our model. That is, multiple key-frame to key-frame sequences can be stacked along the batch dimension to improve the throughput of the model. We observe significant throughput gains until a batch size being 4, where the runtime per frame results to around 44 ms. We observe no noticeable throughput gains when increase the batch size to more than 4.

Power analysis. Table 2 shows the average current for the various hardware components. Since we heavily duty cycle the high power camera, its average current is reduced by a factor of 25 (1000 ms/40 ms). Since we do not perform compression on device, the radios require significant current to transmit the data to the basestation. We note that while one can reduce the wireless power consumption by using compression, compression adds to the power consumption of computing. One alternative to further reduce power consumption is to use backscatter, which is not in the scope of this work.

Table 2: Power consumption of different components in camera hardware. *This camera requires 27.52mA current for its operation, but this sensor is only active for 40 ms and hence is heavily duty cycled to reduce its average current.

5 Discussion and Conclusion

We introduce the first deep learning-based system that can capture video from low-power dual-mode IoT camera systems. Here, we discuss some of our design decisions.

Power asymmetry. We reduce the power consumption at the camera device while using a neural network decoder at the plugged-in router. The IoT devices are typically battery-
powered and hence are more power constrained than
the plugged-in routers. We exploit this asymmetry to reduce to
power consumption at the IoT device. We believe that this
design choice is reasonable given that 1) the plugged-in router
is not power constrained like the IoT camera, and 2) neural
accelerator chips are being developed to run neural networks
more energy-efficiently than using GPUs.

Compression. We note that our current prototype does not
perform compression. This is because it was not feasible to
implement MPEG compression on our target microcontrollers
at a frame rate of 15 fps. While custom ASICs might reduce
the power consumption of compression, this is complementa-
tory to our work. Future implementations can compress the
video at the IoT device using custom ASIC, thus, reducing the
communication bandwidth and potentially further reducing
the energy consumption of our system.

References

[1] High-speed vga image sensor for video, gaming and
portable entertainment. https://pdf1.alldatasheet.
et/datasheet-pdf/view/981148/OMNIVISION/OV7735.html.

[2] Hm01b0-mna-0f870 compact camera module. https://
cdn.sparkfun.com/assets/7/f/c/8/3/HM01B0-
MNA-Datasheet.pdf.

[3] Investigating power consumption of blink. https://
cam-do.com/blogs/camdo-blog/investigating-
power-consumption-of-blind.

[4] Control of noise and background in scientific cmos tech-
nology, teledyne imaging, 2019. https://www.phot-
omeometrics.com/wp-content/uploads/2019/11/C0
ntrol-of-Noise-and-Background-in-Scientifi
c-CMOS-Technology.pdf.

[5] Raspberry Pi Camera Module 2., 2022.

[6] M.J. Agan, B.H. Olson, C.R. Pasqualino, and G.L.
Stevens. A highly miniaturized, battery operated, com-
mandable, digital wireless camera. In IEEE Military
Communications Conference. Proceedings. MILCOM
98 (Cat. No.98CH36201), volume 3, pages 913–918
vol.3, 1998.

[7] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Ben-
gio. Neural machine translation by jointly learning to
align and translate. CoRR, abs/1409.0473, 2015.

[8] Andrea Bejarano-Carbo, Hyochan An, Kyojin Choo,
Shiyu Liu, Dennis Sylvester, David Blaauw, and Hun-
Seok Kim. Millimeter-scale ultra-low-power imaging
system for intelligent edge monitoring, 2022.

[9] Fernanda Brandi, Ricardo de Queiroz, and Debargha
Mukherjee. Super-resolution of video using key frames
and motion estimation. In 2008 15th IEEE International
Conference on Image Processing, pages 321–324, 2008.

[10] Jose Caballero, Christian Ledig, Andrew Aitken, Ale-
jandro Acosta, Johannes Totz, Zehan Wang, and Wen-
zhe Shi. Real-time video super-resolution with spatio-
temporal networks and motion compensation. In 2017
IEEE Conference on Computer Vision and Pattern
Recognition (CVPR), pages 2848–2857, 2017.

[11] Kelvin CK Chan, Xiantao Wang, Ke Yu, Chao Dong, and
Chen Change Loy. Basicvsr: The search for essential
components in video super-resolution and beyond. In
Proceedings of the IEEE conference on computer vision
and pattern recognition, 2021.

[12] Kelvin C.K. Chan, Shangchen Zhou, Xiangyu Xu, and
Chen Change Loy. Basicvsr++: Improving video super-
resolution with enhanced propagation and alignment. In
Proceedings of the IEEE conference on computer vision
and pattern recognition, 2022.

[13] Ranveer Chandra and Stewart Collis. Smart farming
with technologies such as iot, computer vision, and ai
can improve agricultural efficiency, transparency, prof-
ability, and equity for farmers in low-and middle-
income countries. COMMUNICATIONS OF THE ACM,
64(12):75–84, 2021.

[14] A. Chandrakasan, A. Dancy, J. Goodman, and T. Simon.
A low-power wireless camera system. In Proceedings
Twelfth International Conference on VLSI Design. (Cat.
No.PR00013), pages 32–36, 1999.

[15] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Bar-
laud. Two deterministic half-quadratic regularization
algorithms for computed imaging. In Proceedings of
1st International Conference on Image Processing, vol-
ume 2, pages 168–172 vol.2, 1994.

[16] Ming Cheng, Zhan Ma, M. Salman Asif, Yiling Xu, Hao-
jie Liu, Wenbo Bao, and Jun Sun. A dual camera sys-
tem for high spatiotemporal resolution video acquisition.
IEEE Transactions on Pattern Analysis and Machine In-
telligence, 43:3275–3291, 2021.

[17] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep
colorization. In 2015 IEEE International Conference
on Computer Vision (ICCV), pages 415–423, 2015.

[18] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong
Zhang, Han Hu, and Yichen Wei. Deformable convolu-
tional networks. In 2017 IEEE International Conference
on Computer Vision (ICCV), pages 764–773, 2017.
[19] Bert De Brabandere, Xu Jia, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS’16, page 667–675, Red Hook, NY, USA, 2016. Curran Associates Inc.

[20] Runmin Dong, Lixian Zhang, and Haohuan Fu. Rrsgan: Reference-based super-resolution for remote sensing image. IEEE Transactions on Geoscience and Remote Sensing, pages 1–17, 2021.

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. ICLR, 2021.

[22] Y. Endo, S. Iizuka, Y. Kanamori, and J. Mitani. Deep-prop: Extracting deep features from a single image for edit propagation. Computer Graphics Forum, 35, 2016.

[23] W.T. Freeman, T.R. Jones, and E.C. Pasztor. Example-based super-resolution. IEEE Computer Graphics and Applications, 22(2):56–65, 2002.

[24] Muhammad Haris, Greg Shakharovich, and Norimichi Ukita. Recurrent back-projection network for video super-resolution. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[25] Mingming He, Dongdong Chen, Jing Liao, Pedro V. Sander, and Lu Yuan. Deep exemplar-based colorization. ACM Trans. Graph., 37(4), jul 2018.

[26] Pan Hu, Junha Im, Zain Asgar, and Sachin Katti. Starfish: Resilient Image Compression for AIoT Cameras, page 395–408. Association for Computing Machinery, New York, NY, USA, 2020.

[27] Yan Huang, Wei Wang, and Liang Wang. Bidirectional recurrent convolutional networks for multi-frame super-resolution. In Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS’15, page 235–243, Cambridge, MA, USA, 2015. MIT Press.

[28] Edson Mintus Hung, Ricardo L. de Queiroz, Fernanda Brandi, Karen França de Oliveira, and Debargha Mukherjee. Video super-resolution using codebooks derived from key-frames. IEEE Transactions on Circuits and Systems for Video Technology, 22(9):1321–1331, 2012.

[29] Andrey D. Ignatov, Andrés Romero, Heewon Kim, Rada Timofte, Chiu Man Ho, Zibo Meng, Kyoung Mu Lee, Yuxiang Chen, Yutong Wang, Zeyu Long, Chen-hao Wang, Yifei Chen, Boshen Xu, Shuhang Gu, Lixin Duan, Wen Li, Wang Bofei, Zhang Diankai, Zheng Chengjian, Liu Shaoli, Gao Si, Zhang Xiaofeng, Luan Kaidi, Xu Tianyu, Zheng Hui, Xinbo Gao, Xiumei Wang, Jiaming Guo, Xueyi Zhou, Hao Jia, and Youliang Yan. Real-time video super-resolution on smartphones with deep learning, mobile ai 2021 challenge: Report. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 2535–2544, 2021.

[30] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Let there be color! joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. ACM Trans. Graph., 35(4), July 2016.

[31] Takashi Isobe, Xu Jia, Shuhang Gu, Songjiang Li, Shengjin Wang, and Qi Tian. Video super-resolution with recurrent structure-detail network. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, Computer Vision – ECCV 2020, pages 645–660, Cham, 2020. Springer International Publishing.

[32] S. Itoh, S. Kawahito, and S. Terakawa. A 2.6mw 2fps qvga cmos one-chip wireless camera with digital image transmission function for capsule endoscopes. In 2006 IEEE International Symposium on Circuits and Systems, pages 4 pp.–3356, 2006.

[33] Vikram Iyer, Ali Najafi, Johannes James, Sawyer Fuller, and Shyamnath Gollakota. Wireless steerable vision for live insects and insect-scale robots. Science Robotics, 5:eabb0839, 07 2020.

[34] Varun Jampani, Raghudeep Gadde, and Peter V. Gehler. Video propagation networks. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2017.

[35] Suyao Ji, Jing Pu, Byong Chan Lim, and Mark Horowitz. A 220pj/pixel/frame cmos image sensor with partial settling readout architecture. In 2016 IEEE Symposium on VLSI Circuits (VLSI-Circuits), pages 1–2, 2016.

[36] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 3224–3232, 2018.

[37] Colleen Josephson, Lei Yang, Pengyu Zhang, and Sachin Katti. Wireless computer vision using commodity radios. In 2019 18th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), pages 229–240, 2019.
[38] Med Lassaad Kaddachi, Leila Makkaoui, Adel Soudani, Vincent Leclaire, and Jean-Marie Moureaux. FPGA-based image compression for low-power wireless camera sensor networks. In 2011 3rd International Conference on Next Generation Networks and Services (NGNS), pages 68–71, 2011.

[39] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. International Conference on Learning Representations, 12 2014.

[40] Pang Wei Koh, Shiori Sagawa, Henrik Marklund, Sang Michael Xie, Marvin Zhang, Akshay Balsubramani, Weihua Hu, Michihiro Yasunaga, Richard Lanas Phillips, Irena Gao, Tony Lee, Etienne David, Ian Stavness, Wei Guo, Berton Earnshaw, Imran Haque, Sara M Beery, Jure Leskovec, Anshul Kundaje, Emma Pierson, Sergey Levine, Chelsea Finn, and Percy Liang. Wilds: A benchmark of in-the-wild distribution shifts. In Marina Meila and Tong Zhang, editors, Proceedings of the 38th International Conference on Machine Learning, volume 139 of Proceedings of Machine Learning Research, pages 5637–5664. PMLR, 18–24 Jul 2021.

[41] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 105–114, 2017.

[42] Junyong Lee, Myeonghee Lee, Sunghyun Cho, and Seunghyong Lee. Reference-based video super-resolution using multi-camera video triplets, March 2022, arxiv.

[43] Martin Lefebvre, Ludovic Moreau, Rémi Dekimpe, and David Bol. 7.7 a 0.2-to-3.6tops/w programmable convolutional imager soc with in-sensor current-domain ternary-weighted mac operations for feature extraction and region-of-interest detection. In 2021 IEEE International Solid-State Circuits Conference (ISSCC), volume 64, pages 118–120, 2021.

[44] Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. ACM Trans. Graph., 23(3):689–694, August 2004.

[45] Robert LiKamWa, Bodhi Priyantha, Matthai Philipose, Lin Zhong, and Paramvir Bahl. Energy characterization and optimization of image sensing toward continuous mobile vision. In Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys ’13, page 69–82, New York, NY, USA, 2013. Association for Computing Machinery.

[46] Ce Liu and Deqing Sun. On bayesian adaptive video super resolution. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(2):346–360, 2014.

[47] Ding Liu, Zhaowen Wang, Yuchen Fan, Xianming Liu, Zhangyang Wang, Shiyu Chang, and Thomas Huang. Robust video super-resolution with learned temporal dynamics. In 2017 IEEE International Conference on Computer Vision (ICCV), pages 2526–2534, 2017.

[48] Xinqiao Liu, Abbas El Gamal, Mark A. Horowitz, and Brian A. Wandell. CMOS image sensors dynamic range and SNR enhancement via statistical signal processing, stanford ph.d. thesis, 2002.

[49] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. DVC: An end-to-end deep video compression framework. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 10998–11007, 2019.

[50] Qing Luan, Fang Wen, Daniel Cohen-Or, Lin Liang, Ying-Qing Xu, and Heung-Yeung Shum. Natural image colorization. In Proceedings of the 18th Eurographics Conference on Rendering Techniques, EGSR’07, page 309–320, Goslar, DEU, 2007. Eurographics Association.

[51] Sachin Mehta and Mohammad Rastegari. Mobilevit: Light-weight, general-purpose, and mobile-friendly vision transformer. CoRR, abs/2110.02178, 2021.

[52] Simone Meyer, Victor Cornillère, Abdelazziz Djelouah, Christopher Schroers, and Markus Gross. Deep video color propagation. In Proceedings of the British Machine Vision Conference BMVC, 2018.

[53] Simone Meyer, Alexander Sorkine-Hornung, and Markus Gross. Phase-based modification transfer for video. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, Computer Vision – ECCV 2016, pages 633–648, Cham, 2016. Springer International Publishing.

[54] Fukashi Morishita, Norihito Kato, Satoshi Okubo, Takao Toi, Mitsuru Hiraki, Sugako Otani, Hideaki Abe, Yuji Shinohara, and Hiroyuki Kondo. A CMOS image sensor and an AI accelerator for realizing edge-computing-based surveillance camera systems. In 2021 Symposium on VLSI Circuits, pages 1–2, 2021.

[55] Saman Naderiparizi, Mehrdad Hessar, Vamsi Talla, Shyamnath Gollakota, and Joshua R. Smith. Towards battery-free HD video streaming. In Proceedings of the 15th USENIX Conference on Networked Systems Design and Implementation, NSDI’18, page 233–247, USA, 2018. USENIX Association.
[56] Saman Naderiparizi, Aaron N. Parks, Zerina Kapetanovic, Benjamin Ransford, and Joshua R. Smith. Wispcam: A battery-free rfid camera. In 2015 IEEE International Conference on RFID (RFID), pages 166–173, 2015.

[57] Saman Naderiparizi, Pengyu Zhang, Matthai Philipose, Bodhi Priyantha, Jie Liu, and Deepak Ganesan. Glimpse: A programmable early-discard camera architecture for continuous mobile vision. In Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys ’17, page 292–305, New York, NY, USA, 2017. Association for Computing Machinery.

[58] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In CVPR Workshops, June 2019.

[59] Seyedreza Najafi and Shahram Shirani. Regularization function for video super-resolution using auxiliary high resolution still images. In 2012 Conference Record of the Forty Sixth Asilomar Conference on Signals, Systems and Computers (ASILOMAR), pages 1713–1717, 2012.

[60] Nuo Nan, Haian Zhou, Xin Li, Qi Tang, Xingtai Feng, and Yuan Yuan. Design of a light and high-resolution video camera system based on CMV12000 sensor. In Junhong Su, Lianghui Chen, Junhao Chu, Shining Zhu, and Qifeng Yu, editors, Eighth Symposium on Novel Photoelectronic Detection Technology and Applications, volume 12169, pages 2619 – 2625. International Society for Optics and Photonics, SPIE, 2022.

[61] Lorenzo Piccardi, Basilio Noris, Olivier Barbey, Aude Billard, Giuseppina Schiavone, Flavio Keller, and Claes von Hofsten. Wearcam: A head mounted wireless camera for monitoring gaze attention and for the diagnosis of developmental disorders in young children. In ROMAN 2007 - The 16th IEEE International Symposium on Robot and Human Interactive Communication, pages 594–598, 2007.

[62] Oren Rippel, Sanjay Nair, Carissa Lew, Steve Branson, Alexander G. Anderson, and Lubomir D. Bourdev. Learned video compression. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 3453–3462, 2019.

[63] Davide Rossi, Francesco Conti, Manuel Eggiman, Stefan Mach, Alfio Di Mauro, Marco Guermandi, Giuseppe Tagliavini, Antonio Pullini, Igor Loi, Jie Chen, Eric Flamand, and Luca Benini. 4.4 a 1.3tops/w @ 32gops fully integrated 10-core soc for iot end-nodes with 1.7uw cognitive wake-up from mram-based state-retentive sleep mode. In 2021 IEEE International Solid-State Circuits Conference (ISSCC), volume 64, pages 60–62, 2021.

[64] Ali Saffari, Mehrdad Hessar, Saman Naderiparizi, and Joshua R. Smith. Battery-free wireless video streaming camera system. In 2019 IEEE International Conference on RFID (RFID), pages 1–8, 2019.

[65] Mehdi S. M. Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 6626–6634, 2018.

[66] Patsorn Sangkloy, Jingwan Lu, Chen Fang, Fisher Yu, and James Hays. Scribbler: Controlling deep image synthesis with sketch and color. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 6836–6845, 2017.

[67] Wenxie Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1874–1883, 2016.

[68] Gyumin Shim, Jinsun Park, and In So Kweon. Robust reference-based super-resolution with similarity-aware deformable convolution. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 8422–8431, 2020.

[69] Mehrdad Khani Shirkoohi, Vibhaalakshmi Sivaraman, and Mohammad Alizadeh. Efficient video compression via content-adaptive super-resolution. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 4501–4510, 2021.

[70] Sander Smets, Toon Goedemé, Anurag Mittal, and Marian Verhelst. 2.2 a 978gops/w flexible streaming processor for real-time image processing applications in 22nm fdsoi. In 2019 IEEE International Solid-State Circuits Conference - (ISSCC), pages 44–46, 2019.

[71] Byung Cheol Song, Shin-Cheol Jeong, and Yanglim Choi. Video super-resolution algorithm using bi-directional overlapped block motion compensation and on-the-fly dictionary training. IEEE Transactions on Circuits and Systems for Video Technology, 21(3):274–285, 2011.

[72] Vamsi Talla, Bryce Kellogg, Benjamin Ransford, Saman Naderiparizi, Shyamnath Gollakota, and Joshua Smith. Powering the next billion devices with wi-fi. In ACM CONEXT, pages 1–13, 12 2015.
[73] Yapeng Tian, Yulun Zhang, Yun Fu, and Chenliang Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[74] Tengfei Wang, Jiaxin Xie, Wenxiu Sun, Qiong Yan, and Qifeng Chen. Dual-camera super-resolution with aligned attention modules. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pages 1981–1990, 2021.

[75] Xintao Wang, Kelvin C.K. Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2019.

[76] Yingqian Wang, Xinyi Ying, Longguang Wang, Jun-gang Yang, Wei An, and Yulan Guo. Symmetric parallax attention for stereo image super-resolution. CoRR, abs/2011.03802, 2020.

[77] Yong Wang, Dianhong Wang, Xufan Zhang, Jun Chen, and Yamin Li. Energy-efficient image compressive transmission for wireless camera networks. IEEE Sensors Journal, 16(10):3875–3886, 2016.

[78] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, 2004.

[79] Chao-Yuan Wu, Nayan Singhal, and Philipp Krähenbühl. Video compression through image interpolation. In ECCV, 2018.

[80] Han Xu, Ziru Li, Ningchao Lin, Qi Wei, Fei Qiao, Xunzhao Yin, and Huazhong Yang. Macsen: A processing-in-sensor architecture integrating mac operations into image sensor for ultra-low-power bnn-based intelligent visual perception. IEEE Transactions on Circuits and Systems II: Express Briefs, 68(2):627–631, 2021.

[81] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T Freeman. Video enhancement with task-oriented flow. International Journal of Computer Vision (IJCV), 127(8):1106–1125, 2019.

[82] Peng Yi, Zhongyuan Wang, Kui Jiang, Junjun Jiang, and Jiayi Ma. Progressive fusion video super-resolution network via exploiting non-local spatio-temporal correlations. In IEEE International Conference on Computer Vision (ICCV), pages 3106–3115, 2019.

[83] Pengyu Zhang, Pan Hu, Vijay Pasikanti, and Deepak Ganesan. Ekhonet: High-speed ultra low-power backscatter for next generation sensors. GetMobile Mob. Comput. Commun., 19:14–17, 2015.

[84] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. ACM Transactions on Graphics (TOG), 9(4), 2017.

[85] Zhifei Zhang, Zhaowen Wang, Zhe Lin, and Hairong Qi. Image super-resolution by neural texture transfer. pages 7974–7983, 06 2019.

[86] Haitian Zheng, Mengqi Ji, L. Han, Z. Xu, Haoqian Wang, Yebin Liu, and Lu Fang. Learning cross-scale correspondence and patch-based synthesis for reference-based super-resolution. In BMVC, 2017.

[87] Haitian Zheng, Mengqi Ji, Haoqian Wang, Yebin Liu, and Lu Fang. Crossnet: An end-to-end reference-based super resolution network using cross-scale warping. In Proceedings of the European Conference on Computer Vision (ECCV), pages 88–104, 2018.

[88] X. Zhu, H. Hu, S. Lin, and J. Dai. Deformable convnets v2: More deformable, better results. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9300–9308, Los Alamitos, CA, USA, Jun 2019. IEEE Computer Society.