# Accelerating Vision Task Processing with Sensor RAW Images

Anonymous ICCV submission

Paper ID 3769

## Abstract

*This work explores the feasibility to directly execute high-level vision tasks using the raw sensor data (i.e., RAW image), by which we can completely bypass the Image Signal Processor (ISP) devised in digital cameras for decades. Due to the lack of large-scale RAW image datasets to train robust RAW-domain vision models, we have developed a generative adversarial network (GAN) to transform existing abundant JPEG compressed RGB images to corresponding RAW-domain representations. We have also collected and labelled a small-scale RAW image dataset dedicated for high-level detection and segmentation tasks. This is the first dataset of its kind that can be also used with GAN-based learning to capture the characteristics of real RAW images. Experiments demonstrate the competitive efficiency of RAW-domain object detection and semantic segmentation over the conventional paradigm using RGB images. In addition, performing vision tasks using RAW images removes the need for ISP, which makes it attractive for devices and applications with stringent power and latency constraints. We will make our dataset and relevant material publicly accessible for reproducible research.*

## 1. Introduction

Computer vision (CV) enables high-level understanding of digital images and videos that are mostly processed in the RGB format. This is probably because many existing CV algorithms mimic the cortical process of human visual system (HVS) to perceive the colors (e.g., red/green/blue light radiance sensation of retinal cells) and understand the scene for decision making [15]. Digital cameras have separated the imaging sensor and the ISP to enable different functionality [3], where a standard CMOS or CCD sensor first aggregates light photons as the RAW image to characterize the photometric attributes of captured scene, and then the Image Signal Processor (ISP) converts input RAW image into corresponding HVS perceivable RGB image through a series of linear and nonlinear steps (e.g., demosaicing, white balance, exposure control, gamma correction, compression). Subsequently, these RGB images are often processed by certain CV algorithms for appropriate task execution (e.g., classification, object detection, segmentation). Such classical RGB-domain CV processing pipeline can be simply represented as

$$y = \mathbb{T}\left(\mathbb{I}\left(x_{raw}\right)\right), \tag{1}$$

with $\mathbb{I}(\cdot)$ denotes the ISP function, $\mathbb{T}(\cdot)$ denotes a specific CV task, $x_{raw}$ and $y$ denote the RAW image and task-specific outcome, respectively. This conventional paradigm has been massively deployed in existing large-scale image/video analytic infrastructure, such as event detection and traffic monitoring [19].

Recent years have witnessed the explosive growth of CV applications. One exciting example is the autonomous cars that have multiple video cameras and other sensors (e.g., ultrasonic, LiDAR, and radar) equipped for event detection, recognition, and decision-making. Although conventional RGB-domain CV engines still dominate in practice, they pose critical challenges for ultra low-latency applications since *delay* is inevitably introduced by the serial transformations in the ISP module [13]. It is generally reported that the autonomous cars generally require less than 0.1s for prompt event response. Taking an On-semi ISP ( https://www.onsemi.com/) chip as an example, it requires approximately 0.033s to transform an input RAW image to its corresponding RGB representation, which is nearly 33% of the response time and imposes a critical impact in applications.

Aforementioned issue can be largely resolved if we remove the ISP subsystem by directly processing sensor RAW images for tasks. This raises a fundamental question: can we maintain the same vision task efficiency in RAW domain as in the conventional RGB domain? To address this question, this paper attempts to accelerate the execution of vision tasks using sensor RAW images directly. We test our proposed methods on two prevalent CV tasks, namely object detection and semantic segmentation, to demonstrate the efficiency and competitive accuracy compared to conventional methods using RGB images converted by ISP. Marginal accuracy drop (nearly $1\%$) may come from the

fact that CV algorithms used for demonstration have been extensively optimized for RGB content rather than the RAW data. We expect better performance if these CV algorithms can be refined using a large number of RAW samples. This is deferred as our future study.

In order to perform RAW-domain tasks, we first collected 1153 RAW images using iPhone XS Max, which we call iPhone RAW Scape 1k (iRAW). The particular focus of the collected data was on autonomous car driving applications. This dataset is later manually labelled with detection bounding boxes and segmentation cues with the help of a third-party professional image labelling firm. Nearly 1000 samples are insufficient for training a robust neural model in practice; therefore, the iRAW dataset is primarily used for testing/inferenc and knowledge transferring in RAW-domain fine-tuning .

Even though we have several image datasets (e.g., ImageNet) for various vision tasks, they are stored in RGB format and cannot be used for RAW processing directly. Given the nonlinear transformations involved in camera ISP, we propose to approximate the inverse functions in camera ISP, noted as "invISP", and another mirroring "neural-ISP" as the reverse process of invISP, of which we follow the CycleGAN [23] architecture to well capture the cross-domain characteristics. Since most RGB images are JPEG compressed for storage, we also include the compression artifacts removal and quality estimation into the training to improve the model robustness. For simplicity, we use the term "JPEG" or simply "RGB" to represent decoded RGB image with compression noise.

In the end, the GAN-based invISP works in an unsupervised manner by mapping an RGB source into its corresponding RAW image, while using randomly selected real-life RAW images from in iRAW as a discriminator to efficiently captures the cross-domain (e.g., RGB to RAW) characteristics for a target camera. Such a GAN-based invISP demonstrates an encouraging efficiency in processing and can potentially help resolve the shortage of RAW images for model training.

**Contributions.**

1) To the best of our knowledge, this iRAW is the first sensor RAW image dataset with high-level detection and segmentation labels, which is different from existing sensor RAW images used for low-level denoising [21, 2] and KITTI raw images[1] (uncompressed RGB images). We will continue acquiring RAW images from various scenarios and camera models to expand this iRAW dataset.

2) Our proposed GAN-based invISP simulates the inverse ISP function in an unsupervised fashion, which helps us generate sufficient training samples using existing RGB dataset to retrain the YOLOv5 [16] for object detection and HRNetv2 [17] for semantic segmentation. Experimental results show that our method achieves comparable accuracy to the native solutions that use RGB images, while reducing latency and power consumption since our RAW-domain processing completely bypasses the ISP.

[1] http://www.cvlibs.net/datasets/kitti/raw_data.php

## 2. Related Work

This section briefly reviews the techniques involved in traditional ISP, and RAW image processing.

### 2.1. Conventional ISP Basics

A serial processing steps have been involved in modern ISPs to convert sensor RAW data to final RGB representation, as shown in Figure 1(a). First, linear transformations including the demosaicing, white balance estimation, color contrast adjustment, etc., are applied to convert the native input RAW data to image format conforming to the CIE 1931 XYZ color space [9]. Subsequently, nonlinear transforms are facilitated to translate the image from the XYZ domain to RGB color space, mimicking the nonlinear processing capabilities of the HVS. Well-known nonlinear operations include the denoising, exposure control, gamma correction. Then, JPEG or MPEG based coder is used to reduce the size of RGB images for storage or transmission. This processing pipeline is widely used in modern cameras as the ISP subsystem.A brief white paper on ISP system can also be found here [2].

### 2.2. RAW Image Processing

Though most algorithms for image processing have been applied on RGB images, recent years have shown a number of explorations on RAW image processing [10, 2, 21, 1]. For instance, Zamir *et al.* [21] proposed a CycleISP to remap the RGB image to its RAW presentation for denoising, by which better image quality was achieved than that in RGB domain. Recently, Afifi *et al.* [2] demonstrated better results for various low-level tasks including denoising, color correction, super-resolution, etc., using RAW images that were reversely restored from the input RGB samples. These studies have promised encouraging prospect of RAW imaging processing, arising the desire of original RAW images.

Since there is not sufficient real-life RAW images to train CV models, a straightforward way is to employ the inverse ISP functions to reversely transform the existing abundant RGB images into corresponding RAW samples [10, 21, 2, 4, 14]. But most algorithms like "reverse imaging pipeline" (a.k.a., InvGamma) [10] only utilized the inverse gamma correction to simulate the nonlinear function and overlooked other important nonlinear processing

[2] https://www.pathpartnertech.com/camera-tuning-understanding-the-image-signal-processor-and-isp-tuning/

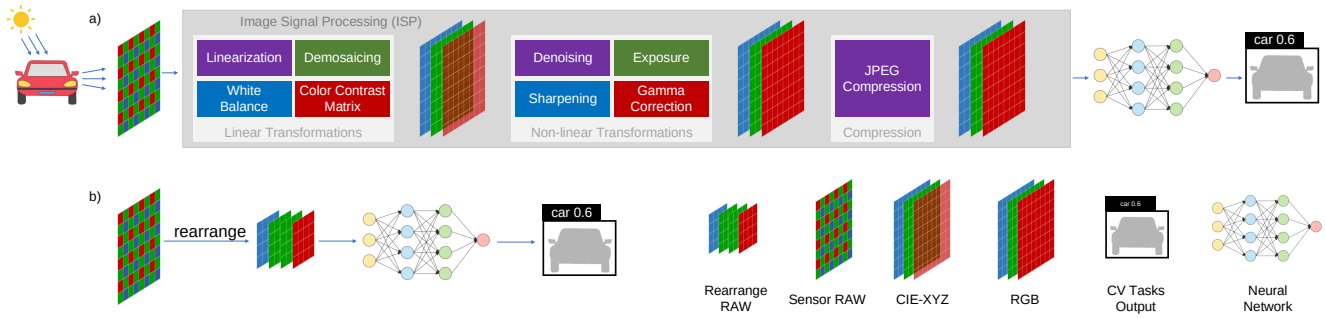


Figure 1. **Camera Pipeline for High-Level Vision Tasks.** (a) Traditional RGB-domain CV from senor RAW imaging, ISP processing to task execution on RGB samples; (b) Proposed RAW-domain CV by removal of ISP and task execution on RAW images directly.

in camera ISP. With the help of deep learning techniques, CycleISP [21] and CIE-XYZ Net [2] suggested to learn inverse ISP functions through a supervised approach. As a result, a large amount of RAW and RGB image pairs are required to supervise the learning and the learnt model will be camera ISP dependent, making it difficult to generalize itself to various camera ISP models.

## 3. Methods

This section describes our exploration in RAW domain to perform the vision tasks, the dataset generation, and the inverse ISP development.

### 3.1. Problem Formulation

As illustrated in Figure 1(a), the conversion of a RAW image $x_{raw} \in \mathrm{R}^{h \times w}$ to JPEG coded RGB representation[3] $x_{jpeg} \in \mathrm{R}^{h \times w \times 3}$ can be divided into three steps: linear transformation, nonlinear transformation, and JPEG compression. Here $h$ and $w$ represent the image height and width, respectively. We can represent mapping between $x_{raw}$ and $x_{jpeg}$ as the following steps:

$$x_{xyz} = \mathbb{L}\left(x_{raw}\right), \; x_{rgb} = \mathbb{N}\left(x_{xyz}\right), \; x_{jpeg} = \mathbb{J}\left(x_{rgb}\right),$$

where $\mathbb{L}$ is a linear transformation from RAW to CIE-XYZ image $x_{xyz} \in \mathrm{R}^{h \times w \times 3}$, $\mathbb{N}$ is a nonlinear trasnform that maps CIE-XYZ to RGB image, and $\mathbb{J}$ denotes the JPEG compression to reduce the size of native RGB images for efficient storage and transmission. Since different cameras can have different types of sensors and ISPs, the processing pipeline for a specific camera $c$ can be represented as

$$x_{jpeg_c} = \mathbb{J}\left(\mathbb{N}_c\left(\mathbb{L}_c\left(x_{raw_c}\right)\right)\right). \tag{2}$$

In practice, RAW image acquisition is dependent on camera/ISP; ideally we would like to assure the same camera/ISP for the training and testing. However, existing RGB datasets that have been captured by different cameras do not offer explicit labels of their ISPs. Therefore, it is difficult to map these RGB images back to their original RAW data. Without the loss of generality, let $s$ and $t$ represent the source camera for capturing RGB datasets and the target camera for inferring the RAW samples, respectively. The process of reconstructing the RAW image from a JPEG image for a target camera can be written as

$$\tilde{x}_{raw_t} = \mathbb{L}_t^{-1}\left(\mathbb{N}_t^{-1}\left(\mathbb{C}_{s \to t}\left(\mathbb{J}^{-1}\left(x_{jpeg_s}\right)\right)\right)\right). \tag{3}$$

$\mathbb{C}_{s \to t}(\cdot)$ describes cross-domain adaptation from the RGB image of source camera to that of the target camera. $\tilde{x}_{raw}$ denotes a simulated version of $x_{raw}$. Both $\mathbb{C}_{s \to t}(\cdot)$ and $\mathbb{N}_t^{-1}(\cdot)$ are non-linear operations. For simplicity, we merge them using a single nonlinear function $\mathbb{N}_{t \to s}^{-1}$:

$$\tilde{x}_{raw_t} = \mathbb{L}_t^{-1}\left(\mathbb{N}_{t \to s}^{-1}\left(\mathbb{J}^{-1}\left(x_{jpeg_s}\right)\right)\right). \tag{4}$$

We can use the function in (4) to map any RGB image to its corresponding RAW representation, and train a task model $\mathbb{T}\left(\theta_{\texttt{optimal}}\right)$ that processes RAW images in inference, but uses RGB images in training, i.e.,

$$\tilde{y} = \mathbb{T}\left(\theta,\, \mathbb{L}_t^{-1}\left(\mathbb{N}_{t \to s}^{-1}\left(\mathbb{J}^{-1}\left(x_{jpeg_s}\right)\right)\right)\right). \tag{5}$$

Ideally we can optimize the network parameters $\theta_{\texttt{optimal}}$ by solving the following optimization problem:

$$\theta_{\texttt{optimal}} = \arg\min_{\theta_{\texttt{optimal}}} \|\tilde{y} - y\|. \tag{6}$$

The inference process after the training can be described as

$$\tilde{y} = \mathbb{T}\left(\theta_{\texttt{optimal}}, x_{raw_t}\right). \tag{7}$$

### 3.2. Learned invISP

We propose a CycleGAN [23]-based approach for self-supervised domain transform. CycleGAN [23] requires two generators for source-to-target and target-to-source domain transformations, and two discriminators (one each in the

[3] We exemplify our method using JPEG because most RGB datasets are coded using popular JPEG standard.

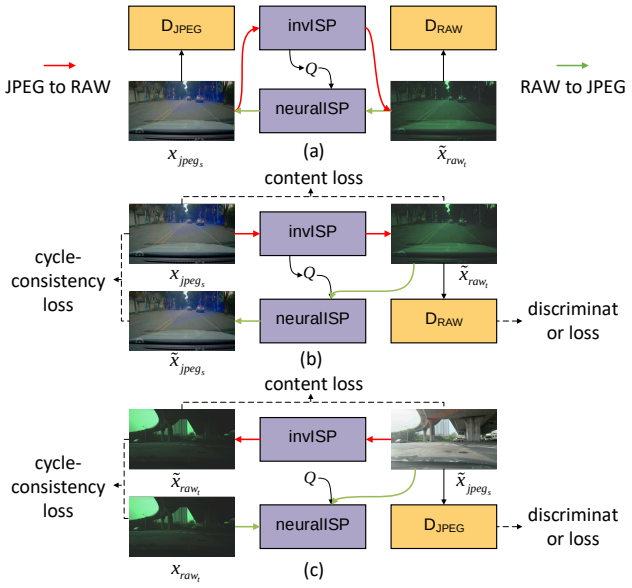


Figure 2. **The GAN-based invISP**: (a) Our model follows the CycleGAN [23] architecture, having respective JPEG to RAW mapping and RAW to JPEG mapping, as well as associated adversarial discriminators $D_{raw}$ and $D_{jpeg}$. Cycle consistency loss is used to assure the cross-domain consistency, *discriminator loss* forces generated image to the real image as close as possible, and content loss is to maximize the context similarity between input source and generated output. The quality factor $Q$ estimated from JPEG is used to help neuralISP adjust the compression level. (b) Forward pipeline start from JPEG: $x_{jpeg_s} \rightarrow$ invISP $(\tilde{x}_{jpeg_s}) \rightarrow x_{raw_t} \rightarrow$ neuralISP $(\tilde{x}_{raw_t}) \rightarrow \tilde{x}_{jpeg_s}$. (c) Backward pipeline start from RAW: $x_{raw_t} \rightarrow$ invISP $(x_{raw_t}) \rightarrow \tilde{x}_{jpeg_s} \rightarrow$ neuralISP $(\tilde{x}_{jpeg_s}) \rightarrow \tilde{x}_{raw_s}$. The $Q$ in the backward process remains the same as the forward process. It is worth mentioning that the $x_{raw}$ mentioned in our paper is bilinearly upsampled from mosaic $x_{raw\_mosaic}$ to maintain a consistent resolution with $x_{rgb}$.

source and target domain) to distinguish between real and generated images. In particular, as shown in Figure 2, we need an invISP to convert JPEG coded RGB image to its RAW format, a neuralISP to convert a RAW image to its JPEG coded representation, one discriminators to distinguish between $x_{jpeg_s}$ and $\tilde{x}_{jpeg_s}$, and the other discriminator to distinguish between $x_{raw_t}$ and $\tilde{x}_{raw_t}$. To make the generated JPEG images closer to the original JPEG images, we use a quality estimation network to estimate the quality factor $Q$.

Following the modular decomposition in (4), we have five sub-networks: linear transformation network, nonlinear transformation network, JPEG artifacts removal network, discriminator, and quality estimation network, as illustrated in Fig. 3 to fulfill the invISP and neuralISP under the CycleGAN framework.

Our linear transformation network is inherited from the CIE-XYZ Net [2], having input at a fixed size of $128 \times 128$. It includes five blocks of $3 \times 3$ conv - LReLU - $2 \times 2$ max pooling layers. Different from CIE-XYZ Net [2], we use reflection padding instead of zero padding to avoid information loss. The max pooling layers have a stride factor of 2 used for down-sampling. Then FCN with 18 output is used to formulate 3x6 color matrix to perform a linear transformation.

Our nonlinear transformation network and quality estimation network are modified from CA-Unet [7, 12], which consists of channel attention-convolutions (CA-Convs) blocks. CA-Convs block first uses global pooling to extract spatial information from convolutional features, and then transforms them via fully connected (FC) layers, ReLU, and sigmoid. Finally, it multiplies the convolutional features with sigmoid's output, which represent the channel attention's weights. For nonlinear transformation network, the number of output channels is set to 6 to interface with the linear transform network. While for quality estimation network, we added the average pooling layer after the output layer with only one channel to perform the compression quality factor estimation.

The JPEG artifacts removal network is adopted from QGCN [11], which consists of restoration branch and global branch. The restoration branch consists of several residual blocks for extracting the local features and the global branch is used to extract the global features. Both of them are merged together in the middle of the restoration branch. Different with QGCN, we drop out the input qtables because we found that the network can adjust the noise reduction adaptively without applying prior knowledge of qtables.

Finally, our discriminator directly adopts the widely used SA-GAN [22], which allows attention-driven, long-range dependency modeling for image generation tasks. We believe that self-attention can be paired with our nonlinear transform network to treat different objects in the scene differently.

### 3.3. RAW-domain Vision Tasks

After obtaining the invISP, we can convert $x_{jpeg_s}$ from any JPEG dataset to simulated RAW image $\tilde{x}_{raw_t}$ to train RAW-domain vision models.

Note that image resolution impacts the the computer vision task. This is because if the resolution is not sufficient large, small objects may be missed by the algorithm. Therefore, for the detection and segmentation tasks, we integrate the upsampling of RAW data re-arrangement for algorithm input, e.g., $\tilde{x}_{up_t} = bilinear\_upsample(mosaic(\tilde{x}_{raw_t}))$. In this work, we simply use a bilinear filter to avoid any potential performance impacts from the advanced upsampling filters, particularly for those learnt approaches. This also ensures the fair comparison of task-oriented algorithms on

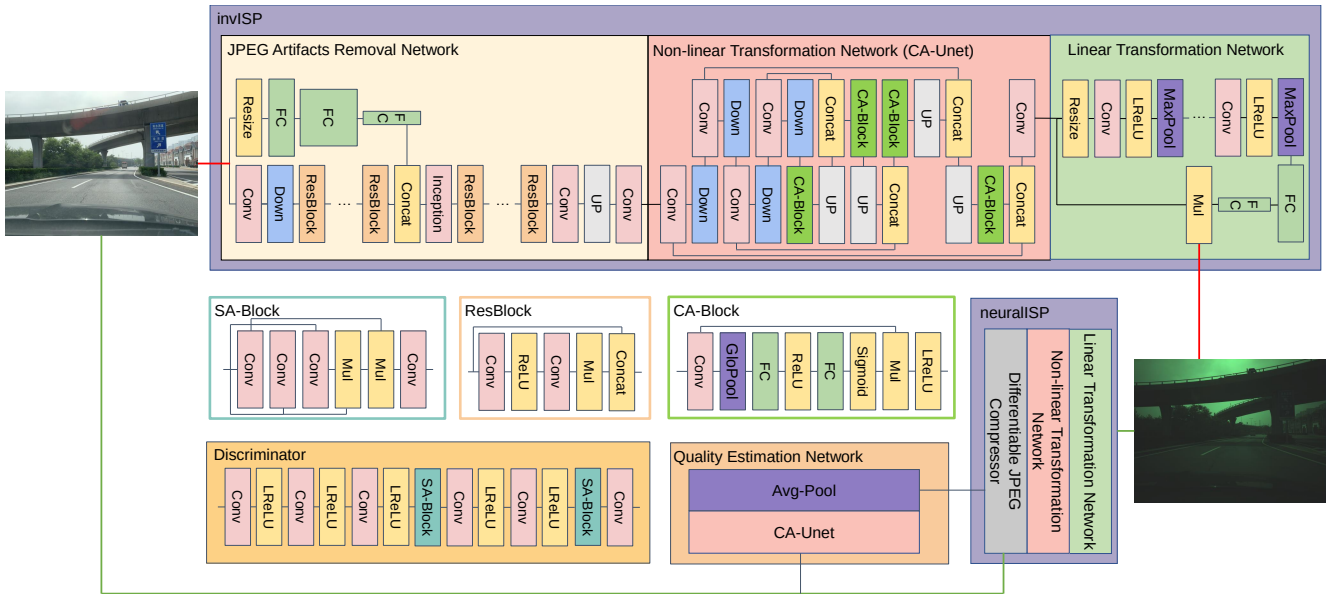


Figure 3. The architecture of our invISP and the corresponding discriminator, neuralISP. The red line indicates the process from JPEG to RAW, the green line indicates the process from RAW to JPEG.

different input sources (e.g., RGB versus RAW image). As mentioned above, we use the famous YOLOv5 [16] to perform object detection. We perform segmentation using the HRNetv2 [17].

### 3.4. Loss Function

Under the CycleGAN framework, we attempt to simultaneously optimize the restoration quality of both generated RAW image and generated (JPEG coded) RGB image. Towards this goal, we respectively apply the cycle consistency loss for measuring the cross-domain discrepancy and content loss for context similarity between input source and generated output. More specifically, we use $L_1$ as cycle consistency loss and the well-known VGG [8] based loss (e.g., $\mathrm{L_{vgg}} = VGG_{19}\left(x', x_{xyz_g}\right)$) as the content loss.

In the meantime, we wish to have the JPEG-converted RAW output close to the RAW images captured by a target camera, and the RAW-converted JPEG close to the original JPEG in native dataset. Thus we introduce an adversarial loss function to minimize the cross-domain discrepancy between source and target images, which can be written as

$$\mathrm{L}_{adv} = \mathbb{E}\left[1 - \log\left(\mathrm{D_{RAW}}\left(\tilde{x}_{raw_t}\right)\right)\right] \quad (8)$$
$$+ \mathbb{E}\left[1 - \log\left(\mathrm{D_{JPEG}}\left(\tilde{x}_{jpeg_s}\right)\right)\right]. \quad (9)$$

It then leads to the total loss function of the generator as

$$\mathrm{L}_g = \mathrm{L}_1 + \lambda_{\mathrm{vgg}} \cdot \mathrm{L}_{\mathrm{vgg}} + \lambda_{adv} \cdot \mathrm{L}_{\mathrm{adv}}. \quad (10)$$

For the discriminator network, we need to distinguish the $\tilde{x}_{raw_t}$ from the source domain and the $x_{raw_t}$ from the target domain. In this work, $x_{raw_t}$ is captured using iPhone Xs Max, and randomly selected for tests in the discriminator. We can optimize the discriminator using cross entropy loss, given as

$$\mathrm{L}_d = \mathbb{E}\left[\log\left(\mathrm{D_{RAW}}\left(\tilde{x}_{raw_t}\right)\right)\right] \quad (11)$$
$$+ \mathbb{E}\left[1 - \log\left(\mathrm{D_{RAW}}\left(x_{raw_t}\right)\right)\right] \quad (12)$$
$$+ \mathbb{E}\left[\log\left(\mathrm{D_{JPEG}}\left(\tilde{x}_{jpeg_s}\right)\right)\right] \quad (13)$$
$$+ \mathbb{E}\left[1 - \log\left(\mathrm{D_{JPEG}}\left(x_{jpeg_s}\right)\right)\right]. \quad (14)$$

The loss function of YOLOV5 consists of objectness score, class probability score, and the bounding box regression score. The detection loss are binary cross-entropy with Logits loss for class probability and object score. GIOU loss is used for the loss calculation of bounding box.

The segmentation loss $\mathrm{L}_{seg}$ is widely used, and could be formulated as the pixel-wise cross-entropy loss, given as

$$\mathrm{L}_{seg} = -\sum_{h=1}^{H}\sum_{w=1}^{W}\sum_{c=1}^{C} y^i \log\left(T\left(\theta_{seg},\ \tilde{x}^i_{up_t}\right)\right), \quad (15)$$

where $H$ and $W$ denote the height and width of the input image, and $C$ is the number of segmentation classes.

## 4. Experiments

We performed comprehensive experiments to demonstrate the efficiency of vision tasks execution in the RAW domain.

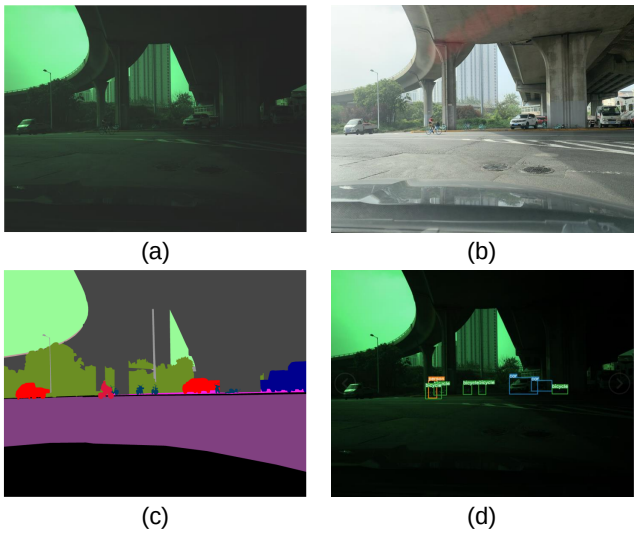


Figure 4. Examples of iRAW dataset. (a) RAW image. (b) RGB image. (c) Visualized semantic segmentation. (d) Bounding box. *Zoomed in for better details.*

### 4.1. Datasets

**iPhone RAW Scape 1K.** To study the RAW-domain vision tasks, we created a RAW-domain verification dataset, iPhone RAW Scape 1K (iRAW). The entire dataset was shot using the iPhone XS Max mobile phone fixed on the dashboard in a car to simulate the autonomous driving scenario. We collected 1153 RAW images at $4k \times 3k$ resolution and generated corresponding RGB images using built-in ISP. We manually labeled all detection bounding boxes, including car, person, bicycle, and motorcycle. We marked all the images with panoramic segmentation including sixteen categories of road, sidewalk, building, wall, fence, pole, traffic light, traffic sign, plant, sky, person, rider, car, truck, bus, and cylce (which includes bicycle, motorcycle, tricycle). Figure 4 shows an example image from our dataset. For object detection, we randomly selected 10% RAW images as the test set and used the remaining to fine-tune the YOLOv5 model. Since the amount of data is sufficient for full-scene segmentation, we train and test directly on iRAW.

**BDD & D$^2$-City.** We collected nearly 1K RAW images in iRAW, but they are insufficient to train a reliable detection network model. Therefore, we propose to convert the existing large-scale RGB datasets such as BDD 100K [20], which contains 100,000 720p30FPS (frame per second) videos, into RAW-domain representation for pre-training. It covers the image data captured from a variety of US major metro areas including the New York City, San Francisco, and Bay area. It also includes various scenes such as the city street, residential area, and highway. In total, we retrieve 79,326 key frames provided by BDD as training images. Additionally, we use the D$^2$-City [5], which is another large-scale auto driving video dataset, providing more than 10,000 forward-looking video data recorded by the dashcam in China cities. All videos are recorded in either 720p or 1080p resolution at 30 FPS. Note that D$^2$-City provides frame by frame annotations for nearly 1000 videos, including target frame location, target category and tracking ID information, covering a total of 12 categories. Since D$^2$-City did not provide the key frames of videos, we use inter frame similarity algorithm to extract the 24,706 key frames. Finally, we combine BDD 100k and D$^2$-City as our training dataset.

| Metrics | FID↓ | | MOS↑ | |
|---|---|---|---|---|
| Dataset | BDD | D$^2$-City | BDD | D$^2$-City |
| RGB | 155.69 | 197.98 | 0.0 | 0.0 |
| InvGamma [10] | 107.48 | 110.96 | 1.1 | 1.3 |
| CIE-XYZ Net [2] | 90.68 | 133.86 | 2 | 2.6 |
| CycleISP [21] | 93.09 | 106.28 | 3.6 | 3.2 |
| Our invISP | **83.37** | **92.56** | **4** | **4.1** |

Table 1. The FID & MOS results of InvGamma [10], CIE-XYZ Net [2], CycleISP [21] and our GAN-based invISP on BDD and D$^2$-City.

### 4.2. Training Details

**Learned invISP Network.** Because of the significant differences between BDD and D$^2$-City, we train two separate GANs for the two datasets respectively. The linear transformation network and the nonlinear transformation network were pre-trained for 300 epochs on the iRAW dataset. Meanwhile, w We randomly generated the JPEG compressed RGB images on the iRAW dataset using the quality factor $Q \in [1, 100]$, and the JPEG artifacts removal network was pre-trained for 200 epochs on the paired data. Adam optimizer with batch size 2 is applied for training, and the initial learning rate is set to 0.00005 and 0.0001 for the generator and discriminator, respectively. We set the total number of iteration to 100k. $\lambda_{vgg}$ is set to 0.01, and $\lambda_{adv}$ is set to 10.

**Object Detection Network.** We deploy YOLOv5 [6] as the baseline model. The input image is resized to $1024 \times 704$ for training. After training 100 epochs on BDD and D$^2$-City, we fine-tune with iRAW dataset for another 20 epochs.

**Segmentation Network.** We use HRNetv2 [18] as our baseline model. The input image is resized to $1024 \times 512$. For rearranged $x_{reraw_g}$, its input size is $512 \times 256 \times 4$. We employ SGD optimizer with batch size 24 and the initial learning rate at 0.0025. Following the configuration in [18], we decay learning rate at every iteration after first 150k iterations. For the segmentation network, we only train our models on iRAW dataset.

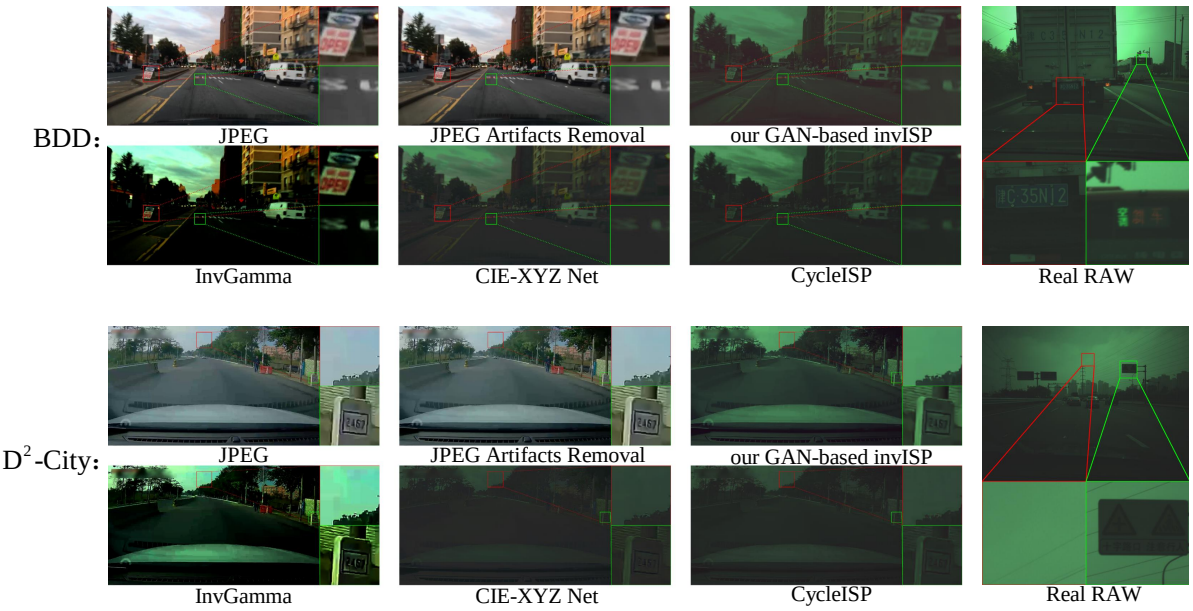


Figure 5. Visualization of RAW images generated by InvGamma [10], CIE-XYZ Net [2], CycleISP [21] and our GAN-based invISP on BDD and D$^2$-City. Real RAW images from iRAW dataset are presented for comparison.

| Domain | Pretrain | Finetuning | Test | car | person | bicycle | motorcycle | delay | mAP0.5 |
|---|---|---|---|---|---|---|---|---|---|
| **RGB** | BDD + D$^2$-City | iRAW$^{\text{RGB}}_{\text{train set}}$ | iRAW$^{\text{RGB}}_{\text{test set}}$ | **90.9** | **80.3** | 67.0 | **76.1** | 35.7ms | **78.6** |
| RAW | RAW$_{\text{invISP}}$ | iRAW$_{\text{train set}}$ | iRAW$_{\text{test set}}$ | 90.6 | 79.2 | **69.0** | 69.4 | **2.4ms** | 77.1 |

Table 2. Comparative studies of object detection in RGB domain and RAW domain.

## 4.3. Results

### 4.3.1 Learned invISP

Figure 5 presents the results of converting the RGB images in BDD & D$^2$-City dataset into RAW images using three methods: our proposed GAN-based invISP, invGamma [10], CIE-XYZ Net [2], and CycleISP [21]. To be fair, all compared networks were finetune on our iRAW dataset with 300 epochs. We observe that the color and details of the learned invISP generated images are closer to the RAW images from iRAW dataset. This suggests the efficiency of GAN-based invISP model that can clearly learn the cross-domain information and remove the JPEG artifact. To evaluate the quality of our invISP generated images, we use Fréchet inception distance (FID) and Mean Opinion Score (MOS). FID is commonly used in GAN to measure the distribution gap between the generated images and the source domain images. Table 1 shows that our method provides FID score nearly 10 points lower than that of CycleGAN. For MOS, we invited 5 testers to score our generated images, with a score of 5 if they cannot distinguish whether they are RAW images captured by iPhone XS Max or 0 if they are closer to the original JPEG images. Results suggest that our images generated by our proposed invISP had better subjective similarity to RAW images.

### 4.3.2 Detection Results

As shown in Table 2 and Figure 6, we see that the ISP delay is significantly reduced. Using a 30 FPS camera as an example, executing CV tasks in RAW domain can save 93% time compared to processing in the RGB domain with little loss in performance (-1.5 mAP). In some categories, such as cycle, the detection accuracy in RAW domain exceeds accuracy in the RGB domain.

### 4.3.3 Segmentation Results

Table 3 presents segmentation results in the RAW and RGB domains for the six categories: road, traffic light, traffic sign, person, rider and car. The results show that the segmentation task in the RAW domain reduces latency with little loss (-1.3 mIOU) in segmentation performance. Figure 6 shows example results of our segmentation task in iRAW. We can see that the segmentation results of iRAW is very close to that of the RGB segmentation.

## 5. Conclusion and Discussion

In this paper, we explore and verify the feasibility of executing high-level computer vision tasks in RAW domain. In order to verify the potential of RAW processing in real-

| Domain | Road | Sidewalk | Building | Wall | Fence | Pole | Traffic Light | Traffic Sign | Plant | Sky | Person | Rider | Car | Truck | Bus | Cycle | Delay | mIoU |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **RGB** | 97.29 | **60.57** | **91.34** | **40.88** | **84.57** | **69.57** | 54.80 | 86.55 | **94.57** | **99.02** | 59.56 | 64.43 | 92.28 | **84.02** | **82.24** | **46.91** | 955.7ms | **75.54** |
| **RAW** | **97.48** | 60.31 | 90.63 | 36.58 | 84.17 | 68.4 | **57.69** | **87.55** | 94.07 | 98.99 | **59.85** | **66.29** | **93.1** | 70.72 | 75.77 | 46.19 | **922.4ms** | 74.24 |

Table 3. Comparative studies of semantic segmentation in RGB domain and RAW domain.

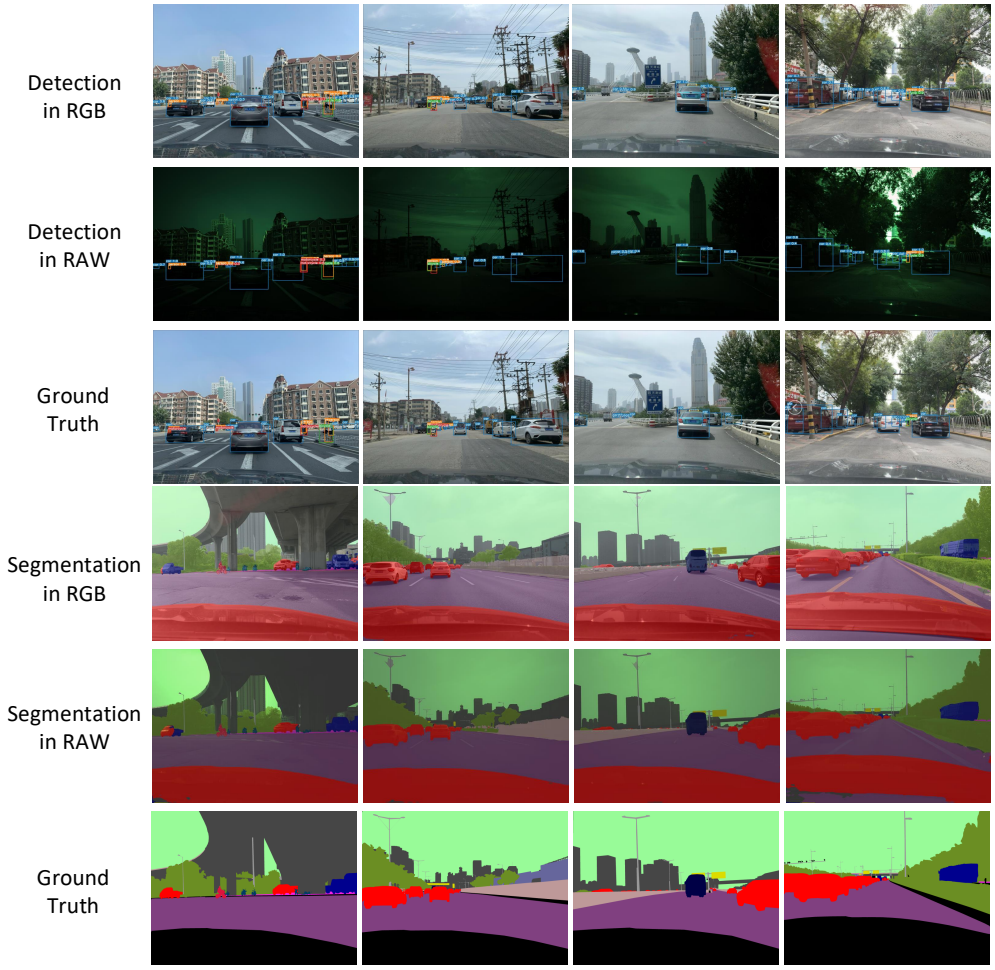


Figure 6. The visualization of object detection and semantic segmentation in both RGB and RAW domain.

world, we have collected and labeled the first RAW domain dataset iRAW. Due to the fact that most datasets are RGB format, we propose a GAN based learnt inverse ISP to extend the CV tasks in RAW domain. Experimental results show that CV tasks in RAW domain can effectively reduce the time delay and calculation, thus potentially saving the power with less computation.

The performance gap is very small, e.g., 1%, for the accuracy of CV tasks in RAW domain and that in RGB domain. we think this is because although we use GAN to narrow the gap between inverse RAW and real RAW, it can not be completely eliminated. This can be solved by training directly on real RAW, but due to the cost of acquisition, the number of iRAW datasets is still not enough to train directly on it. Therefore, we hope to collect a larger RAW dataset for direct training to better evaluate the gap between CV tasks in RAW and RGB domain.

At the same time, our network architecture completely uses the network architecture that performs tasks in the RGB domain. However, because the characteristics of RAW domain are different from that of RGB domain, whether there is a network architecture more suitable for RAW domain is also a question worthy of discussion. At the same time, for tasks that depend on the input resolution, where to add upsample in the network to restore the input resolution and reduce the amount of calculation is also our next research direction.

## References

[1] Abdelrahman Abdelhamed, Mahmoud Afifi, Radu Timofte, and Michael S Brown. Ntire 2020 challenge on real image denoising: Dataset, methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 496–497, 2020. 2

[2] Mahmoud Afifi, Abdelrahman Abdelhamed, Abdullah Abuolaim, Abhijith Punnappurath, and Michael S Brown. Cie xyz net: Unprocessing images for low-level computer vision tasks. *arXiv preprint arXiv:2006.12709*, 2020. 2, 3, 4, 6, 7

[3] David J Brady, Lu Fang, and Zhan Ma. Deep learning for camera data aquisition, control and image estimation. *Advances in Optics and Photonics*, Sept. 2020. 1

[4] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11036–11045, 2019. 2

[5] Zhengping Che, Guangyu Li, Tracy Li, Bo Jiang, Xuefeng Shi, Xinsheng Zhang, Ying Lu, Guobin Wu, Yan Liu, and Jieping Ye. $D^2$-city: A large-scale dashcam video dataset of diverse traffic scenarios. *arXiv preprint arXiv:1904.01975*, 2019. 6

[6] Jocher Glenn, Stoken Alex, Borovec Jirka, NanoCode012, ChristopherSTAN, Changyu Liu, Laughing, tkianai, Hogan Adam, lorenzomammana, yxNONG, AlexWang1900, Diaconu Laurentiu, Marc, wanghaoyang0106, ml5ah, Doug, Ingham Francisco, Frederik, Guilhen, Hatovix, Poznanski Jake, Fang Jiacong, Yu Lijun, changyu98, Wang Mingyu, Gupta Naman, Akhtar Osama, PetrDvoracek, and Rai Prashant. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements, Oct. 2020. 6

[7] Guoheng Huang, Junwen Zhu, Jiajian Li, Zhuowei Wang, Lianglun Cheng, Lizhi Liu, Haojiang Li, and Jian Zhou. Channel-attention u-net: Channel attention mechanism for semantic segmentation of esophagus and esophageal cancer. *IEEE Access*, 8:122798–122810, 2020. 4

[8] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016. 5

[9] Hakki Can Karaimer and Michael S Brown. A software platform for manipulating the camera imaging pipeline. In *European Conference on Computer Vision*, pages 429–444. Springer, 2016. 2

[10] Samu Koskinen, Dan Yang, and Joni-Kristian Kämäräinen. Reverse imaging pipeline for raw rgb image augmentation. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2896–2900. IEEE, 2019. 2, 6, 7

[11] Jianwei Li, Yongtao Wang, Haihua Xie, and Kai-Kuang Ma. Learning a single model with a wide range of quality factors for jpeg image artifacts removal. *IEEE Transactions on Image Processing*, 29:8842–8854, 2020. 4

[12] Zhihao Li and Zhan Ma. Robust white balance estimation using joint attention and angular loss optimization. In *Thirteenth International Conference on Machine Vision*, volume 11605, page 116051E. International Society for Optics and Photonics, 2021. 4

[13] Junichi Nakamura. *Image sensors and signal processing for digital still cameras*. CRC press, 2017. 1

[14] Rang MH Nguyen and Michael S Brown. Raw image reconstruction using a self-contained srgb-jpeg image with only 64 kb overhead. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1655–1663, 2016. 2

[15] S.J.D. Prince. *Computer Vision: Models Learning and Inference*. Cambridge University Press, 2012. 1

[16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 2, 5

[17] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. In *CVPR*, 2019. 2, 5

[18] Ke Sun, Yang Zhao, Borui Jiang, Tianheng Cheng, Bin Xiao, Dong Liu, Yadong Mu, Xinggang Wang, Wenyu Liu, and Jingdong Wang. High-resolution representations for labeling pixels and regions. *arXiv preprint arXiv:1904.04514*, 2019. 6

[19] Sifeng Xia, Kunchangtai Liang, Wenhan Yang, Ling-Yu Duan, and Jiaying Liu. An emerging coding paradigm vcm: A scalable coding approach beyond feature and signal. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020. 1

[20] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2636–2645, 2020. 6

[21] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2696–2705, 2020. 2, 3, 6, 7

[22] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019. 4

[23] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2, 3, 4