*electronics*

MDPI

*Article*

# Image Sampling Based on Dominant Color Component for Computer Vision

**Saisai Wang [1], Jiashuai Cui [2], Fan Li [1,2] and Liejun Wang [1,*]**

[1] School of Information Science and Engineering, Xinjiang University, Urumqi 830046, China; wangsaisai.xju@foxmail.com (S.W.); lifan@mail.xjtu.edu.cn (F.L.)
[2] School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an 710049, China; morganc@stu.xjtu.edu.cn
* Correspondence: wlj@xju.edu.cn

**Abstract:** Image sampling is a fundamental technique for image compression, which greatly improves the efficiency of image storage, transmission, and applications. However, existing sampling algorithms primarily consider human visual perception and discard irrelevant information based on subjective preferences. Unfortunately, these methods may not adequately meet the demands of computer vision tasks and can even lead to redundancy because of the different preferences between human and computer. To tackle this issue, this paper investigates the key features of computer vision. Based on our findings, we propose an image sampling method based on the dominant color component (ISDCC). In this method, we utilize a grayscale image to preserve the essential structural information for computer vision. Then, we construct a concise color feature map based on the dominant channel of pixels. This approach provides relevant color information for computer vision tasks. We conducted experimental evaluations using well-known benchmark datasets. The results demonstrate that ISDCC adapts effectively to computer vision requirements, significantly reducing the amount of data needed. Furthermore, our method has a minimal impact on the performance of mainstream computer vision algorithms across various tasks. Compared to other sampling approaches, our proposed method exhibits clear advantages by achieving superior results with less data usage.

**Keywords:** image sampling; computer vision; color feature



**Citation:** Wang, S.; Cui, J.; Li, F.; Wang, L. Image Sampling Based on Dominant Color Component for Computer Vision. *Electronics* **2023**, *12*, 3360. https://doi.org/10.3390/electronics12153360

Academic Editor: Silvia Liberata Ullo

Received: 7 July 2023
Revised: 31 July 2023
Accepted: 3 August 2023
Published: 6 August 2023



## 1. Introduction

Recent years have witnessed the rapid development of image applications associated with the use of artificial intelligence technology. Deep learning as well as neural networks have demonstrated their efficiency in terms of image-understanding tasks. The combination of machine learning and swarm intelligence approaches has been proved to be able to obtain outstanding results in different areas [1,2]. However, the expanding computation and memory costs caused by the larger resolution of the input image and the growing volume of the deep model creates an emerging challenge. Such intensive computation, transmission, and storage requirements can hardly be satisfied in lightweight devices, e.g., mobile phones and battery-operated remote IoT sensors. To this end, image sampling, by preprocessing the original high-resolution images to retain effective information for tasks, is of great significance because the input data volume can be directly decreased to reduce the resource costs and guarantee application efficiency. Figure 1 presents a diagram to explain the advantages of employing image sampling in computer vision tasks. Once the input data volume is reduced by image sampling, not only the transmission and storage costs but also the computation complexity of the computer vision algorithm can be decreased.

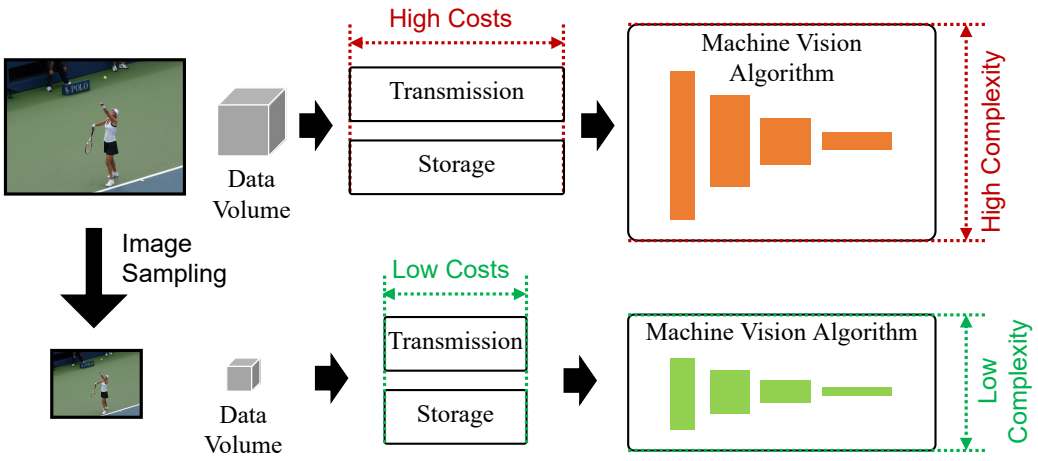



**Figure 1.** The advantages of image sampling.

In the past, researchers have investigated many image sampling methods targeting human eye perception. Uniform sampling following the Nyquist–Shannon sampling theorem [3] guarantees that the original images can be restored perfectly without any loss of information, leading to an impeccable vision effect, but it contains too much redundant information. Since human eyes tend to prefer the foreground and ignore the background, non-uniform sampling methods focus on sampling points in the foreground to achieve both greater compression and the maintenance of informative areas to ensure an acceptable vision effect. In the industry, considering the stronger sensitivity of human eyes for luminance, the Joint Photographic Experts Group (JPEG) [4] coding standard converts the image into a YUV space for sampling, saving the luminance component completely and downsampling the color components of chrominance and chroma to obtain a balance between reduction in data and good vision perception.

However, the increasing demand for computer vision supported by deep learning has highlighted the inefficiency of existing sampling methods designed for human eyes because of the difference between human and computer vision. Human visual perception may place more emphasis on semantic understanding and emotional recognition. The human brain has powerful cognitive capabilities, allowing for high-level semantic reasoning through the observation and analysis of visual information, such as objects, colors, and textures, in the environment. People also have emotional responses to certain visual scenes or objects, such as a preference for beautiful landscapes or disgust towards horrifying images. Computer vision, on the other hand, is based on powerful computing and analysis by computer which typically focuses solely on significant features of an image for subsequent analysis, such as color, shapes and structures. These features, characterized by a substantial amount of information and task-directedness, are more concise and amenable to analysis for computer vision tasks. In sum, human visual perception is more comprehensive and focused on semantic understanding and emotional recognition, while computer vision prioritizes fast and accurate image processing and analysis.

Most computer vision algorithms typically use RGB format images as input data to mimic human perception. However, these images are often redundant for computer vision tasks as they do not require fine-grained color analysis. In scenarios where image acquisition and analysis are separated, transmitting large volumes of RGB images presents challenges for image compression algorithms and communication bandwidth. In order to avoid severe distortion caused by limited bit-rate image compression, it is necessary to reduce the data volume at the source. We argue that the ideal image sampling method for computer vision tasks should be non-RGB, retaining minimal yet sufficient color information for analysis while significantly compressing data.

In this paper, we present an image sampling method based on a dominant color component (ISDCC) for computer vision. Building upon our previous work [5] published

at a conference, ISDCC aims to greatly reduce the data volume while retaining the effective information used in downstream vision tasks. The main contributions of our method are summarized as follows:

- A gray feature map with the original resolution is extracted to retain the main structural information of the image, allowing for object boundary distinction. By reducing the image data depth to 8 bits instead of 24 bits, the data volume is significantly reduced by two-thirds compared to the original RGB images. Additionally, the retained boundary feature ensures minimal performance loss in computer vision tasks.
- A succinct color feature map is constructed using the dominant color components of pixels to capture the distinguishing properties of objects. Specifically, the index number of color channels with the largest value at each pixel is used to construct the color feature. The spatial resolution of this feature map is downsampled to effectively represent the color feature with minimal data.
- The proposed method generates compact compressed data through simple non-deep-learning computations. This results in a low-complexity and efficient method that can adapt to various tasks, such as image classification and object detection, without requiring modifications. The experimental results obtained demonstrate the efficiency of the proposed method in terms of computation complexity, compression ability, and generalization.

The rest of this paper is organized as follows: In Section 2, we review algorithms relevant to this paper, including image sampling algorithms designed for human eyes and image preprocessing algorithms designed for computer vision. In Section 3, we explore the key color feature in computer vision, which is the motivation for this paper. Then, Section 4 introduces ISDCC, the proposed image sampling algorithm for computer vision. The experimental analysis is presented in Section 5, and the conclusions are presented in Section 6.

## 2. Related Work

### *2.1. Image Sampling for Human Eyes*

Nyquist pointed out that a band-limited signal can be recovered without any loss of information as long as the sampling frequency is more than twice the highest frequency. However, when specific to the field of image sampling, the sampling interval of the whole image is always dense since it is determined by the most drastic region, and the above sampling theorem must be satisfied in three channels at the same time. So uniform sampling must result in serious redundancy and has almost no value for reducing the amount of data.

Non-uniform sampling is proposed, which focuses on assigning more sampling points to the foreground regions of interest. Adaptive mesh sampling was proposed in [6], modeling the distance between pixels as the elastic system to sample points. Elder [7] introduced the farthest point strategy into the field of image sampling and derived the best sampling points through the error function. Farthest point sampling is also one of the most popular methods in the field of point clouds. The distortion of the image luminance is taken into account for sampling in [8]. In addition, the authors of [9–13] propose methods for allocating sampling points from different views. Wavelets were introduced into the field of image sampling [14–17] following the idea of signal processing. From the perspective of a manifold, the authors of [18–21] sought to model and solve the problem of image sampling in a mathematical style. From a statistical point of view, an image can be regarded as a Markov random field, which was applied in [22]. Much research has focused on the sparsity and low sampling rate [23–25]. With the development of deep learning, related methods have been developed to obtain good performance [26] but are associated with high complexity.

In industrial applications, the JPEG coding standard first converts the image into the YUV color space for sampling. The luminance component Y is retained completely, and the color components U and V can be downsampled with different factors taking into account

the need to balance reducing the amount of data and the feel of human eyes. The idea of keeping the luminance and the downsampling color information is valuable for reference.

The above traditional sampling methods are all designed for human eyes and are unable to retain the information required by machine analysis, so the amount of data is inevitably redundant.

### *2.2. Image PreProcessing for Computer Vision*

Focusing on learning-based computer vision algorithms, existing studies have confirmed that image preprocessing modules can improve the performance of machine analysis tasks, with image sampling representing one of the methods. In the image coding field, previous studies [27,28] have shown the necessity of coding aimed at computer vision.

Specifically for preprocessing methods of collected images, image enhancement approaches, such as super-resolution, pretransformation, and denoising have been explored. A super-resolution network was adopted in [29] to preprocess low-resolution images and was trained jointly with the detection network, effectively improving the performance of the detection task. Gandal [30] improved the quality of images generated by GAN networks by introducing a texture loss function, which ensured the following visual tasks worked well. In [31], a dual-directed capsule network combining high-resolution image anchor loss and reconstruction loss was used to reconstruct very-low-resolution images to enable face recognition. Suzuki [32] used a deep encoder-decoder to pretransform and compress images, keeping the accuracy for recognition while reducing the image bit rates. The authors of [33] used dynamic convolution for filtering to enhance images and to improve the performance for classification. A dual-channel model and denoising algorithm were used in [34] to improve the quality of noisy images, thereby improving recognition accuracy. RSRGAN was proposed in [35], which utilizes super-resolution to enlarge small objects in infrared images and can obtain better detection performance. The authors of [36] noted the positive effect of image enhancement for the detection of COD in eye fundus images and proposed practical methods. In [37], a novel illumination normalization method was proposed to remove illumination boundaries and to improve the image quality under dark conditions, improving face detection. A multi-scale fusion of various prior features was used in [38] to enhance underwater images and to facilitate subsequent visual tasks for the capture of underwater scenes. A preprocessing method was proposed in [39] to suppress background interference for infrared pedestrian object detection.

Though offering better computer vision performance, the above-mentioned methods do not fully take computation costs into consideration. Learning-based networks are generally adopted for such preprocessing modules. Therefore, they will inevitably introduce additional computation and resource costs to the original computer vision algorithm, and the complexity of these network models is normally significant. Moreover, the spatial resolution of the input image is not decreased, whereas the image will have an even larger resolution when super-resolution-based approaches are used. With reference to the previous discussion that the spatial resolution is proportionate to the costs for image processing, these methods will have limited scope for application in lightweight scenarios.

To this end, learning-based image-resizing methods have also been examined to achieve image sampling. The authors of [40] designed an image resizer network with the target of achieving optimal visual task performance, which achieved excellent recognition performance on ImageNet [41] and AVA datasets through joint training with visual algorithms. ThumbNet and related training strategies were proposed in [42], which can reduce the image size before performing visual tasks, and which can even ensure the accuracy of downstream tasks, with 16-fold smaller images. Chen et al proposed decomposing the input image into two low-resolution sub-images carrying low-frequency and high-frequency information in [43], thereby accelerating the processing of visual tasks.

In these studies, the spatial resolution of the input image can be reduced, and the loss of machine analysis performance is limited as far as possible by training. However, their generalization ability is limited. In general, the application of a learning-based image

resizing model must match the backbone network that participates in the training process. The above approaches only demonstrate strong performance on image classification tasks. Moreover, they also introduce much additional complexity.

In addition to image sampling, immersive data sampling, including point clouds sampling, is also of great significance for the development of computer vision tasks. There are a growing number of tasks that work directly on point clouds. As the size of the point cloud grows, so do the computational demands of these tasks. A possible solution is to sample the point cloud first. A widely used method is farthest point sampling (FPS) [44,45]. FPS starts from a point in the set and iteratively selects the farthest point from the points already selected. Ref. [46] introduced a novel differentiable relaxation for point cloud sampling that approximated the sampled points as a mixture of points in the primary input cloud. Ref. [47] proposed a resolution-free point clouds sampling network to directly sample the original point cloud to different resolutions, which was performed by optimizing the non-learning-based initial sampled points to better positions. Furthermore, data distillation was introduced to assist the training process by considering the differences between the task network outputs from the original point clouds and the sampled points. Ref. [48] proposed an objective point cloud quality index with structure guided resampling to automatically evaluate the perceptual visual quality of 3D dense point clouds. Ref. [48] exploited the unique normal vectors of point clouds to execute regional preprocessing, which involved key point resampling and local region construction.

For applications in other fields, such as sampling for training physics-informed neural networks, ref. [49] proposed a novel sampling scheme, called dynamic mesh-based importance sampling, to speed up the convergence without significantly increasing the computational cost. To reduce the computational cost, a novel sampling weight estimation method was introduced, called dynamic mesh-based weight estimation, which constructs a dynamic triangular mesh to estimate the weight of each data point efficiently.

## 3. Motivation

To reduce the amount of data while preserving the main structural information of the whole image, a suitable sampling method for computer vision is necessary. Gray image sampling is an ideal option as it contains edge information that distinguishes the foreground and background, resulting in clear object shapes. Moreover, gray images retain the essential structural information of the original images while reducing the data volume by two-thirds compared to RGB images. Similarly, sampling in the JPEG standard preserves the luminance component as the main body of the sampled image data.

However, images that only contain structural information are not sufficient for fulfilling the requirements of computer vision tasks. Gray images can only distinguish gray levels and lack color information that is critical for computer vision compared to the original RGB images.

On the one hand, color information plays a vital role in identifying objects as it is a key attribute for distinguishing different classes of objects. For example, the COCO [50] dataset contains photographs of broccoli and cauliflower, as shown in Figure 2, both of which are unidentifiable in shape and cannot be distinguished through gray images, even by humans. The critical feature that differentiates the two classes is their color—broccoli is green while cauliflower is yellow.

On the other hand, color information can help objects stand out from the background. In complex scenes, objects can easily be hidden in the background without color information as a reference. To illustrate this point, we compare the saliency maps of gray images and of RGB images in Figure 3 using an image from the COCO dataset and a classical saliency algorithm [51]. The traffic signs in the gray image are close to the background street in gray level and are difficult to identify. If the color information is considered in an RGB image, the red attributes of the signs can make them conspicuous in the background. It can be seen more clearly from the saliency maps that the color information produces high responses

at the locations of the traffic signs to make them stand out, while the gray image cannot provide this information.

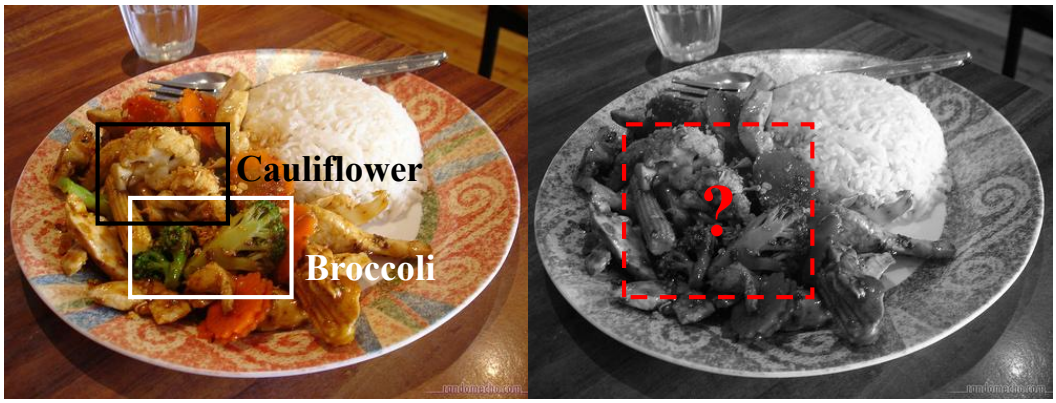



**Figure 2.** Color information is a key property of objects.


**Figure 3.** Color information can help objects stand out from the background. We compare the saliency maps of gray images and of RGB images. The red squares denote the saliency location in RGB. While the salient region in the gray image are close to the background in gray level and are difficult to identify.

In computer vision, color information is crucial in accurately identifying certain objects. It is important to determine the most critical color information and to represent it with the smallest amount of data in a sampling algorithm. Therefore, we assume that the ordinal relationship of the color channels is more vital, rather than the specific value of the color channels for RGB images. This relationship determines which channel among R, G, and B has the largest value and is the dominant component of color. Thus, it is critical to maintain this information in color feature analysis.

To validate our assumption, we conducted experiments on a well-trained ResNet50 [52] classification network. We changed the input format for test images in two ways: (1) by uniformly increasing the luminance value of every pixel on the R, G, and B channels by 20, without altering the ordinal relationship of the channels, and (2) by changing the ordinal

relationship of the channels. Here, we change the channel order of the input images by swapping the brightness values of the RGB channels. The process of changing the channel order begins by considering each pixel in the input image. For every pixel, the three-channel values are evaluated, and the brightness values of the maximum and minimum channels are swapped. This step ensures that the relative ordering of the channels is effectively altered. The pixel-wise adjustment is performed by a strategy for this purpose. Table 1 clearly shows that change in the ordinal relationship had a more significant impact on the recognition ability of the network. Such results support our assumptions and highlight the importance of ordinal information in color feature analysis.

**Table 1.** What is the most critical factor in a color feature?

| | Precision/% | Decrease Compared to RGB/% |
|---|---|---|
| Numerical Change | 58.58 | −14.18 |
| Ordinal Change | 51.81 | −20.94 |

According to the preceding analysis, an ideal image sampling algorithm must preserve the essential structural information while also efficiently supplementing the color information. Moreover, the inter-channel ordinal relationship plays a crucial role in extracting the color features. In light of these considerations, we propose ISDCC, an image sampling method that integrates gray information with the dominant color channel index information.

## 4. Method

### *4.1. Overview*

We propose ISDCC, an image sampling method based on the dominant color component. The framework, as shown in Figure 4, mainly contains two parts: (1) gray imaging, designed to represent the structural and fundamental information, where the depth is 8 bits/pixel rather than 24 to reduce the transmission cost; (2) color feature generation, in which the dominant color features are obtained by our proposed concise representation strategy with a volume of 2 bits/pixel, while we also adopt downsampling to further compress the image information.

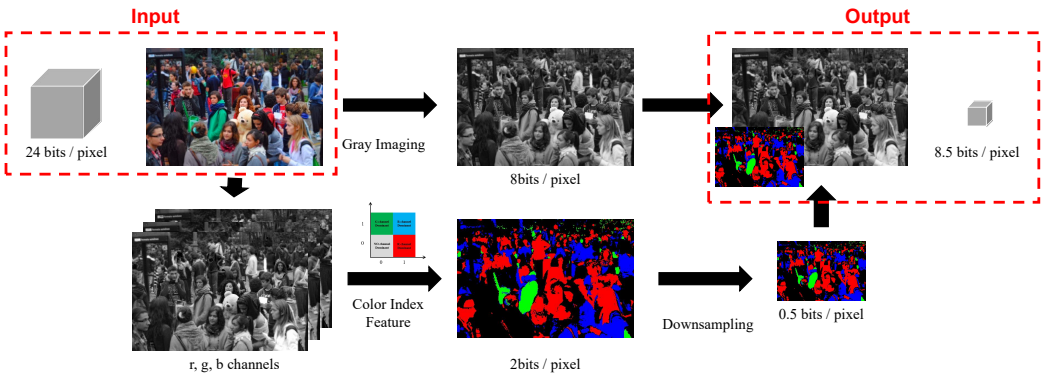



**Figure 4.** ISDCC, the proposed image sampling method combining gray image and color index features. The original image will be sampled as a gray image to decrease the volume of data and a color feature map to save key information for computer vision. The detailed method is as shown: Firstly, gray imaging is conducted to save the main structural information of the original images; while the index number of channels at pixels with the largest value can be used to construct a color feature; when the RGB channels have little difference and the pixel is nearly gray as a whole, the fourth code is adopted but not the index number of channels to represent as there is no dominant color component; finally, the color feature map will be downsampled to further reduce the data volume.

The ISDCC takes an RGB image as input and can be directly applied without deep learning. Since the majority of current image capture devices generate RGB data, working in the RGB domain ensures compatibility with commonly used imaging equipment. What is more, RGB is widely used as the color space in most computer vision tasks. Transforming RGB images into other domains would introduce additional computational overhead, which is not aligned with our objective. By utilizing the RGB domain, our approach maintains consistency with other image-processing tasks. Moreover, the RGB domain reflects the dominant color components of pixels and is consistent with human visual perception. By leveraging the RGB color domain, our approach aligns with the natural visual perception of the human visual system. Therefore, we choose to use RGB as the color representation domain for the sampling process.

#### 4.1.1. Gray Imaging

The use of gray images is common in computer vision applications due to their ability to preserve the key elements of an image such as the edges, foreground, and background. Grayscale images can be easily derived from the original image through simple grayscale operations, reducing the amount of data per pixel from 24 bits to 8 bits. Furthermore, the lack of color information caused by the grayscale operation can be compensated for by incorporating a color feature provided by another branch. This approach guarantees high performance of the computer vision tasks.

#### 4.1.2. Color Feature

After conducting our exploration, we discovered that ordinal information plays a more significant role in color features. In light of this, and drawing upon the operation used to downsample color information in JPEG sampling, we propose a novel approach to constructing a color feature map. To achieve this, we prioritize the dominant components of the RGB channels to represent the ordinal relationships of colors. This approach allows us to construct a more comprehensive and informative representation of color features.

Representation of Color Feature. The RGB color space is a common representation used for digital image processing. Each pixel in an image can be divided into three color channels: red, green, and blue. Among the three channels, the one with the largest numerical value in each pixel position represents the dominant color component. This implies that the index number of the channel with the largest value can be utilized to construct a color feature representing that pixel. As there are three possible index numbers, namely 1, 2, 3, four combinations of bit representation are adequate to cover all possible cases, as illustrated in Figure 5a. The construct of a color feature in this approach is analogous to vector quantization, but has a clearer physical interpretation, as each value can be directly associated with the proximity of the pixel color to red, green, or blue.

Use of Feature Space. The 2 bits can represent four cases, and the range of index numbers covers only three cases, so it can make full use of the feature space to encode the fourth case. The proposed representation of color features distinguishes different colors by their dominant channel but ignores the fact it is inappropriate to classify some ambiguous cases directly to a specific color feature. For a pixel, when the RGB channels have little difference, the pixel is nearly gray as a whole, and there is no dominant component. It is not reasonable to code a gray pixel as a 'Red' feature just because the value of its R channel is only one or two higher than the values of channels G and B. Therefore, considering such cases separately, for pixels whose range of three channels is less than a specific threshold, the color feature will be encoded as the fourth case to represent a case of no dominant color component of the pixel, as shown in Figure 5b.

Compression of Color Feature. The color information in images does not require precise boundaries, like structural information. Instead, it primarily needs to represent the dominant color of areas. This representation can aid in object recognition and segmentation by helping objects stand out in machine analysis. To process less important color information, spatially sampling color maps in JPEG compression serves as a valuable operation

for reference. Consequently, downsampling the color feature map further reduces the data volume without undermining the efficacy of visual tasks.

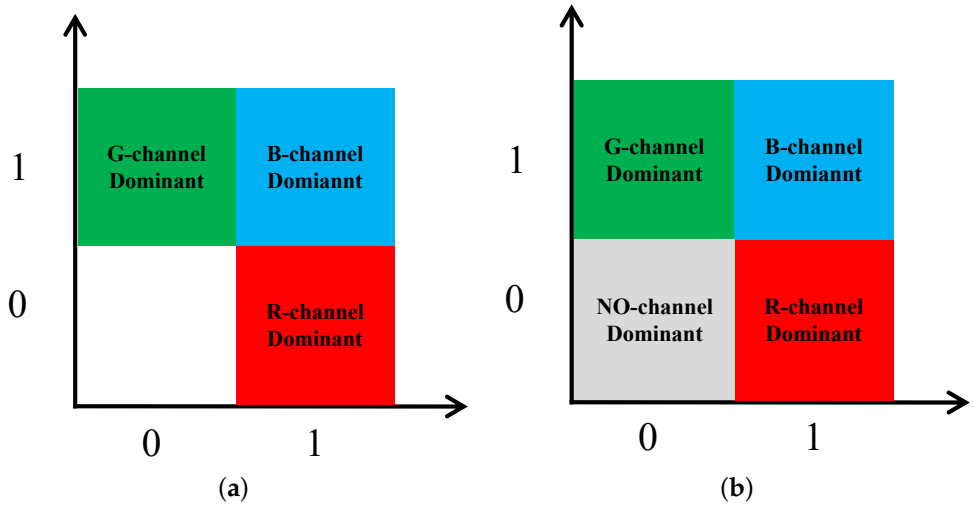



**Figure 5.** Representation of color feature with dominant channel. (**a**) Representation of color feature. (**b**) Full use of feature space.

### *4.2. Sampling Result Visualization*

The proposed sampling method aims to obtain structural information in the form of gray images and a concise color feature map, which can effectively supplement color information. To visualize the effectiveness of the proposed method, we present the sampling output in Figure 6. Notably, the gray image shows the red traffic sign that is barely distinguishable from the background, indicating a defect in the structural information. However, the color feature map highlights the red traffic sign, filling the gaps in the structural information. These results demonstrate the efficacy of our proposed sampling method and its ability to enhance both structural and color information. Similarly, crowds are chaotic and hard to distinguish in gray images but can be made distinct with the help of the sampled color feature maps. There is a good sampling effect as expected.

**Figure 6.** Visualization of sampling results.

### *4.3. Cost Assessment*

We evaluate the resource cost of the proposed method compared to directly using raw RGB images and only using gray images, with results shown in Table 2.

**Table 2.** Source assessment of the proposed sampling method.

| Input | Bit/Pixel | FLOPs/Pixel | Time/ms |
|---|---|---|---|
| RGB | 24 | 0 | 5.24 |
| Gray | 8 | 5 | 5.60 |
| ISDCC | 8.5 | 10 | 6.55 |

The proposed method exhibits high efficiency in terms of space cost as it is capable of reducing nearly two-thirds of the data compared to the initial RGB form. This is achieved through the construction of efficient color features. Furthermore, our method results in a marginal increase in the amount of data when compared to grayscale images. However, it can significantly enhance the performance of computer vision when its sampling results are utilized as input. This will be elaborated upon in subsequent sections.

Regarding the time cost, we will discuss the amount of computation in theory and the actual processing time. When utilizing raw RGB images, no additional processing is required. However, in the case of grayscale processing, averaging is performed on each pixel, following the grayscale formula as

$$Gray = rR + gG + bB \tag{1}$$

where $R$, $G$, and $B$ are the channel values of a pixel and $r$, $g$, $b$ are the corresponding coefficients with values of $r = 0.299$, $g = 0.587$, and $b = 0.114$. The computational complexity of the proposed method can be evaluated in terms of floating-point operations. Specifically, each pixel in the color feature map undergoes a total of three multiplications and two additions, which are needed to obtain the corresponding feature value. Moreover, during the subsequent process of feature map sampling and averaging, an additional five floating-point operations are required. These operations include three comparison operations and one subtraction operation to calculate the range of values among the $R$, $G$, and $B$ channels, as well as one comparison operation to check whether the range exceeds a predefined threshold. Notably, no calculation is needed during the downsampling process, as the corner value of each cell is directly adopted as the final result. Overall, by carefully considering the computational complexity, the proposed method provides an efficient and effective approach for analyzing color image data.

The aforementioned comparison of computational complexity is predicated on ideal conditions and does not account for practical factors, such as data access efficiency and operator optimization. To alleviate this drawback, we conducted a timed test to measure the actual processing time of 5000 images scaled to a size of 224 × 224 from the COCO validation set. The test was performed on an Intel i5-8400 CPU and the results demonstrate that our proposed sampling method incurred a negligible time cost.

In light of the above, it is evident that our proposed sampling method can significantly reduce the amount of data required for computer vision tasks without imposing a substantial time cost. As such, it is a promising technique for addressing computer vision tasks at low cost.

## 5. Experiments

In order to evaluate the efficacy of the proposed method for enhancing computer vision tasks through retaining critical information, a robust set of experiments was conducted on multiple datasets and models. The experimental results demonstrate that the proposed sampling method incurs only marginal performance loss, while considerably reducing the amount of data. Moreover, the color features embedded in this method serve to complement the existing color information with minimal overhead, thereby further enhancing its effectiveness. When compared to using gray images as input to computer vision algorithms, the proposed approach achieves notably superior performance in enhancing computer vision tasks. All of the experiments were carried out on the Pytorch1.7 and CUDA10.2-based deep learning environment, running on NVIDIA GTX 2080TI hardware.

### *5.1. Object Detection*

In the following experiments regarding object detection, mAP is the authoritative metric used. In the field of object detection, precision is the proportion of correct results in all the bounding boxes predicted under the premise of certain confidence. The recall is the proportion of the bounding boxes predicted by the algorithm in the ground truth of all images. A curve drawn with precision as the vertical axis and recall as the horizontal axis, AP, i.e., the average precision, is mathematically the area under the curve to balance the accuracy and omission. For a multi-class task, mAP is the mean value of APs of every class. In practice, the COCO standard is used to calculate the mAP metric where mAP@0.5 means the confidence threshold is 0.5 and mAP@0.5–0.95 is the mean value of the mAPs under each threshold from 0.5 to 0.95.

#### 5.1.1. mAP on Different Datasets

In this study, we evaluate the performance of the YOLOv5-s model [53–56] using different input data formats, including raw RGB images, grayscale images, and sample images, which are processed by the proposed sampling method. We compare their results in terms of various detection metrics on the Pascal VOC and COCO datasets, with the aim of assessing the effectiveness of our proposed method.

During the training process, we carefully control the experimental settings to ensure fair comparisons, with the exception of the input data format and the HSV augmented settings. Notably, we did not perform any color augmentation on the sampled images since the color features of the sampled images can change greatly. All the other training settings are kept consistent to make accurate performance comparisons.

The experimental results presented in Table 3 demonstrate that our proposed sampling method is successful in achieving the expected results on the Pascal VOC and COCO datasets in the field of object detection. Compared to gray inputs, our method provides rich color information and significantly improves object detection performance. Specifically, on the larger and more challenging COCO dataset, our proposed method yields greater improvements, indicating its robustness and adaptability to complex scenarios. Overall, these findings highlight the practical value of our proposed method for object detection tasks.

**Table 3.** YOLOv5-s on different datasets.

| Dataset | Input | mAP@0.5/% | mAP@0.5–0.95/% |
|---|---|---|---|
| VOC | RGB | 86.6 | 62.6 |
| | Gray | 85.6 | 60.8 |
| | ISDCC | 85.8 | 61.4 |
| COCO | RGB | 55.4 | 36.7 |
| | Gray | 53.6 | 35.0 |
| | ISDCC | 55.4 | 35.8 |

#### 5.1.2. mAP under Different Detection Algorithms

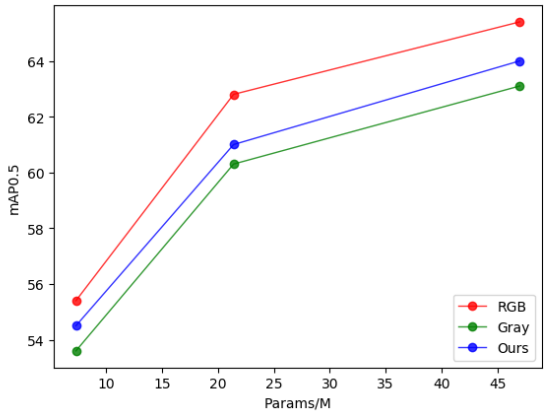
In this study, we aimed to evaluate the impact of our proposed sampling method on the performance of widely used object detection algorithms, including YOLOv5, Faster R-CNN, and CenterNet, on the Pascal VOC dataset. We compared the results of our method against three different inputs. Table 4 summarizes the obtained results, indicating that our sampling approach consistently improves the performance of the mainstream one-stage, two-stage, and anchor-free object detection algorithms. Our findings demonstrate the versatility of our proposed approach, showing that it can be effectively applied to different detection paradigms.

**Table 4.** Different algorithms on Pascal VOC (mAP@0.5/%).

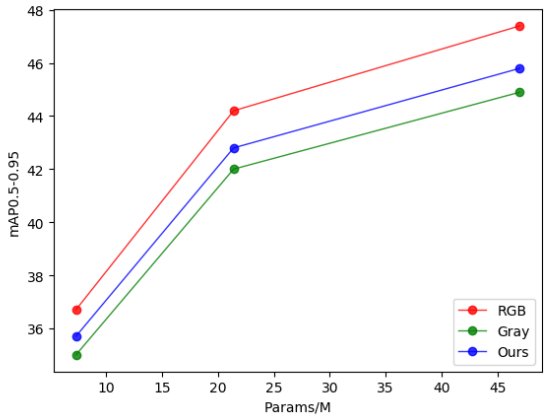
| | **YOLOv5-m** | **Faster R-CNN** [57] | **CenterNet** [58] |
|---|---|---|---|
| RGB | 90.1 | 70.1 | 74.9 |
| Gray | 89.2 | 65.4 | 72.4 |
| ISDCC | 89.5 | 66.0 | 72.8 |
| Increase | +0.3 | +0.6 | +0.4 |

#### 5.1.3. mAP under Different Scales of Models

In our study, we compare the detection performance of various YOLOv5 series networks with different scales on the COCO dataset using three different input forms. Specifically, we explore the effectiveness of the proposed sampling method in improving the detection results of networks with different sizes. Our experimental results, as illustrated in Figures 7 and 8, indicate that the sampling method shows consistent improvement across all network scales, including large, medium, and small models. This suggests that the proposed sampling method can be generalized and applied to a wide range of detection networks.




(**a**)




(**b**)

**Figure 7.** YOLOv5 models of different scale on COCO. (**a**) mAP@0.5/%. (**b**) mAP@0.5–0.95/%.

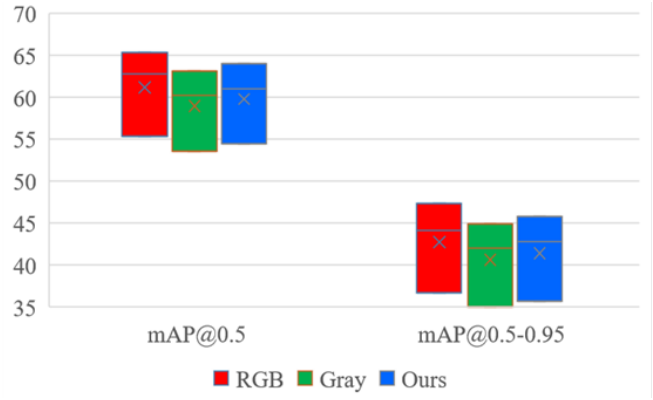



**Figure 8.** Box and whiskers diagram of YOLOv5 model performance on COCO.

### *5.2. Image Classification*

In the subsequent experiments, the criterion of top-1 accuracy was utilized as a metric. Specifically, the correctness of a prediction is evaluated based on whether the class label with the highest confidence, as predicted by the algorithm, matches the true label of the corresponding image. Therefore, the top-1 accuracy can be defined as the ratio of the number of correctly classified instances to the total number of instances in the dataset. This metric serves as a fundamental indicator of the classification capability of the algorithm under consideration.

#### 5.2.1. Accuracy on Different Datasets

In the present study, we conducted a comparative analysis of the accuracy achieved by the ResNet50 backbone network, utilizing three different types of inputs. The results, as highlighted in Table 5, indicate that the proposed method outperforms the gray image approach, and exhibits consistent efficacy across both authoritative datasets evaluated in this study.

**Table 5.** Classification precision of ResNet50 on different datasets (%).

| | **Cifar100** | **ImageNet** |
|---|---|---|
| RGB | 78.65 | 72.76 |
| Gray | 71.44 | 70.95 |
| ISDCC | 73.92 | 71.34 |
| Increase | +2.48 | +0.39 |

#### 5.2.2. Accuracy under Different Scales of Models

In this study, we evaluate the performance of ResNet networks of varying scales on the Cifar100 dataset with respect to image classification. Specifically, the comparison is conducted on three different input forms. Our findings, as illustrated in Figures 9 and 10, indicate that our proposed approach exhibits consistent improvement across models of different sizes when compared to gray image classification. As such, our results suggest that the proposed method is effective and widely applicable for enhancing the performance of classification networks of varying scales.

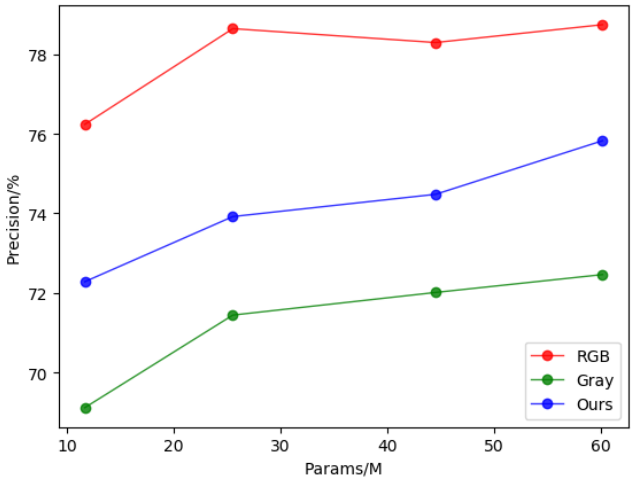



**Figure 9.** Classification precision of ResNet models in different sizes on Cifar100.

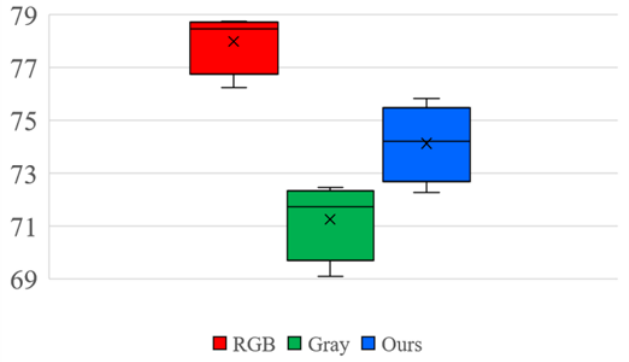



**Figure 10.** Box and whiskers diagram of classification precision of ResNet models.

#### 5.2.3. Accuracy under Different Classification Algorithms

We also investigate the effects of three input forms on mainstream backbone networks using the Cifar100 dataset. Specifically, we compare the performance of these networks using our proposed sampling method.

As depicted in Table 6, our findings suggest that our proposed approach is applicable to image classification networks utilizing varying paradigms, and it consistently exhibits improvements over mainstream networks. These results indicate that our proposed sampling method can significantly enhance the performance of image classification networks of different paradigms.

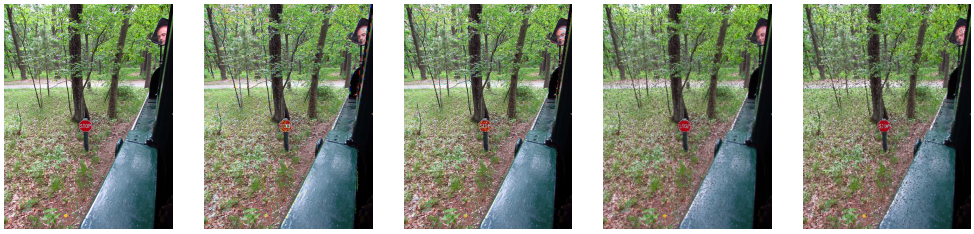
**Table 6.** Classification precision of different networks on Cifar100 (%).

| | **MobileNet-v2 [59]** | **Vgg16 [60]** | **DenseNet121 [61]** |
|---|---|---|---|
| RGB | 68.61 | 72.38 | 79.24 |
| Gray | 60.45 | 63.92 | 72.04 |
| ISDCC | 64.76 | 67.81 | 75.62 |
| Increase | +4.31 | +3.89 | +3.58 |

### *5.3. Comparison with Other Image Sampling Methods*

In order to showcase the advantages of our proposed image sampling method, we conduct a thorough comparison with other popular techniques. Firstly, we discuss the computational complexity of these methods while generating output of the same or ap-

proximate bits. Next, we evaluate the performance of downstream visual tasks which use input images sampled by various methods under fair testing conditions. We present the comparison techniques alongside the images they have generated in Figure 11. Through this comparison, we strive to reinforce the efficacy and superiority of our proposed method in terms of its performance and computational efficiency.



(**a**) (**b**) (**c**) (**d**) (**e**)

**Figure 11.** Visualization of sampled images by comparison methods. (**a**) Original. (**b**) Y:U:V = 4:1:1. (**c**) Y:U:V = 4:2:0. (**d**) Farthest. (**e**) Random.

The sampling method adopted in the JPEG standard is the primary comparison target, which converts RGB images to YUV format and samples the U and V components spatially, including Y:U:V = 4:1:1 and Y:U:V = 4:2:0, where different proportions distinguish whether the U and V components are interlaced in sampled lines but are not based on means only sampling the U component and discarding the V component. Farthest point sampling is popular in the sampling of image and point cloud data, involving sampling a point that is at the farthest distance from the set of currently sampled points as every new point is used to ensure that all the sampled points can uniformly cover the original image in the end. It achieves excellent performance at a cost of extremely high computational complexity for calculating the distance and deciding which point to sample at every iteration. Furthermore, we also compare random sampling, which involves randomly sampling points from all pixels of an image. It should be noted that the two sampling methods used in JPEG will sample a pixel to be the equivalent of 12 bits, while the other two comparison methods are set to output 8.5 bits per pixel, the same as ISDCC. These methods and ours are adopted to sample images in public datasets to train models of YOLOv5-s.

It can be seen from Table 7 that the proposed method has the advantage of conciseness because of the lower computation. It should be noted that the floating point operations per pixel of the sampling method used in JPEG and ours can be calculated accurately, while the other two methods have many loops per pixel that are of huge complexity, beyond the magnitude of ours, especially for farthest point sampling. With respect to the performance of object detection with the input of sampled images using these methods, the sampling methods in JPEG only surpass ours very slightly but at the cost of nearly 50% greater data volume. So the proposed method has the large advantage of achieving a comparable performance with a significantly lower data volume. Compared to farthest point sampling and random sampling which are restored by the nearest neighbor pixel and set to the same data volume, our method has obvious advantages in terms of performance. The experiments undertaken on two authoritative public datasets support our conclusion.

To summarize, the proposed method is one of the most competitive image sampling methods with excellent performance and lower cost. It also demonstrates that the idea of sampling aimed at computer vision is correct.

**Table 7.** YOLOv5-s on different datasets.

| Dataset | Method | FLOPs/pixel | Bit/Pixel | mAP@0.5/% | mAP@0.5–0.95/% |
|---|---|---|---|---|---|
| VOC | Y:U:V = 4:1:1 | 15 | 12 | 85.8 | 61.5 |
| | Y:U:V = 4:2:0 | 15 | 12 | 85.9 | 61.5 |
| | Farthest | $O(n^2)$ | 8.5 | 85.9 | 61.5 |
| | Random | O(n) | 8.5 | 85.9 | 61.5 |
| | ISDCC | 10 | 8.5 | 85.8 | 61.4 |
| COCO | Y:U:V = 4:1:1 | 15 | 12 | 55.4 | 35.8 |
| | Y:U:V = 4:2:0 | 15 | 12 | 55.5 | 35.8 |
| | Farthest | $O(n^2)$ | 8.5 | 51.9 | 33.0 |
| | Random | O(n) | 8.5 | 49.6 | 31.3 |
| | ISDCC | 10 | 8.5 | 55.4 | 35.8 |

### *5.4. Ablation Experiments*

#### 5.4.1. Performance with Different Bits of Color Feature

In the proposed method, each pixel is encoded as a low-bits color feature map to obtain a compact representation of color information. To satisfy the encoding requirement for different dominant color channels, only 2 bits are needed. This assumption was investigated using the COCO dataset and YOLOv5 network; the experimental results revealed that using 2 bits was the most appropriate approach for constructing the color feature map. The experiments also demonstrated that using more or fewer bits did not result in any improvement. Thus, it is suggested that a 2-bit representation should be employed for encoding the color feature map in order to achieve the best results.

The experimental settings are described as follows: It involves using the gray image without color information for 0 bits. In the experiment of 1 bit, the color feature is encoded as 0 if the values in the three channels are approximate and as 1 where there is one dominant channel. The experiment of 2 bits is the same as for the method adopted currently. In the experiment of 3 bits, similar to vector quantization, every channel corresponds to encoding one bit as 1 if the channel is larger than 127 and as 0 if not. As shown in Figure 12, the performance of 2 bits is better than for the other groups. We believe that more bits mean more fine-grained color features and stronger representation of color information. The advantage of 2 bits over 3 bits can be explained on the basis that there is a clearer physical meaning of red, green, blue, and no color component is dominant. Therefore, it is most appropriate to construct a color feature map with 2 bits.

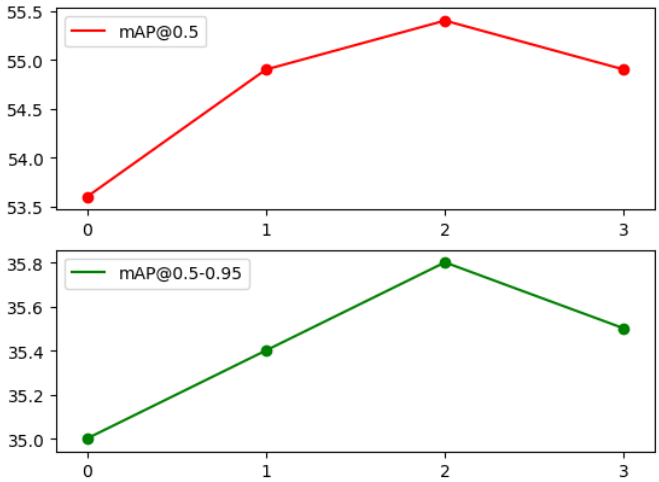



**Figure 12.** Exploring the most approximate bits of the color feature (bit-mAP/%).

5.4.2. Performance with Different Thresholds

In constructing a color feature, we propose a method that judges the three-channel range value of each pixel. If this value is lower than a certain threshold, we encode the color feature of the pixel as the fourth case, representing pixels that are close to gray without a dominant color. On the other hand, if the range value is higher than the set threshold, the feature is encoded as the max channel index value. We validate the necessity of setting a threshold to encode the non-dominant color case individually in the COCO dataset and YOLOv5 network.

As presented in Figure 13, we observed that the performance of the networks was significantly lower when using no-threshold-sample images as input. This result strongly advocates for the necessity of encoding the case of no dominant color individually. The purpose of setting a range threshold for the difference between the three-color channels of a pixel is to classify it as gray if the difference is below a certain range. However, if the threshold is too large, it would misclassify pixels that have significant differences and the dominant color channels as gray, which is obviously unreasonable. Therefore, through multiple settings comparison, we identified and selected the optimal threshold of 32 as the criterion to determine whether there is a dominant color in the pixel, which yields the best effect.

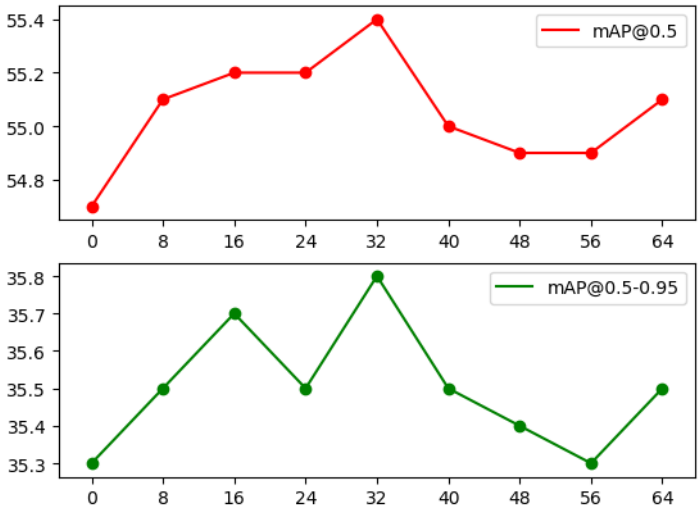



**Figure 13.** Exploring range threshold (threshold-mAP/%).

5.4.3. Performance with Different Downsampling Factors

The proposed method performs downsampling on the color feature map to compress the amount of data under the premise of ensuring the effect when applied to computer vision tasks. Based on the COCO dataset and the YOLOv5 object detection network, we verify the necessity of downsampling the color feature map and explore the optimal downsampling factor.

As shown in Figure 14, appropriate downsampling by $1/21 = 1/2$ can actually improve the performance of computer vision tasks. We argue it is because the color feature is not as fine as edge information and may generate errors at the boundary of regions. Downsampling can eliminate such tiny distortions so that the networks can focus on the color information. In addition, although other factors associated with downsampling can further reduce the amount of data, the marginal utility is small, and the benefits are not sufficient to make up for the performance drop. Therefore, downsampling is necessary and we choose $1/2$ as the optimal factor.

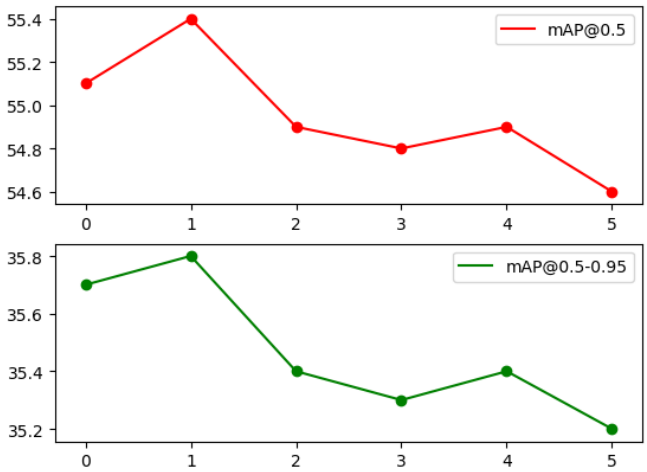



**Figure 14.** Exploring downsampling factor ($2^{power}$-mAP/%).

#### 5.4.4. Performance with Different Downsampling Methods

The above experiments show the necessity of downsampling the color feature map. Furthermore, we focus on which downsampling method can maximize the benefits. Based on the COCO dataset and the YOLOv5 network, we compare four different downsampling methods and consider the best one.

As shown in Table 8, considering both the cost and the detection performance of networks trained with sampled images, the best approach to downsampling is to directly take the top-left value in every cell of four pixels according to the index, which is what is adopted currently. We believe that color features are not expected to be very fine-grained and that taking the value directly is enough to satisfy requirements. In addition, the color feature in the proposed method represents four different but equal-level cases; so the feature value is not additive, and the pooling downsamplings lack meanings, not to mention the extra computation for the addition and comparison operations in a cell that are required.

**Table 8.** Exploring downsampling method.

| Factor | FLOPs | mAP@0.5/% | mAP@0.5–0.95% |
|---|---|---|---|
| Left-top | 0 | 55.4 | 35.8 |
| Right-down | 0 | 55.0 | 35.4 |
| Max-pooling | 3 | 55.2 | 35.5 |
| Min-pooling | 3 | 55.0 | 35.5 |
| Average-pooling | 5 | 54.9 | 35.5 |

#### 5.4.5. Performance with Orders of Downsampling

The meaning of downsampling the color feature map has been demonstrated, and we have determined the best downsampling method and its factors. In addition, we observe that there are two paths to obtain the color feature map of a downsampled image. One is to obtain a full resolution color feature map from the original image and then to downsample it, just as described in the Section 4. The other is to first downsample the original image with the factor and then to directly extract the color feature. Based on the COCO dataset and the YOLOv5 network, we compare these two approaches and explore the influence of different orders of downsampling.

As shown in Table 9, the order of first extracting the color feature map from the full-resolution gray image and then downsampling the map is better than the opposite on the performance of the downstream visual task. Moreover, considering the demands of the structural information branch for the full-resolution gray image, it does not need to generate extra downsampled gray images to increase the space occupation during sampling.

Combining these two points, we are convinced that it is optimal to extract the color features from the full-resolution gray image first and to downsample the map second.

**Table 9.** Exploring the order of downsampling.

| Downsample | mAP@0.5/% | mAP@0.5–0.95% |
|---|---|---|
| Gray Image | 54.9 | 35.3 |
| Color Feature Map | 55.4 | 35.8 |

#### 5.4.6. Significance Test

We present the statistical significance of the performance differences among the compared models in Table 10, which shows the results of a one-sided *t*-test on the predicted results of the models.

The null hypothesis was that the performance of the row model and the column model were indistinguishable at the 95% confidence level, indicated by '-' in the table. The alternate hypothesis states that the performance of the row model and the column model were different with 95% confidence, where a value 1 (or 0) indicates the row model is superior (or inferior) compared to the column model. As shown in Table 10, the results prove that our method is statistically advanced compared with grayscale images.

**Table 10.** One-sided *t*-test results at 95% confidence level. The number X\Y indicates the predicted results. A value of 1 (or 0) indicates that the row algorithm is statistically superior (inferior) to the column algorithm, while '-' indicates the algorithms are statistically indistinguishable.

| Method | RGB | Gray | Ours |
|---|---|---|---|
| RGB | -\- | 1\0 | 1\0 |
| Gray | 0\1 | -\- | 0\1 |
| Ours | 0\1 | 1\0 | -\- |

#### 5.4.7. Convergence Speed

We plot the convergence curve diagram as illustrated in Figure 15. As shown in the figure, the model converges at a normal speed, achieving stable slight oscillations in loss around 150 epochs. This demonstrates the rationality and applicability of our method.

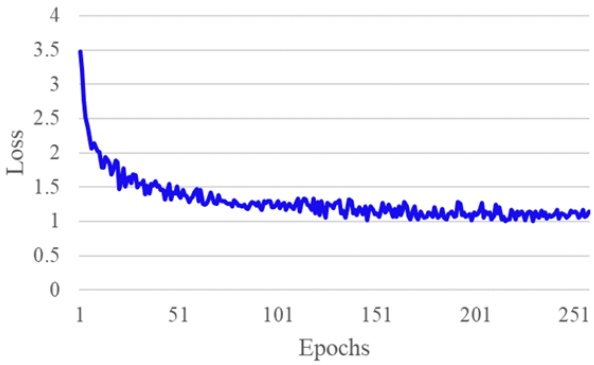



**Figure 15.** Convergence curve of training model.

### *5.5. Effect Visualization*

To demonstrate the effectiveness of the proposed method in enhancing the performance of visual analysis tasks, we analyze the feature maps generated by the YOLOv5 network after processing the sampled image, as shown in Figure 16.

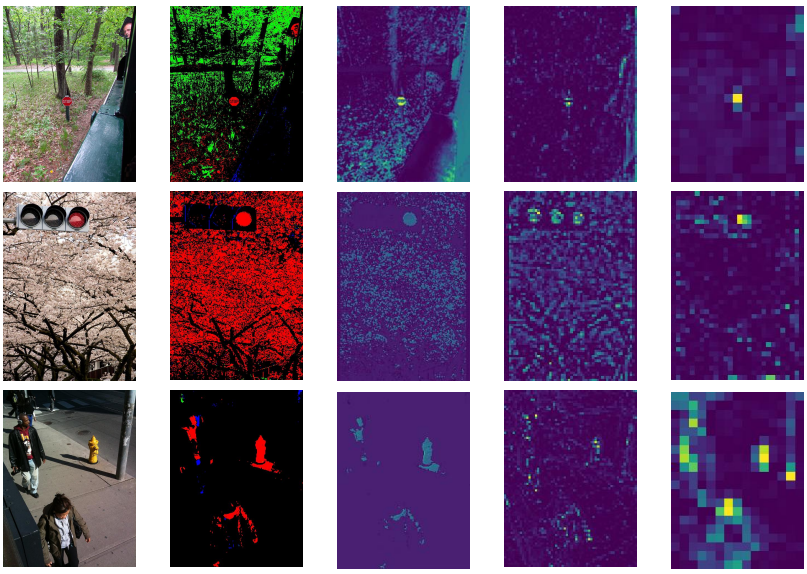


**Figure 16.** Responses in shallow, middle, and deep feature maps with the sampled images as input.

The proposed method aims to highlight key regions by leveraging color information while reducing the amount of data. As observed from the analysis, the sampled image consisting of gray image and color features generates strong responses in the shallow, middle, and deep layers of the deep network at the position of the red traffic sign. This indicates that the proposed method can effectively extract and utilize color features to achieve improved performance in visual analysis tasks. Therefore, the proposed method can be employed as a reliable technique for enhancing computer vision performance by focusing on key regions while optimizing data utilization. Similarly, with the sampled result of ISDCC as input, the vision algorithm can generate strong responses at the locations of the traffic lights and the fire hydrant.

## 6. Conclusions

We present a novel image sampling method, called image sampling based on the dominant color component (ISDCC), which utilizes a concise color feature and a gray image representation to reduce the performance loss of visual tasks while compressing the data size for computer vision requirements. Our method was extensively evaluated on widely used datasets, such as COCO, VOC, ImageNet, and Cifar, using state-of-the-art deep models. The results of our experiments demonstrate the effectiveness and generality of ISDCC. Nevertheless, the complexity of IoT applications and transceiver systems imposes limitations on the practical use of sampling methods in real-world scenarios. To address this issue, we propose integrating the ISDCC method with encoders and applying it to computer vision tasks where image acquisition and processing are separated. This integration will expand the potential applications of our approach. However, one limitation of this work is the lack of practical deployment using actual image/video coding systems. To overcome this limitation, we suggest considering the adoption of an H.264/AVC-related Macroblock sampled strategy. Additionally, implementing a more efficient sampling strategy for large areas of the same color could significantly reduce the required data quantities.

**Author Contributions:** Methodology, S.W. and L.W.; Validation, S.W. and J.C.; Formal analysis, F.L.; Writing—original draft, S.W. and J.C.; Writing—review & editing, F.L. and L.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References


1. Jain, D.K.; Zhao, X.; González-Almagro, G.; Gan, C.; Kotecha, K. Multimodal pedestrian detection using metaheuristics with deep convolutional neural network in crowded scenes. *Inf. Fusion* **2023**, *95*, 401–414. [CrossRef]
2. Zivkovic, M.; Bacanin, N.; Antonijevic, M.; Nikolic, B.; Kvascev, G.; Marjanovic, M.; Savanovic, N. Hybrid CNN and XGBoost Model Tuned by Modified Arithmetic Optimization Algorithm for COVID-19 Early Diagnostics from X-ray Images. *Electronics* **2022**, *11*, 3798. [CrossRef]
3. Nyquist, H. Certain Topics in Telegraph Transmission Theory. *Trans. Am. Inst. Electr. Eng.* **1928**, *47*, 617–644. [CrossRef]
4. Wallace, G. The JPEG still picture compression standard. *IEEE Trans. Consum. Electron.* **1992**, *38*, xviii–xxxiv. [CrossRef]
5. Cui, J.; Li, F.; Wang, L. Image Sampling for Machine Vision. In Proceedings of the CAAI International Conference on Artificial Intelligence, Beijing, China, 27–28 August 2022.
6. Terzopoulos, D.; Vasilescu, M. Sampling and reconstruction with adaptive meshes. In Proceedings of the Computer Vision and Pattern Recognition, Maui, HI, USA, 3–6 June 1991; pp. 70–75.
7. Eldar, Y.; Lindenbaum, M.; Porat, M.; Zeevi, Y. The farthest point strategy for progressive image sampling. *IEEE Trans. Image Process.* **1997**, *6*, 1305–1315. [CrossRef] [PubMed]
8. Ramoni, G.; Carrato, S. An adaptive irregular sampling algorithm and its application to image coding. *Image Vis. Comput.* **2001**, *19*, 451–460. [CrossRef]
9. Wei, L.; Wang, R. Differential domain analysis for non-uniform sampling. *ACM Trans. Graph.* **2011**, *30*, 1–10.
10. Marvasti, F.; Liu, C.; Adams, G. Analysis and recovery of multidimensional signals from irregular samples using nonlinear and iterative techniques. *Signal Process* **1994**, *36*, 13–30. [CrossRef]
11. Devir, Z.; Lindenbaum, M. Blind adaptive sampling of images. *IEEE Trans. Image Process.* **2012**, *21*, 1478–1487. [CrossRef]
12. Vipula, S.; Navin, R. Data Compression using non-uniform sampling, 2007. In Proceedings of the International Conference on Signal Processing, Chennai, India, 22–24 February 2007; pp. 603–607.
13. Laurent, D.; Nira, D.; Armin, I. Image compression by linear splines over adaptive triangulations. *Signal Process.* **2006**, *86*, 1604–1616.
14. Chen, W.; Ioth, S.; Shiki, J. Irregular sampling theorems for wavelet subspace. *IEEE Trans. Inf. Theory* **1998**, *44*, 1131–1142. [CrossRef]
15. Liu, Y. Irregular sampling for spline wavelet. *IEEE Trans. Inf. Theory* **1996**, *42*, 623–627.
16. Bahzad, S.; Nazanin, R. Model-based nonuniform compressive sampling and recovery of natural images utilizing a wavelet-domain universal hidden Markov model. *IEEE Trans. Signal Process* **2017**, *65*, 95–104.
17. Lorenzo, P.; Lorenzo, G.; Pierre, V. Image compression using an edge adapted redundant dictionary and wavelets. *Signal Process.* **2006**, *86*, 444–456.
18. Oztireli, A.; Alexa, M.; Gross, M. Spectral sampling of manifolds. *AMC Trans. Graph.* **2010**, *29*, 1–8. [CrossRef]
19. Sochen, N.; Kimmel, R.; Malladi, R. A general framework for low level vision. *IEEE Trans. Image Process.* **1998**, *7*, 310–318. [CrossRef] [PubMed]
20. Cheng, S.; Dey, T.; Ramos, E. A manifold reconstruction from point samples. In Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Vancouver, BC, Canada, 23–25 January 2005; pp. 1018–1027.
21. Saucan, E.; Appleboime, E.; Zeevi, Y. Geometric approach to sampling and communication. *Sampl. Theory Signal Image Process.* **2010**, *11*, 1–24. [CrossRef]
22. Krishnamoorthi, R.; Seetharaman, K. Image compression based on a family of stochastic models. *Signal Process.* **2007**, *87*, 408–417. [CrossRef]
23. Ji, S.; Xue, Y.; Lawrence, C. Bayesian compressive sensing. *IEEE Trans. Signal Process* **2008**, *56*, 2346–2356. [CrossRef]
24. Matthew, M.; Robert, N. Near-optimal adaptive compressed sensing. *IEEE Trans. Inf. Theory* **2014**, *60*, 4001–4012.
25. Ali, T.; Farokh, M. Adaptive Sparse Image Sampling and Recovery. *IEEE Trans. Comput. Imaging* **2018**, *4*, 311–325.
26. Dai, Q.; Henry, C.; Emeline, P.; Oliver, C.; Marc, W.; Aggelos, K. Adaptive Image Sampling Using Deep Learning and Its Application on X-Ray Fluorescence Image Reconstruction. *IEEE Trans. Multimed.* **2020**, *22*, 2564–2578. [CrossRef]
27. Wang, Z.; Li, F.; Xu, J.; Pamela, C. Human-Machine Interaction Oriented Image Coding for Resource-Constrained Visual Monitoring in IoT. *IEEE Internet Things J.* **2022**, *9*, 16181–16195. [CrossRef]
28. Mei, Y.; Li, L.; Li, Z.; Li, F. Learning-Based Scalable Image Compression with Latent-Feature Reuse and Prediction. *IEEE Trans. Multimed.* **2022**, *24*, 4143–4157. [CrossRef]
29. Muhammad, H.; Greg, S.; Norimichi, U. Task-Driven Super Resolution: Object Detection in Low-resolution Images. *arXiv* **2018**, arXiv:1803.11316.
30. Muhammad, W.; Bernhard, S.; Michael, H. The Unreasonable Effectiveness of Texture Transfer for Single Image Super-resolution. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 80–97.
31. Maneet, S.; Shruti, N.; Richa, S.; Mayank, V. Dual Directed Capsule Network for Very Low Resolution Image Recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; pp. 340–349.
32. Satoshi, S.; Motogiro, T.; Kazuya, H.; Takayuki, O.; Atsushi, S. Image Pre-Transformation for Recognition-Aware Image Compression, 2019. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–29 September 2019; pp. 2686–2690.

33. Vivek, S.; Ali, D.; Davy, N.; Michael, B.; Luc, V.; Rainer, S. Classification Driven Dynamic Image Enhancement. In Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4033–4041.
34. Jonghwa, Y.; Kyung-Ah, S. Enhancing the Performance of Convolutional Neural Networks on Quality Degraded Datasets. In Proceedings of the International Conference on Digital Image Computing: Techniques and Applications (DICTA), Sydney, Australia, 29 November–1 December 2017; pp. 1–8.
35. Ren, K.; Gao, Y.; Wan, M.; Gu, G.; Chen, Q. Infrared small target detection via region super resolution generative adversarial network. *Appl. Intell.* **2022**, *52*, 11725–11737. [CrossRef]
36. Veena, M.; Sowmya, K.; Uma, K.; Divyalakshmi, K.; Rajendra, A. An empirical study of preprocessing techniques with convolutional neural networks for accurate detection of chronic ocular diseases using fundus images. *Appl. Intell.* **2023**, *53*, 1548–1566.
37. Chen, J.; Zeng, Z.; Zhang, R.; Wang, W.; Zheng, Y.; Tian, K. Adaptive illumination normalization via adaptive illumination preprocessing and modified weber-face. *Appl. Intell.* **2019**, *49*, 872–882. [CrossRef]
38. Zhou, J.; Zhang, D.; Zhang, W. Underwater image enhancement method via multi-feature prior fusion. *Appl. Intell.* **2022**, *52*, 16435–16457. [CrossRef]
39. Xu, X.; Zhan, W.; Zhu, D.; Jiang, Y.; Chen, Y.; Guo, J. Contour information-guided multi-scale feature detection method for visible-infrared pedestrian detection. *Entropy* **2023**, *25*, 1022. [CrossRef]
40. Hossein, T.; Peyman, M. Learning to Resize Images for Computer Vision Tasks. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, BC, Canada, 11–16 October 2021; pp. 487–496.
41. Jia, D.; Wei, D.; Richard, S.; Li, L.; Kai, L.; Li, F. ImageNet: A large-scale hierarchical image database, 2009. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
42. Chen, Z.; Bernard, G. ThumbNet: One Thumbnail Image Contains All You Need for Recognition, 2020. In Proceedings of the 28th ACM International Conference on Multimedia ACM, Seattle, WA, USA, 12–16 October 2020; pp. 1506–1514.
43. Chen, T.; Lin, L.; Zuo, W.; Luo, X.; Zhang, L. Learning a Wavelet-like Auto-Encoder to Accelerate Deep Neural Networks, 2017. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 6722–6729.
44. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. PointCNN: Convolution On X -Transformed Points. In Proceedings of the Advances in Neural Information Processing Systems (NIPS 2018), Montreal, QC, Canda, 3–8 December 2018; Volume 31.
45. Qi, C.; Litany, O.; He, K.; Guibas, L. Deep Hough Voting for 3D Object Detection in Point Clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 october—2 November 2019; pp. 9276–9285.
46. Lang, I.; Manor, A.; Avidan, S. SampleNet: Differentiable Point Cloud Sampling. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 7575–7585. [CrossRef]
47. Huang, T.; Zhang, J.; Chen, J.; Liu, Y.; Liu, Y. Resolution-Free Point Cloud Sampling Network with Data Distillation. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–24 October 2022; pp. 54–70.
48. Zhou, W.; Yang, Q.; Jiang, Q.; Zhai, G.; Lin, W. Blind Quality Assessment of 3D Dense Point Clouds with Structure Guided Resampling. *arXiv* **2022**, arXiv:2208.14603.
49. Yang, Z.; Qiu, Z.; Fu, D. DMIS: Dynamic Mesh-based Importance Sampling for Training Physics-Informed Neural Networks. *arXiv* **2022**, arXiv:2211.13944.
50. Lin, T.; Marie, M.; Balongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollar, P.; Zitnick, L. Microsoft COCO: Common objects in context. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
51. Itti, L.; Koch, C.; Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **1998**, *20*, 1254–1259. [CrossRef]
52. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
53. Joseph, R.; Santosh, D.; Ross, G.; Ali, F. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
54. Joseph, R.; Ali, F. YOLO9000: Better, faster, stronger, 2017. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
55. Joseph, R.; Ali, F. YOLOv3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
56. Gleen, J. YOLOv5. 2020. Available online: https://github.com/ultralytics/yolov5 (accessed on 1 March 2023).
57. Ren, S.; He, K.; Ross, G.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef]
58. Zhou, X.; Koltun, V.; Krähenbühl, P. Tracking Objects as Points. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 474–490.
59. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]
60. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Learning Representation, San Diego, CA, USA, 7–9 May 2015; pp. 1–14.
61. Huang, G.; Liu, Z.; Laurens, V.; Kilian, Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269.