

Article

# A Mobile Image Aesthetics Processing System with Intelligent Scene Perception

Xiaoyan Zhao <sup>1,2</sup> , Ling Shi <sup>1,2</sup>, Zhao Han <sup>1,2</sup> and Peiyan Yuan <sup>1,2,\*</sup> 

<sup>1</sup> School of Computer and Information Engineering, Henan Normal University, Xinxiang 453007, China; zhaoxiaoyan@htu.cn (X.Z.); 2108283070@stu.htu.edu.cn (L.S.); h24z@outlook.com (Z.H.)

<sup>2</sup> Big Data Engineering Laboratory for Teaching Resources, Xinxiang 453007, China

\* Correspondence: peiyan@htu.cn; Tel.: +86-15-9373-48382

**Abstract:** Image aesthetics processing (IAP) is used primarily to enhance the aesthetic quality of images. However, IAP faces several issues, including its failure to analyze the influence of visual scene information and the difficulty of deploying IAP capabilities to mobile devices. This study proposes an automatic IAP system (IAPS) for mobile devices that integrates machine learning and traditional image-processing methods. First, we employ an extremely computation-efficient deep learning model, ShuffleNet, designed for mobile devices as our scene recognition model. Then, to enable computational inferencing on resource-constrained edge devices, we use a modern mobile machine-learning library, TensorFlow Lite, to convert the model type to TFLite format. Subsequently, we adjust the image contrast and color saturation using group filtering, respectively. These methods enable us to achieve maximal aesthetic enhancement of images with minimal parameter adjustments. Finally, we use the InceptionResNet-v2 aesthetic evaluation model to rate the images. Even when employing the benchmark model with an accuracy of 70%, the score of the IAPS processing image is verified to be higher and more effective compared with a state-of-the-art smartphone's beautification function. Additionally, an anonymous questionnaire survey with 100 participants is conducted, and the result shows that IAPS enhances the aesthetic appeal of images based on the public's preferences.

**Keywords:** scene recognition; image attributes; image aesthetics processing; aesthetic evaluation; mobile devices



**Citation:** Zhao, X.; Shi, L.; Han, Z.; Yuan, P. A Mobile Image Aesthetics Processing System with Intelligent Scene Perception. *Appl. Sci.* **2024**, *14*, 822. <https://doi.org/10.3390/app14020822>

Academic Editor: Christos Bouras

Received: 7 December 2023

Revised: 11 January 2024

Accepted: 15 January 2024

Published: 18 January 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

With the rapid advancements and pervasiveness of smart devices with photographic capabilities, an increasing number of people now aim to take pictures and share them on the social media. Such postings become social cards and play a significant role in showcasing one's life [1]. As such, mobile photo-editing apps and collage creators have become popular [2]. The production of high-quality images requires post-processing, which requires professional photographers and other experienced technicians. The process, which is largely automated today, involves vital parameter adjustments based on image types [3]. Focusing on the aesthetic quality of images, image aesthetic processing systems (IAPS) have gained significant research attention [4].

Notably, IAP research has been very limited thus far. Image aesthetic processing (IAP) algorithms correct and enhance image content based on composition, contrast, color saturation, and brightness settings. Some extant models perform model training and prediction on the cloud [5–7]; however, it is difficult to deploy them on mobile devices, resulting in poor performance in terms of energy consumption and latency [8]. Other studies do not consider scene category information [9,10], which implies different aesthetic processing standards.

Considering these issues, we try to deploy an IAP deep-learning model on a resource-constrained mobile device for model inferencing and prediction. First, lightweight neural networks must be adapted to the computing and storage limitations of mobile devices [11–14].

For this purpose, we propose a light IAPS that enhances the aesthetics of images. The IAPS integrates professional photography knowledge and aesthetic features to process images based on their categories [15–17], with minimal parameter adjustments to achieve maximum enhancement effects. The significance and potential applications of this study are illustrated in Figure 1.

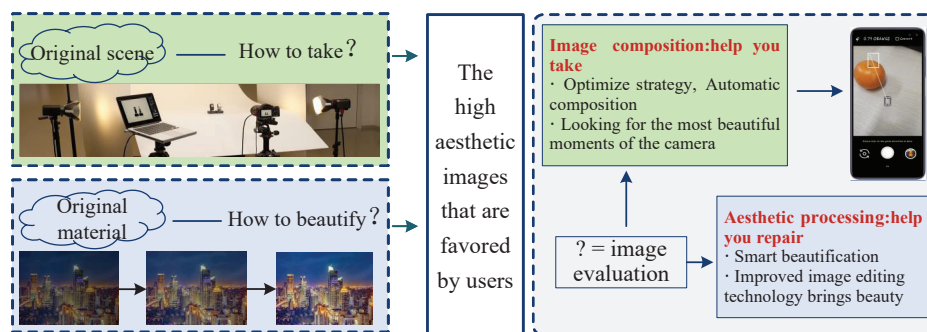


Figure 1. Significance and prospects of image aesthetic processing.

The main contributions of the proposed system are summarized as follows.

- A lightweight IAPS with intelligent scene perception, which automatically aesthetically processes images and reduces the computing burdens of resource-constrained mobile and edge devices.
- The ShuffleNet scene recognition model, which uses the lightweight TensorFlow Lite computer vision toolkit to predict scene categories.
- An automated classification-based aesthetic processing method that verifiably enhances the visual appeal of images.
- A novel contrast and color saturation model that leverages GPUImage filters to perform specific adjustments to image attributes.
- A local mean-square error filter algorithm and log curve procedure that achieves unsurpassed skin smoothing and whitening effects on human features.

The inclusion of acronyms is illustrated in Table 1 to enhance the readability of the study.

Table 1. A table of acronyms.

Complete Words	Acronyms
image aesthetics processing	IAP
IAP system	IAPS
aesthetic visual analysis	AVA
contrast-changed image database	CCID
generative adversarial network	GAN
group convolution	GC
true positives	TPs
true negatives	TNs
false positives	FPs
false negatives	FNs
post-training quantization	PTQ
graphics processing unit	GPU
Native Development Kit	NDK
Java Native Interface	JNI
mean opinion score	MOS

The remainder is organized as follows. Section 2 provides a review of related studies. Section 3 explains the design and training of the proposed system model, while Section 4 describes the model’s lighting. In Section 5, we outline the realization process of multi-attribute image aesthetic processing. Next, in Section 6, we present a functional

evaluation of the system. Finally, Section 7 summarizes the study and discusses prospects for future research.

## 2. Related Work

### 2.1. Scene Recognition Models and Their Deployment

The importance of image attributes depends largely on the scene category. Therefore, the importance of attributes differs significantly based on the scene given suitable aesthetic performance. Wang et al. [18] trained VGGNet for scene recognition on the large-scale Places205 dataset and achieved state-of-the-art performance. Liang et al. [19] proposed a public scene recognition framework based on smartphone sensors, implemented the algorithm on a Linux server, and deployed it for Android devices. Luo et al. [20] proposed a lightweight indoor scene recognition algorithm that used pressure changes to determine whether a user is inside a building.

It desires a lightweight framework to deploy a scene recognition model on resource-constrained terminal and edge devices. TensorFlow Lite is a lightweight, portable solution for mobile and embedded devices [21], and it enables machine learning predictions to be made in resource-constrained environments. Yuan et al. [22] optimized the deployment of image classification models for recognition prediction on mobile devices via federated collaborative training between edge servers and mobile devices. Ahmed et al. [23] ran deep learning algorithms on the terminal user's smartphone using the lightweight MobileNet network with TensorFlow Lite compression and quantization to reduce the edge device's memory usage and computation time. Handhayani et al. [24] developed a practical Android mobile application for garbage classification using the EfficientNet Lite model.

### 2.2. Image Aesthetic Processing

To make the processed image more visually appealing, IAP entails the automatic correction or enhancement of an image's aesthetic content. For example, Zhang et al. [25] proposed a collaborative deep reinforcement learning model for automatic image cropping. Wang et al. [26] designed a two-branch neural network for attention box prediction and aesthetic assessments. Li et al. [27] proposed an adversarial reinforcement learning model for fast aesthetic perception. Murray et al. [28] built the first large-scale aesthetic visual analysis (AVA) dataset, which included over 250,000 images with rich metadata and extensive aesthetic ratings for each image, semantic labels for over 60 categories. Alongside image composition, traditional image attribute adjustment methods are also crucial. Gu et al. [29] constructed a new large-scale specialized contrast-changed image database (CCID2014) and developed an image quality metric for contrast changes. Wang et al. [30] proposed an image enhancement method that used content-adaptive contrast improvements and color saturation adjustments to enhance the effects of outdoor images. Li et al. [31] applied a dual-input/output generative adversarial network (GAN) to deploy BeautyGAN, which generates pleasant facial makeup and accurate transfer results. Aydin et al. [32] used statistical methods to develop an aesthetic attribute perception system that mainly includes rating and optimizing five aesthetic attributes of an image: color, depth of field, sharpness, brightness, and focus. However, these aesthetic processing algorithms do not address the requirements of different scene categories.

### 2.3. Image Aesthetic Attribute Selection Standard

To achieve maximum visual improvement with minimal parameter adjustments, it is necessary to assign appropriate image attributes for aesthetic enhancements based on different scene categories. Specific standards were based on the following prior research.

**Factors affecting image aesthetics:** Ma et al. [16] found that when users take photos of the ocean, they tend to present the image horizontally to showcase the broad foreground. In contrast, to highlight their structural lines and shapes, users use vertical poses when photographing buildings. Le et al. [17] analyzed the relationship between aesthetic image attributes and overall aesthetics, finding that lighting and color attributes (e.g., balance

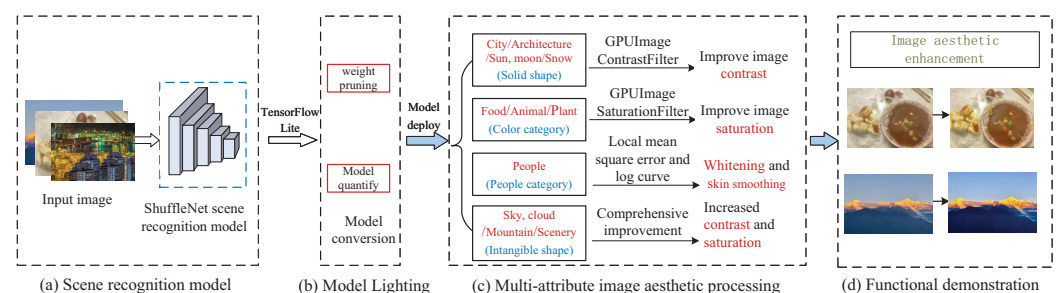
and saturation) are more closely related to the overall aesthetic in natural rural scenes. It is associated with the rich and vivid colors usually seen in natural landscapes. In contrast, in architectural and urban scenes, more attention is paid to presenting a building's sense of line and color contrast. These findings provide a deeper understanding of the relationship between images and their general aesthetics.

**Professional results:** In commercial applications, the smartphone Xiaomi 12S Ultra and the German company Leica provide Leica native dual image quality. On the one hand, classic Leica quality factors reflect strong overall contrast, neutral color restraint, and realistic, soft details; on the other hand, vivid Leica quality factors reflect bright and transparent general photos with vibrant and full colors.

Based on relevant studies, the proposed system classifies city/architecture/sun and moon/snow images as solid shape categories. It can achieve overall natural, realistic pictures by enhancing moderate contrast parameters, emphasizing contours and lines, and strengthening color contrast. Second, animal/plant/food is included in the color category. By enhancing the moderate color saturation parameters, those images' vividness and saturation of those images are further increased. For people categories, perform beautification operations, such as whitening and smoothing, are taken. Finally, sky/clouds/mountains/scenery categories are classified as intangible shape categories, and a comprehensive enhancement of color saturation and contrast is performed. By combining the advantages of the image attributes, the resulting scenery photographs exhibit heightened vibrancy and authenticity. Our category-based method leads to aesthetic enhancements in most everyday images, providing convenient, rapid, productive, and automatic IAP.

### 3. System Design

The purpose of this study is to develop a mobile IAPS that combines scene information and automatically categorizes different image types based on their various aesthetic standards. This capability fills a gap in the research by solving the problem of ordinary users needing the ability to apply advanced post-processing techniques to their images to improve their aesthetic quality. The overall system architecture is shown in Figure 2 and includes four main parts: scene recognition, model lightweighting, multi-attribute IAP, and functional demonstration.



**Figure 2.** Overview of the image aesthetic processing system.

The main challenge of IAP lies in identifying image attributes that are highly correlated with aesthetics and applying these attributes to image processing. Aesthetic attributes are closely related to photography principles. Because it is difficult for a computer system to recognize and interpret every photographic rule, choosing specific aesthetic attributes based on existing criteria is necessary. This study considers the following criteria to determine the necessary aesthetic attributes.

**Generality:** Although image sharpness is a factor specific to each photograph, the excessive enhancement of sharpness in certain scenes (e.g., smooth textures) can lead to the loss of image details, resulting in distortions and reduced quality. Therefore, we focus on universality and generality when choosing image attributes.

**Relationship with photographic rules:** Because our purpose is to enhance aesthetics by improving multiple image attributes, we leverage those that are frequently used in professional photography and strictly follow the general rules thereof.

**Clear definition:** Photography best practices and rules are usually expressed through examples rather than formulas or oral expression. To achieve automated processing of image aesthetics, we chose image attributes that could be defined as well as possible: contrast, color saturation, and portrait beauty.

### 3.1. Scene Recognition Model

To achieve scene recognition, we built a lightweight ShuffleNet network, whose key design philosophy is based on two principles. First, a group convolution (GC) is used as the basic unit with a pointwise convolution, which replaces the standard  $1 \times 1$  convolution and effectively reduces computational and parametric model costs. Second, a channel shuffle is used to solve the problem of the lack of shared feature information between groups caused by GCs and to enhance the model’s feature representability by facilitating channel information exchange. The overall ShuffleNet structure is illustrated in Figure 3.

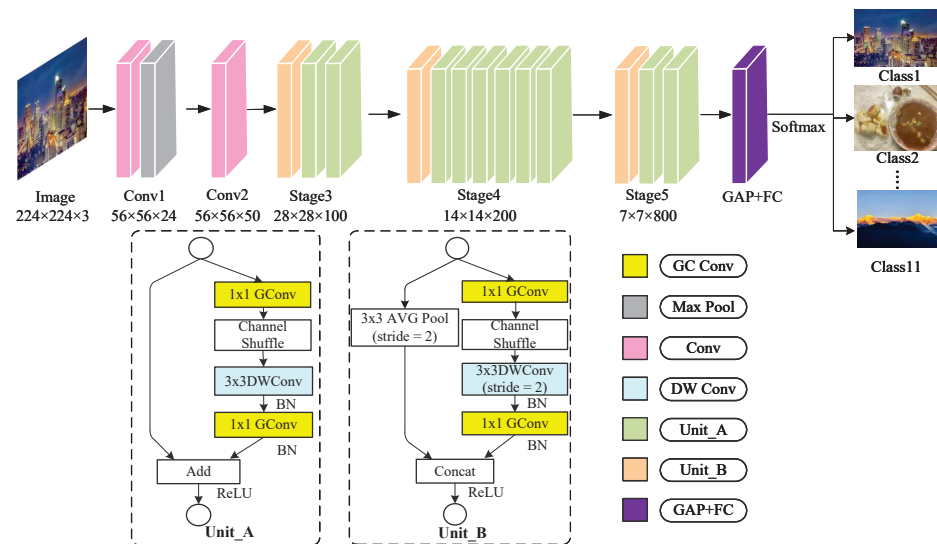


Figure 3. ShuffleNet V1 scene recognition model architecture.

The model consists of several basic stages, and each is composed of multiple essential units. Unit\_A has a stride of one, and Unit\_B requires downsampling with a stride of two. Conv, GC Conv, DW Conv, GAP, and FC denotes regular convolutional layer, group convolution, depthwise convolution, global average pooling, and the fully connected layer, respectively. SoftMax is set in the classifier layer to output the probability distributions of the 11 scene labels.

### 3.2. Model Training Parameters

During model training, it is essential to preprocess the input images using experimental data for augmentation and to prevent overfitting while enhancing robustness. The main data augmentation methods are as follows:

- (1) Images are cropped and resized to a unified size to facilitate training.
- (2) Horizontal flipping and random cropping are applied to some images.
- (3) Random rotation and scaling are applied to some images.
- (4) Image data are converted into tensors for easy model input.

Furthermore, to achieve the global optimum model as quickly as possible while preventing overfitting, we utilize EarlyStopping and ReduceLROnPlateau from the Keras callback library. The EarlyStopping mechanism saves the best model parameters throughout the learning process to prevent overfitting. To track changes in accuracy, we set the

monitor parameter to `val_acc`. If `val_acc` does not increase within 10 epochs, training is halted, and the weights at that point are used as model parameters.

If the initial learning rate is no longer applicable to the model after a certain number of epochs, the learning rate is decreased to improve training efficiency. We use the `ReduceLROnPlateau` callback function from Keras to adaptively adjust the learning rate. The monitor parameter is also set to `val_acc` to track changes in accuracy, and the learning rate is reduced if the accuracy does not increase within five epochs. These optimization methods effectively avoid model overfitting and improve training efficiency. The formula for decreasing the learning rate is as follows:  $lr$  and  $new\_lr$  represent the initial learning rate and the reduced learning rate, respectively.

$$new\_lr = lr * 0.1 \quad (1)$$

Using the callback function, we can achieve a better `val_acc` than not using it. Furthermore, the number of training epochs increases without triggering `EarlyStopping` too early, resulting in better optimization results.

Cross-entropy is used to evaluate the difference between the current probability distribution obtained from training and the true distribution and is highly suitable for the multiclass scene recognition tasks addressed in this study. Therefore, we use the categorical cross-entropy loss for multiclass scene recognition tasks, which may belong to one of many categories. The principal formula for the loss function is as follows:

$$\text{Loss} = - \sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i \quad (2)$$

As shown in Equation (2),  $y_i$  can be either zero or one. When  $y_i = 0$ , the result will be zero; only when  $y_i = 1$  will a corresponding classification result exist. Therefore, the categorical cross-entropy loss is only used for a predetermined result, which is typically used in combination with the `SoftMax` classifier for single-label classifications.

In this study, a general scene recognition model based on the `ShuffleNet` network is trained, and its parameters are saved as H5 files for subsequent use. Simultaneously, the classification categories are saved as corresponding label files to explain and evaluate the predicted model results. The `ShuffleNet` network has fewer parameters and a faster running speed, and accuracy is used as the primary evaluation indicator. Accuracy represents the ratio of the number of times the model correctly classifies results over all judgments. The number of times the model is classified correctly is denoted by the sum of true positives (TPs) plus true negatives (TNs), and the total number of decisions is denoted as false positives (FPs) plus false negatives (FNs) plus (TP + TN). TPs represents the number of times a positive sample is correctly classified as a positive sample; TNs represents the number of times a negative sample is correctly classified as a negative sample; FPs represents the number of times a negative sample is incorrectly classified into a positive sample; FN represents the number of times a positive sample is incorrectly classified as a negative sample.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (3)$$

#### 4. Model Lighting

To reduce latency and transmission costs, we apply a complete cloud-based model training process and deploy the model after lightweight processing on mobile devices. These mobile devices do not execute training tasks but instead rely on models for inferring and prediction. This approach significantly reduces latency and costs while also providing higher levels of privacy and security. During the model lighting process, we use the `TensorFlow Lite` library for post-training quantization (PTQ) and convert the model to the `TFLite` format for deployment on resource-limited mobile devices.

#### 4.1. Model Conversion

Different neural network frameworks can be saved as pretrained models in various formats. However, they always have numerous parameters with bloated architectures. For example, pretrained models can be saved as CKPT or PB files in TensorFlow framework, whereas for the PyTorch framework, they can be saved as PTH files. In this study, the Keras library is used, and the pretrained model is saved as an H5 file and converted to the corresponding TFLite file. The model conversion steps are as follows:

- (1) The load\_model method reads the model configuration and weight parameters of the H5 file, and the custom\_objects method configures the corresponding key-value pairs to enable Keras to recognize custom layer structures.
- (2) The TensorFlow Lite converter is used to convert the supported operators using TensorFlow Lite's built-in operators.
- (3) PTQ is used for model compression after training. It can quantize the weights from floating points to integers, optimize the model to reduce its size, reduce latency and power consumption, and convert the model to a TFLite file. Table 2 provides a detailed description of the size and functional effects before and after model conversion.

**Table 2.** Model compression files comparison.

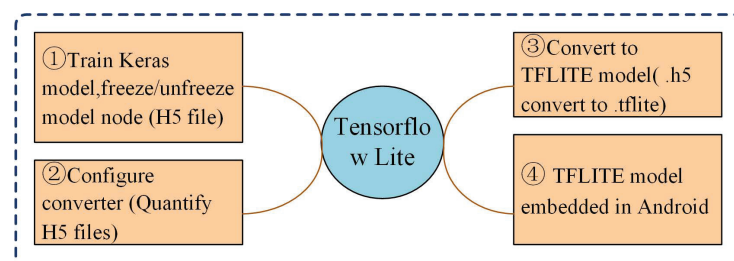
Type	Name	Function	Size (KB)
H5	ShuffleNet	Save weight information and graph information	24,576
Text	Labels	Save label file	1
TFLite	ShuffleNet	Save the structure and weight information	6860

#### 4.2. Model Deployment

In our system, to deploy a TFLite file on an Android mobile device, the following three steps are performed:

- (1) Import the TensorFlow Lite external link library using Gradle build tool and add dependencies as TFLite is not supported by an Android application programming interface.
- (2) Instantiate the interpreter, use multithreading for prediction, and accelerate with the Android API or a graphics processing unit (GPU) using the NnApiDelegate() method.
- (3) Obtain the input data and format them according to the data type required by the model, construct a data structure to store the output data, use the model for inferencing, and display the label with the highest probability on the interface.

In Figure 4, we describe the entire process of converting the H5 file saved using the Keras library into a TFLite file and deploying the prediction to an Android mobile device.



**Figure 4.** Model conversion deployment flowchart.

## 5. Multi-Attribute Image Aesthetic Processing

### 5.1. Comparison and Color Saturation Case

Contrast represents the difference between an image's brightest white and darkest black. The greater the difference, the stronger is the brightness and darkness contrast,

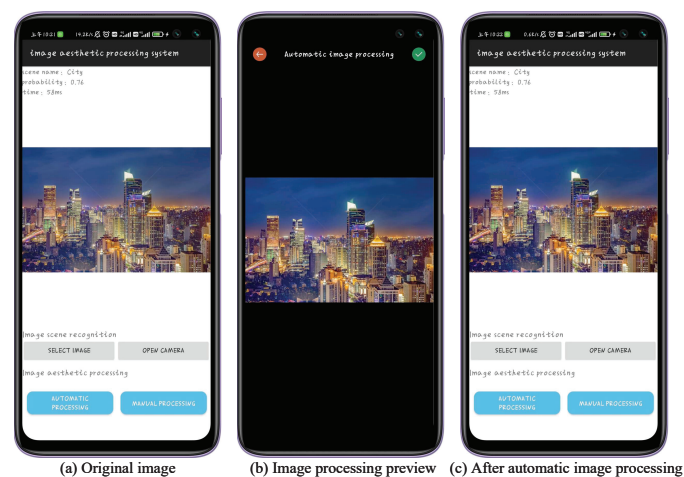
whereas a smaller difference indicates a more moderate contrast. Michelson's definition of contrast is given in Equation (4).

$$C_M = \frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}} \quad (4)$$

In this study,  $I_{\max}$  and  $I_{\min}$  refer to the maximum and minimum brightness of images, respectively. By adjusting the contrast parameter, the difference in the brightness of an image can be made more apparent, thereby enhancing the color and brightness. Conversely, if the contrast difference is small, an image appears grayish, which affects the viewing experience.

The aesthetic appeal of specific scene categories can be enhanced by automatically adjusting the contrast and color saturation attributes of input images. We import the GPUImage image processing open-source library and use the GPUImageFilter class inherited from GPUImageOutput to perform image filtering, following the protocol of GPUImageInput. In particular, we process the contrast and color saturation of the image, essentially changing the coordinates and colors of pixels, and we display the processed image after interpolating the static images or video frames. This process can make the color and linearity of the image more vivid, thereby enhancing its aesthetic appeal. The specific steps for enhancing contrast and color saturation are as follows:

- (1) Use Matisse's open-source library to select images from a photo album and load them into the program using the SetImageBitmap (bitmap) method.
- (2) Add the dependency implementation, jp.co.cyberagent.android.gpuimage:gpuimage-library:1.3.0, in the Build.gradle file to import the GPUImage library. In this study, GPUImageContrastFilter and saturation filter, which are derived from the GPUImageFilter, are used to manipulate and enhance the contrast and color saturation attributes of the image, all in the simple present tense.
- (3) Use mCameraInputFilter to draw the YUV filter data rendered on the SurfaceTexture to the FrameBuffer.
- (4) Draw the texture in the FrameBuffer to the preview screen, read the pixel data, and encapsulate it into a bitmap format for image saving. This completes the automatic processing of contrast and color saturation properties and saves the image to the system album. The effects of image contrast processing are illustrated in Figure 5.

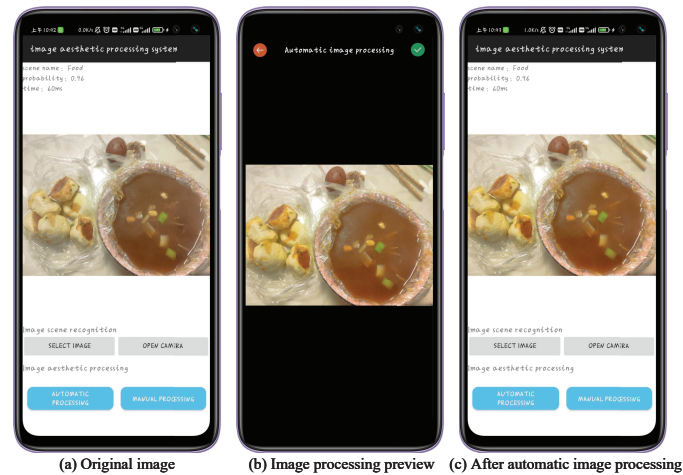


**Figure 5.** Aesthetic processing of image contrast attribute.

In the above scenario, when the image contrast is low, the overall impression shown in Figure 5a becomes dull and fails to leave a lasting impression. The image contrast is increased through automatic processing, as shown in Figure 5c, where the object lines are better highlighted, and the color contrast is significantly improved, enhancing the image's visual effects.

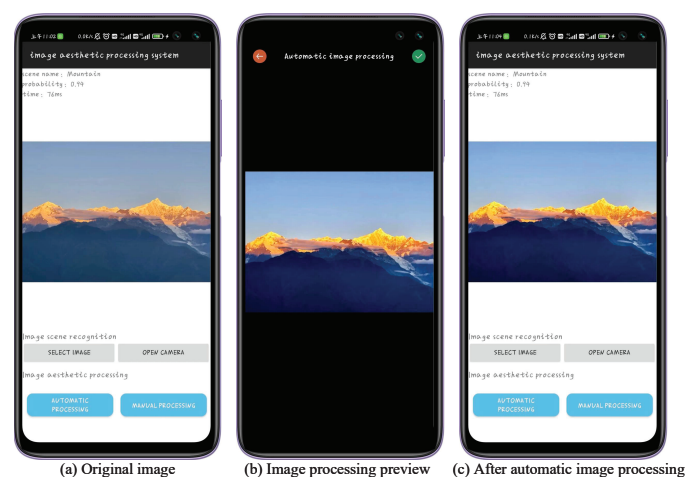


Color saturation refers to the vividness or purity of the colors in an image. Hence, saturation is determined by the proportion of colored and colorless (gray) parts in an image. The higher the color purity, the more vivid are the colors, while lower color purity can make colors appear dull. When taking pictures of animals, plants, food, and other scenes that involve multiple colors, increasing color saturation can make the photo more vivid and rich, enhancing its aesthetic appeal. The effects of processing the image color saturation are shown in Figure 6.



**Figure 6.** Aesthetic processing of image color saturation attribute.

Low color saturation can result in cold and dull visual effects in the above scenarios, as shown in Figure 6a. Automatic processing by the system enhances the vividness of the theme's color, resulting in a more complete and vibrant representation, as shown in Figure 6c, which effectively accentuates the theme. Finally, for intangible shape category images, enhancing both contrast and saturation can effectively improve the aesthetic appeal of the image. The effects of comprehensive aesthetic processing are shown in Figure 7.



**Figure 7.** Comprehensive processing of image contrast and color saturation.

In the above scenarios, color saturation is further enhanced after automatic image processing by the system. The contrast between light and dark in the scene is intensified.

### 5.2. Portrait Beauty Case

Like image denoising, skin smoothing is a basic facial beautification method that aims to remove blemishes and acne, and cover flaws and wrinkles. To enhance realism, this study adopts a local mean-square error filter algorithm for skin smoothing that preserves the

details of the image. The denoising principle of the skin smoothing algorithm is described as follows.

Suppose that there is a grayscale image of size  $N * M$ , and  $x_{ij}$  represents the pixel value at position  $(i, j)$ . Then, the local average value and local mean-square error within the  $(2n + 1)(2m + 1)$  window can be expressed as follows:

$$m_{ij} = \frac{1}{(2n+1)(2m+1)} \sum_{k=i-n}^{n+i} \sum_{l=j-m}^{m+j} x_{kl} \quad (5)$$

The formula for calculating the local variance  $v_{ij}$  is as follows:

$$v_{ij} = \frac{1}{(2n+1)(2m+1)} \sum_{k=i-n}^{n+i} \sum_{l=j-m}^{m+j} (x_{kl} - m_{ij})^2 \quad (6)$$

The filtering formula for the local mean-square error is as follows:

$$x_{i,j} = (1 - k)m_{ij} + kx_{ij} \quad (7)$$

where  $x_{ij}$  represents the pixel value of the input image, and coefficient  $k$  is expressed as follows:

$$k = \frac{v_{ij}}{v_{ij} + \sigma} \quad (8)$$

where  $\sigma$  is the parameter input by the user. The variance  $Var(x)$  is expressed as follows:

$$\begin{aligned} Var(x) &= \sum_{i=1}^n p_i (x_i - \mu)^2 \\ &= \frac{1}{n} \left( \sum_{i=1}^n x_i^2 - \frac{1}{n} (\sum_{i=1}^n x_i)^2 \right) \end{aligned} \quad (9)$$

If the variance value in Equation (9) is small for an image, then the area is smooth grayscale. Hence, the difference in grayscale values between pixels is small. Conversely, the variance increases when it is a local edge or high-frequency area; hence, the user's input parameters can be ignored. After filtering, the processed image is equivalent to the grayscale value of the input. Overall, this algorithm ensures image smoothness while preserving edge information and does not significantly impact the edges.

The importance of fair skin is high in modern aesthetics. Adaptive whitening has to be performed for portrait beautification, as the brightness of the skin is generally low. Additionally, the whitening process can moderately increase image brightness. In this study, the beauty algorithm uses a log curve to enhance the whitening effect. The log curve follows Weber–Fechner's law of visual characteristics, which aligns the whitening effect with human perception. The formula for this principle is as follows:

$$F(x, y) = \frac{\log(L(x, y) \times (\beta - 1) + 1)}{\log(\beta)} \quad (10)$$

where  $F(x, y)$  represents the image after the brightness adjustment,  $L(x, y)$  represents the input image, and  $\beta$  represents the brightness-adjustment parameter. The larger the  $\beta$  value, the greater the whitening effect.

The implementation steps of the portrait beauty function are as follows:

- (1) Import the MagicJni.cpp source file written in C++ as an independent beauty function module. This module includes skin smoothing and portrait whitening algorithms, which can effectively improve image quality.
- (2) Edit the CMakeLists.txt file and use the Android Native Development Kit (NDK) to convert the .cpp file into a .so file.
- (3) Use the Java Native Interface (JNI) to call the C++ function. Therefore, in Java, the MagicJni class is used to call the local mean-square error filter algorithm and the log curve whitening algorithm to achieve beauty processing.
- (4) Package the .so file into a .apk file using the NDK and run it on a smartphone.

Figure 8 illustrates the impact of portrait image beauty. In this situation, a scene recognition model is employed to detect people. By implementing skin smoothing and

whitening, the portrait’s aesthetic appeal can be improved while slightly enhancing the overall brightness of the image, thereby replacing the customary dependence on lighting in portrait photography. Furthermore, if a user is dissatisfied with the automatic aesthetic processing effect, the manual IAP function can be used to personalize the aesthetic impact by adjusting the image attribute parameters. The design of the manual image-processing interface is shown in Figure 8d. Finally, in Figure 9, we present the complete implementation process of IAPS in the form of a flowchart that is clear, concise, and easy to understand. It will provide useful references for the design and implementation of similar applications.

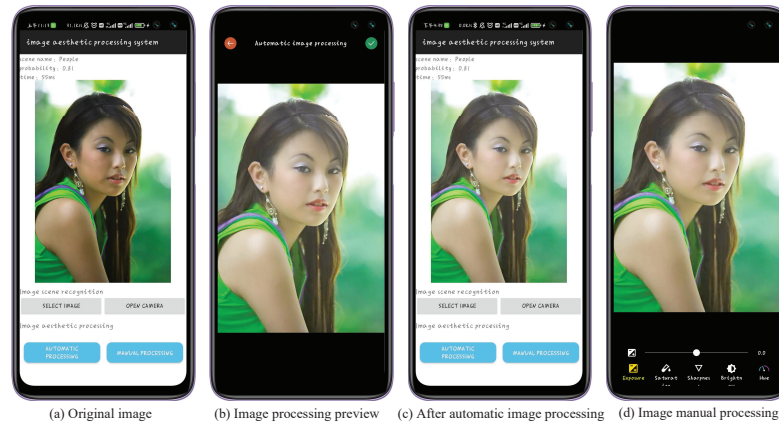


Figure 8. Image portrait beauty and manual parameter processing.

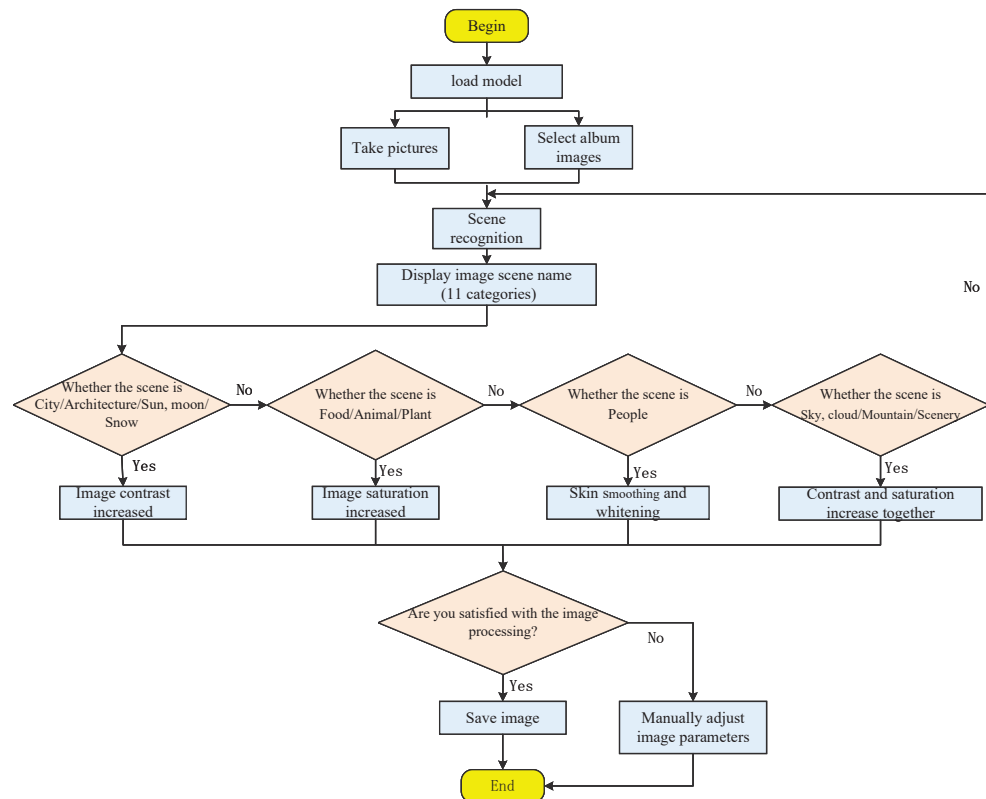


Figure 9. Image aesthetic processing flow design diagram.

## 6. Experiment Evaluation

### 6.1. Simulation Environment

In this study, the experiment was conducted on Ubuntu 20.04 with Python 3.8, TensorFlow 2.9.0, and CUDA 11.2. The model was trained on a computer equipped with a 24 GB

RTX A5000. The dataset for model training in this study drew on the TAD66K theme and a theme-oriented aesthetic dataset containing 66,000 images covering 47 popular themes [33]. We selected and reorganized 40,737 images from 11 major scene categories to construct a public scene recognition dataset. The specific categories are listed in Table 3.

**Table 3.** Composition categories of scene recognition dataset.

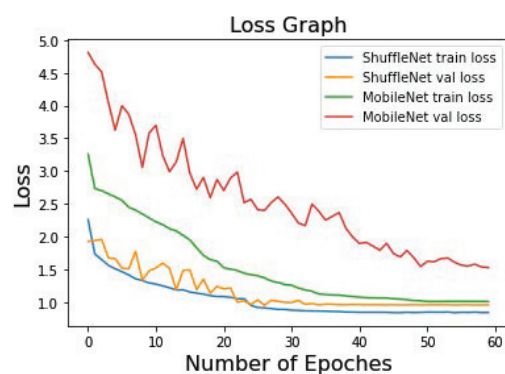
Scene Categories	Category Attributes	Training Numbers	Testing Numbers
City	Solid shape	3482	881
Architecture	Solid shape	2221	587
Sun, moon	Solid shape	3180	869
Snow	Solid shape	1120	291
Food	Color category	1076	291
Animal	Color category	3355	881
Plant	Color category	3496	878
People	People category	3225	832
Sky, cloud	Intangible shape	3200	833
Mountain	Intangible shape	4553	1172
Scenery	Intangible shape	3434	880
Total		32342	8395

## 6.2. Model Performance Evaluation

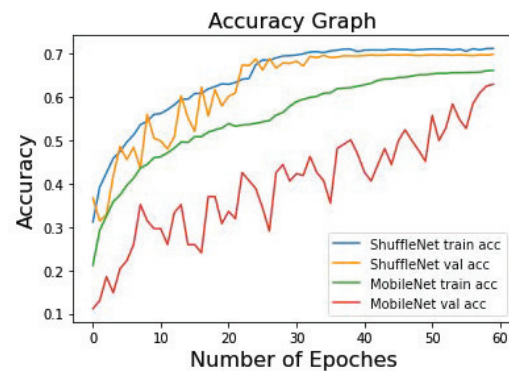
To enhance the comparative experiment of the model, we used the MobileNet network as a comparison model. An adaptive moment estimation (Adam) optimizer was used during the experiment for parameter updates, and the input images were uniformly normalized. The initial learning rate was set to 0.01, and 60 training epochs were performed with batch sizes of 32. The loss function during training is shown in Figure 10, where the global epochs were trained 60 times. At approximately 40 epochs, the model stopped oscillating and began to converge. The model structure and weight parameters were saved using the Keras deep learning library model and the Save () method. The accuracy curve of the training and validation processes is shown in Figure 11, where train\_acc represents the accuracy of the model during training, and val\_acc represents the accuracy of the model on the validation set. Table 4 lists the model training parameters and performance configuration.

**Table 4.** Model configuration and performance analysis.

Model	GPU	Dataset Number	Training Time (h)	Epoch	Parameters	Accuracy (%)
ShuffleNet.h5	RTX A5000	40,737	9.6	60	971,659	70

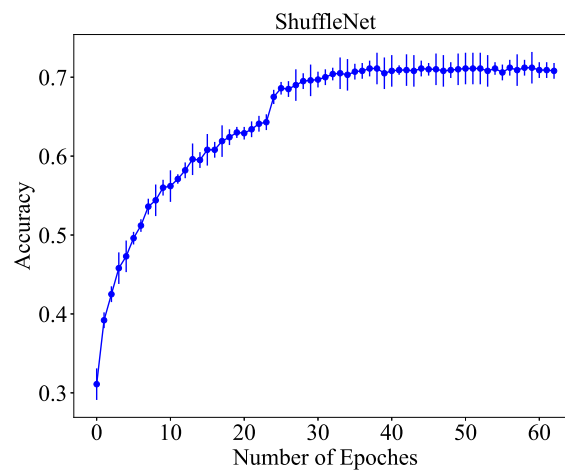


**Figure 10.** Training and validation loss.



**Figure 11.** Training and validation accuracy.

To reduce the bias of the experiment, we repeat experiments for fifty times and plot the standard deviation of the accuracy of ShuffleNet. As can be seen from Figure 12, the proposed ShuffleNet model can maintain a relatively stable accuracy of around 70%. It should be noted that this study mainly focuses on image aesthetic processing optimization after recognition and classification of the image, rather than the specific value of the accuracy. In other words, we can optimize image aesthetic processing with different accuracies as benchmarks.



**Figure 12.** Standard deviation of the accuracy of ShuffleNet.

### 6.3. System Function Evaluation

To verify the functional effectiveness of the IAPS accurately and objectively, we used the InceptionResNet-V2 aesthetic evaluation model [34] trained on the AVA dataset [28] to simulate the mean opinion score (MOS) of each image and output its image quality score. First, we extracted the feature maps of each block in the inception network, concatenated them, and used a regression model to predict the MOS value of the image. In Figure 13, the numbers in square brackets below each image represent the MOS predicted by the aesthetic evaluation model, reflecting the aesthetic quality of the image.

To improve the reliability of the functional test results, we designed a questionnaire survey that compared the before and after aesthetic processing images as reference options and asked ordinary users to make aesthetic choices to verify the practicality of this system. In the questionnaires for users, each class contains three images: the original image, the Redmi K30-processed image, and the image processed by IAPS. According to the images' scenes, the IAPS processes images from four aspects: contrast, saturation, portrait beauty, and contrast saturation. To illustrate these four processing aspects more conveniently, we select a scene for each aspect and use it as an example to represent this processing aspect. We thus select four most common scenes in daily life, respectively taking city, food,

people, and mountain as examples to show the contrast, saturation, portrait beauty and contrast saturation processing of images in this study. The ordinary users we randomly interviewed in public places are different in age, personality, occupation, etc., so they are more universal. We first divide the images in Figure 13 into four groups, (1), (2), (3), and (4), according to the corresponding rows. Then, we organize a questionnaire survey with 100 participants. The 100 participants vote on the aesthetics of these four groups of images, and select the image they think is the best quality. The findings show that most of the participants believed that the aesthetic beauty of the processed images was superior, thereby attesting to the system’s aesthetic processing method being more aligned with the general public’s aesthetic standards. These results further substantiate the efficacy of this feature for ordinary users and its practical applicability. The statistical results of the questionnaire survey are presented in Figure 14.

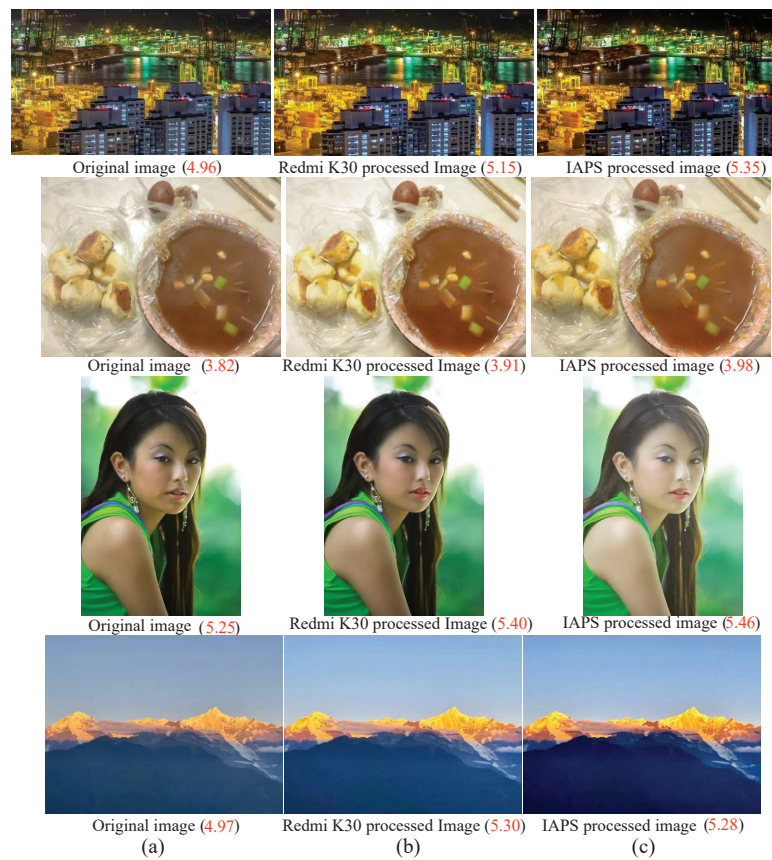


Figure 13. Image aesthetic processing function test ((a): Original image, (b): Redmi K30 processed image, (c): IAPS processed image).

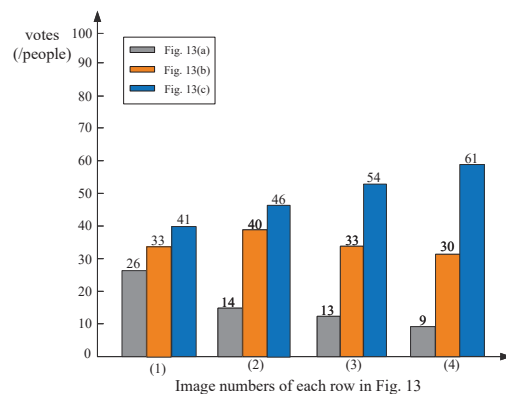


Figure 14. Statistics of the questionnaire survey results.

We also used the one-key beautification function on a state-of-the-art smartphone's Redmi K30 to process the images for experimental comparison, thereby highlighting the beautification effect of the model, as shown in Figure 13b. Figure 13 shows both the score comparison before and after IAPS processing and a subjective comparison with the beauty function of mainstream mobile phone systems. Improving image attributes for specific scene images through scene classification can significantly improve aesthetic and visual effects. The experimental results validated the effectiveness of the functionality of the system.

#### 6.4. Performance Testing

In order to better present the effectiveness and practicability of the IAP system, the performance tests are carried out on three actual phones, including processing time, memory occupation, CPU occupation and stability. The specific data are shown in Table 5.

**Table 5.** Performance testing on different phones.

Phone Model	Redmi k30	HONOR 30 Pro	OPPO Find X
CPU	Snapdragon 730G	Kirin 990	Snapdragon 845
RAM (GB)	8	8	8
Operating System	Android 11	Harmony 2.0.0	Android 10
Processing Time (ms)	63	57	52
Memory Occupation (MB)	73	70	71
CPU Occupation (%)	12	13	9
Stability	3000	3000	3000

In Table 5, the processing time of three actual phones is less than 1 s, which can give a better user experience. Then, we used the Android Monitor to record the memory occupation and CPU occupation of the application on three phones, all of which had small memory occupation and CPU occupation. Finally, we used the Monkey tool to simulate the pseudo-random user time flow, and conducted the stability test by generating 3000 random user operations. From the data in Table 5, during the entire stability testing process, the application did not crash. This indicates the robustness of the designed application. In addition, we observed minimal power consumption throughout the testing process.

## 7. Conclusions

We implemented our IAPS on mobile devices by combining machine learning with traditional image-processing techniques to incorporate image scene information. First, we constructed a general scene recognition dataset and a lightweight network, ShuffleNet. The model was converted using TensorFlow Lite and deployed on a mobile device. Subsequently, the image contrast and color saturation were adjusted using the GPUImage-ContrastFilter and the GPUImageSaturationFilter on the GPU of the device. A local mean-square error filter algorithm and a log curve were employed for skin smoothing and whitening. Even though the accuracy of the model is 70%, the aesthetic quality of the IAPS-processed image is high, which verifies the effectiveness of IAPS. Finally, we validated the functionality of the system using an anonymous questionnaire survey with 100 participants. The findings show that most of the participants believed that the aesthetic beauty of the processed images was satisfactory, indicating that the proposed IAPS is more aligned with the public aesthetic standards. However, our work still faces challenges, such as limited improvements in image aesthetics; moreover, personalized user needs was not considered. These challenges require further investigation in future research.

**Author Contributions:** Conceptualization, L.S.; Methodology, X.Z.; Validation, P.Y.; Writing—original draft, Z.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported in part by the National Natural Science Foundation of China under Grant No. 62072159 and No. 61902112, the Science and Technology Research Project of Henan province under Grant No. 222102210011 and No. 232102211061.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Permission for faces was obtained in the study.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

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

## References

1. Price, Z. Missing the Present: Nostalgia and the Archival Impulse in Gentrification Photography. *Arts* **2023**, *12*, 85. [\[CrossRef\]](#)
2. Valenzise, G.; Kang, C.; Dufaux, F. Advances and challenges in computational image aesthetics. In *Human Perception of Visual Information: Psychological and Computational Perspectives*; Springer: Cham, Switzerland, 2020; pp. 133–181.
3. Zhang, J.; Synave, R.; Delepoulle, S.; Cozot, R. Reconstructing Image Composition: Computation of Leading Lines. *J. Imaging* **2024**, *10*, 5. [\[CrossRef\]](#)
4. Deng, Y.; Loy, C.C.; Tang, X. Image aesthetic assessment: An experimental survey. *IEEE Signal Process. Mag.* **2017**, *34*, 80–106. [\[CrossRef\]](#)
5. Li, L.; Zhu, H.; Zhao, S.; Ding, G.; Lin, W. Personality-assisted multi-task learning for generic and personalized image aesthetics assessment. *IEEE Trans. Image Process.* **2020**, *29*, 3898–3910. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Chen, Q.; Zhang, W.; Zhou, N.; Lei, P.; Xu, Y.; Zheng, Y.; Fan, J. Adaptive fractional dilated convolution network for image aesthetics assessment. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 14114–14123.
7. Kim, H.-U.; Koh, Y.-J.; Kim, C.-S. PieNet: Personalized image enhancement network. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 374–390.
8. Štroner, M.; Urban, R.; Línková, L. Color-Based Point Cloud Classification Using a Novel Gaussian Mixed Modeling-Based Approach versus a Deep Neural Network. *Remote Sens.* **2024**, *16*, 115. [\[CrossRef\]](#)
9. Cai, J.; Gu, S.; Zhang, L. Learning a deep single image contrast enhancer from multi-exposure images. *IEEE Trans. Image Process.* **2018**, *27*, 2049–2062. [\[CrossRef\]](#) [\[PubMed\]](#)
10. Sheng, K.; Dong, W.; Ma, C.; Mei, X.; Huang, F.; Hu, B.-G. Attention-based multi-patch aggregation for image aesthetic assessment. In Proceedings of the 26th ACM international conference on Multimedia, Seoul, Republic of Korea, 22–26 October 2018; pp. 879–886.
11. Zhang, R.; Huang, Z.; Zhang, Y.; Xue, Z.; Li, X. MSGV-YOLOv7: A Lightweight Pineapple Detection Method. *Agriculture* **2024**, *14*, 29. [\[CrossRef\]](#)
12. Qin, B.; Zeng, Y.; Wang, X.; Peng, J.; Li, T.; Wang, T.; Qin, Y. Lightweight DB-YOLO Facemask Intelligent Detection and Android Application Based on Bidirectional Weighted Feature Fusion. *Electronics* **2023**, *12*, 4936. [\[CrossRef\]](#)
13. Liu, Y.; Zhao, Y.; Zhang, X.; Wang, X.; Lian, C.; Li, J.; Shan, P.; Fu, C.; Lyu, X.; Li, L.; et al. MobileSAM-Track: Lightweight One-Shot Tracking and Segmentation of Small Objects on Edge Devices. *Remote Sens.* **2023**, *15*, 5665. [\[CrossRef\]](#)
14. Xu, G.; Yin, X.; Li, X. Lightweight and Secure Multi-Message Multi-Receiver Certificateless Signcryption Scheme for the Internet of Vehicles. *Electronics* **2023**, *12*, 4908. [\[CrossRef\]](#)
15. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856.
16. Ma, S.; Wei, Z.; Tian, F.; Fan, X.; Zhang, J.; Shen, X.; Lin, Z.; Huang, J.; Měch, R.; Samaras, D.; et al. SmartEye: Assisting instant photo taking via integrating user preference with deep view proposal network. In Proceedings of the CHI Conference on Human Factors in Computing Systems, Glasgow, UK, 4–9 May 2019; pp. 1–12.
17. Le, Q.-T.; Ladret, P.; Nguyen, H.-T.; Caplier, A. Computational Analysis of Correlations between Image Aesthetic and Image Naturalness in the Relation with Image Quality. *J. Imaging* **2022**, *8*, 166. [\[CrossRef\]](#) [\[PubMed\]](#)
18. Wang, L.; Guo, S.; Huang, W.; Qiao, Y. Places205-vggnet models for scene recognition. *arXiv* **2015**, arXiv:1508.01667.
19. Liang, S.; Du, X.; Dong, P. Public scene recognition using mobile phone sensors. In Proceedings of the International Conference on Computing, Networking and Communications (ICNC), Kauai, HI, USA, 15–18 February 2016; pp. 1–5.
20. Luo, D.; Luo, H.; Zili, C. An indoor scene recognition algorithm based on pressure change pattern. In Proceedings of the 8th International Conference on Intelligent Computation Technology and Automation (ICICTA), Nanchang, China, 14–15 June 2015; pp. 149–152.
21. Li, S. Tensorflow lite: On-device machine learning framework. *J. Comput. Res. Dev.* **2020**, *57*, 1839.
22. Yuan, P.; Huang, R.; Zhang, J.; Zhang, E.; Zhao, X. Accuracy Rate Maximization in Edge Federated Learning with Delay and Energy Constraints. *IEEE Syst. J.* **2022**, *17*, 2053–2064. [\[CrossRef\]](#)



23. Ahmed, S.; Bons, M. Edge computed NILM: A phone-based implementation using MobileNet compressed by tensorflow lite. In Proceedings of the 5th International Workshop on Non-Intrusive Load Monitoring, Virtual Event, Japan, 18 November 2020; pp. 44–48.
24. Handhayani, T.; Hendryli, J. Leboh: An Android Mobile Application for Waste Classification Using TensorFlow Lite. In *Intelligent Systems and Applications: Proceedings of the 2022 Intelligent Systems Conference (IntelliSys) Volume 3*; Springer: Cham, Switzerland, 2022; pp. 53–67.
25. Zhang, X.; Li, Z.; Jiang, J. Emotion attention-aware collaborative deep reinforcement learning for image cropping. *IEEE Trans. Multimed.* **2020**, *23*, 2545–2560. [[CrossRef](#)]
26. Wang, W.; Shen, J.; Ling, H. A deep network solution for attention and aesthetics aware photo cropping. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *41*, 1531–1544. [[CrossRef](#)] [[PubMed](#)]
27. Li, D.; Wu, H.; Zhang, J.; Huang, K. Fast a3rl: Aesthetics-aware adversarial reinforcement learning for image cropping. *IEEE Trans. Image Process.* **2019**, *28*, 5105–5120. [[CrossRef](#)] [[PubMed](#)]
28. Murray, N.; Marchesotti, L.; Perronnin, F. AVA: A large-scale database for aesthetic visual analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 2408–2415.
29. Gu, K.; Zhai, G.; Lin, W.; Liu, M. The analysis of image contrast: From quality assessment to automatic enhancement. *IEEE Trans. Cybern.* **2015**, *46*, 284–297. [[CrossRef](#)] [[PubMed](#)]
30. Wang, S.; Cho, W.; Jang, J.; Abidi, M.A.; Paik, J. Contrast-dependent saturation adjustment for outdoor image enhancement. *JOSA A* **2017**, *34*, 7–17. [[CrossRef](#)] [[PubMed](#)]
31. Li, T.; Qian, R.; Dong, C.; Liu, S.; Yan, Q.; Zhu, W.; Lin, L. Beautygan: Instance-level facial makeup transfer with deep generative adversarial network. In Proceedings of the 26th ACM international conference on Multimedia, Seoul, Republic of Korea, 22–26 October 2018; pp. 645–653.
32. Aydın, T.O.; Smolic, A.; Gross, M. Automated aesthetic analysis of photographic images. *IEEE Trans. Vis. Comput. Graph.* **2014**, *21*, 31–42. [[CrossRef](#)] [[PubMed](#)]
33. He, S.; Zhang, Y.; Xie, R.; Jiang, D.; Ming, A. Rethinking Image Aesthetics Assessment: Models, Datasets and Benchmarks. In Proceedings of the 31st International Joint Conference on Artificial Intelligence, Vienna, Austria, 23–29 July 2022.
34. Hosu, V.; Goldlucke, B.; Saupe, D. Effective aesthetics prediction with multi-level spatially pooled features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 9375–9383.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.