


## Article

# Machine-Learning-Based Automatic Metallographic Grading System for High-Gloss Anodized Aluminum Profiles

Xuda Xu <sup>1,2</sup>, Feng Jiang <sup>1,\*</sup>, Lurong Li <sup>2</sup>, Hongfeng Huang <sup>3</sup> , Fei Yang <sup>2</sup> and Chunli Jiang <sup>4</sup>

<sup>1</sup> School of Material Science and Engineering, Central South University, Changsha 410083, China; xuxuda1982@163.com

<sup>2</sup> Guangdong Hoshion Aluminium Co., Ltd., Zhongshan 528463, China; lii\_ly@foxmail.com (L.L.); yf2767939446@gmail.com (F.Y.)

<sup>3</sup> College of Materials Science and Engineering, Guilin University of Technology, Guilin 541004, China; hhfeng@glut.edu.cn

<sup>4</sup> Guangdong Institute of Special Equipment Inspection and Research Zhongshan Branch, Zhongshan 528400, China; jiangchunli@zsjcy.com

\* Correspondence: jfeng2@csu.edu.cn

**Abstract:** The excellent “mirror” effect of medium and high-strength aluminum alloy profiles from the 6-series, achieved through anodizing, is highly valued by customers. Metallographic analysis is a key method for predicting the anodizing effect. However, traditional metallographic analysis methods suffer from unstable accuracy and low efficiency. To address these issues, this paper successfully develops a metallographic grading system by constructing a dataset and integrating computer vision with machine-learning techniques. Based on grain classification, the system automatically determines the metallographic grade by analyzing the proportion of good grain areas. After applying SMOTE sampling and 10-fold cross-validation to the machine-learning algorithm, we conducted a comparative analysis of the model’s performance from the perspectives of accuracy, good grain recall rate, bad grain recall rate, and AUC. The XGBoost model, selected as the final predictive model from 18 machine-learning models due to its superior performance, achieved a grain classification accuracy of 96.21% and a good grain recall rate of 98.07%. Both the accuracy and good grain recall standard deviations were less than 0.02. These results indicate that the model can effectively distinguish between good and bad grains with high robustness. Additionally, the average time for metallographic grading is less than 9 s. In comparison to the instability of traditional manual grading, this method significantly enhances both the accuracy and efficiency of metallographic analysis while also reducing grading costs.

**Keywords:** metallographic grading; aluminum alloy; computer vision; machine learning; aluminum profiles for mobile phones; anodizing quality prediction; automated inspection; high gloss



Academic Editor: Theodore E. Simos

Received: 25 February 2025

Revised: 20 March 2025

Accepted: 21 March 2025

Published: 23 March 2025

**Citation:** Xu, X.; Jiang, F.; Li, L.; Huang, H.; Yang, F.; Jiang, C. Machine-Learning-Based Automatic Metallographic Grading System for High-Gloss Anodized Aluminum Profiles. *Symmetry* **2025**, *17*, 482. <https://doi.org/10.3390/sym17040482>

**Copyright:** © 2025 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

A variety of materials are used in mobile phone structural components, such as glass, plastic, steel, aluminum alloy, titanium alloy, etc. Among them, since Apple Inc. first utilized aluminum alloy as the back cover and frame material for the iPhone 5 in 2012, this material has gained widespread adoption due to its superior physical properties and ease of processing. According to statistics, the global annual smartphone shipments have remained stable at over 1.1 billion units over the past decade, driving the consumption of more than 350,000 tons of aluminum alloy annually, accounting for approximately 0.5% of global aluminum production. In recent years, “mirror” anodizing treatment has emerged

as the preferred finishing solution for high-end smartphone frames. During the production process, aluminum profile suppliers typically use metallographic analysis to predict the “mirror” effect after anodizing in order to control production quality. This predictive ability is based on the strong correlation between microstructural features and anodizing performance: grain size and the morphology of second-phase particles significantly influence the growth rate of the oxide film and the surface gloss after treatment [1,2]. By revealing microstructural features such as grain boundaries and second-phase distributions through etching, metallic inspection enables qualitative or semi-quantitative analysis of crucial parameters like grain size and particle morphology. This, in turn, allows for a qualitative assessment of anodizing outcomes. However, since a reliable quantitative relationship between metallographic analysis and anodizing has yet to be established, most manufacturers still primarily rely on expert judgment for metallographic analysis. This evaluation process is prone to human factors such as fatigue and subjective bias, which can lead to distorted results [3,4].

In recent years, the rapid development of computer vision and machine-learning technology has provided technical support for efficient analysis of metallographic structure information of complex metal materials [4]. Naik et al. [5] identified different metallurgical phases in heat-treated steel and extracted the texture features and pixel intensity of metallurgical phases. Dali C et al. [6] developed a semi-supervised learning framework to efficiently identify second-phase components in aluminum alloy metallography. Rusanovsk et al. [7] employed deep semantic segmentation techniques to segment and repair impurities, as well as to identify and quantify grain boundaries in metallographic images. Katika H et al. [8] used morphological manipulation and threshold processing techniques to achieve fine segmentation of microstructure. Majumdar S et al. [9] used the improved U-Net architecture to accurately segment different microstructure features in metal images. Germain et al. [10] used ImageJ 1.53e for image threshold segmentation and extracted the roundness, aspect ratio, and other features of graphite from graphite particles. Wen et al. [11] used the TransUnet to segment metallographic images and extract hole and grain features from cable melt mark images. At present, most metallographic rating methods are mainly completed from three aspects: grain size, microstructure and nonmetallic inclusion, and decarburization layer and nitriding layer [12]. Wang Sen et al. [13] adopted the method of metallographic grain size grading based on deep learning to improve the efficiency and accuracy of the rating. Song Yue et al. [14] automatically calculated the number of grain size levels by deep learning method and realized automatic grain size rating. However, Su Chen [15] used the deep learning framework to segment the residual austenite and martensite microstructures and realized the grade determination of the residual austenite and martensite.

The aforementioned research methods are effective in extracting the grains and morphologies of the second phase in the as-cast state; however, their research dimensions remain relatively limited. Given that the anodization effectiveness of aluminum alloys is influenced by multiple factors with intertwined interactions, accurately identifying and selecting feature values closely related to the anodization process has become the primary focus of our study. Unlike the as-cast state, the second-phase microstructures of aluminum alloy profiles, after processes such as deformation, solution treatment, aging heat treatment, etc., often lose their typical morphologies at the micron scale. Few studies have focused on such alloy states, necessitating the exploration of diverse models and algorithms to correlate metallographic analysis results with anodization performance. Additionally, the semi-supervised learning frameworks and deep learning frameworks employed in these methods may demand substantial computational resources. Deep semantic segmentation techniques often rely on large volumes of high-quality training data, and data annotation

can be costly. While ImageJ is user-friendly, significant manual intervention and parameter adjustments may be required. Algorithms such as gradient-based boundary segmentation and morphological or thresholding techniques are highly sensitive to parameter settings, requiring meticulous parameter tuning to achieve optimal results.

Because of the difference in the research materials, the morphology of the phase and size are not taken as the main research objects. At present, there is little research on the feature analysis and rating of such metallographic images. In light of this, this paper proposes an innovative solution that integrates computer vision and machine-learning technologies. Leveraging Python 3.8.8 and its third-party libraries, the method extracts multiple statistical features of grains to achieve automatic recognition and grading of metallographic images from the perspective of grain characteristics.

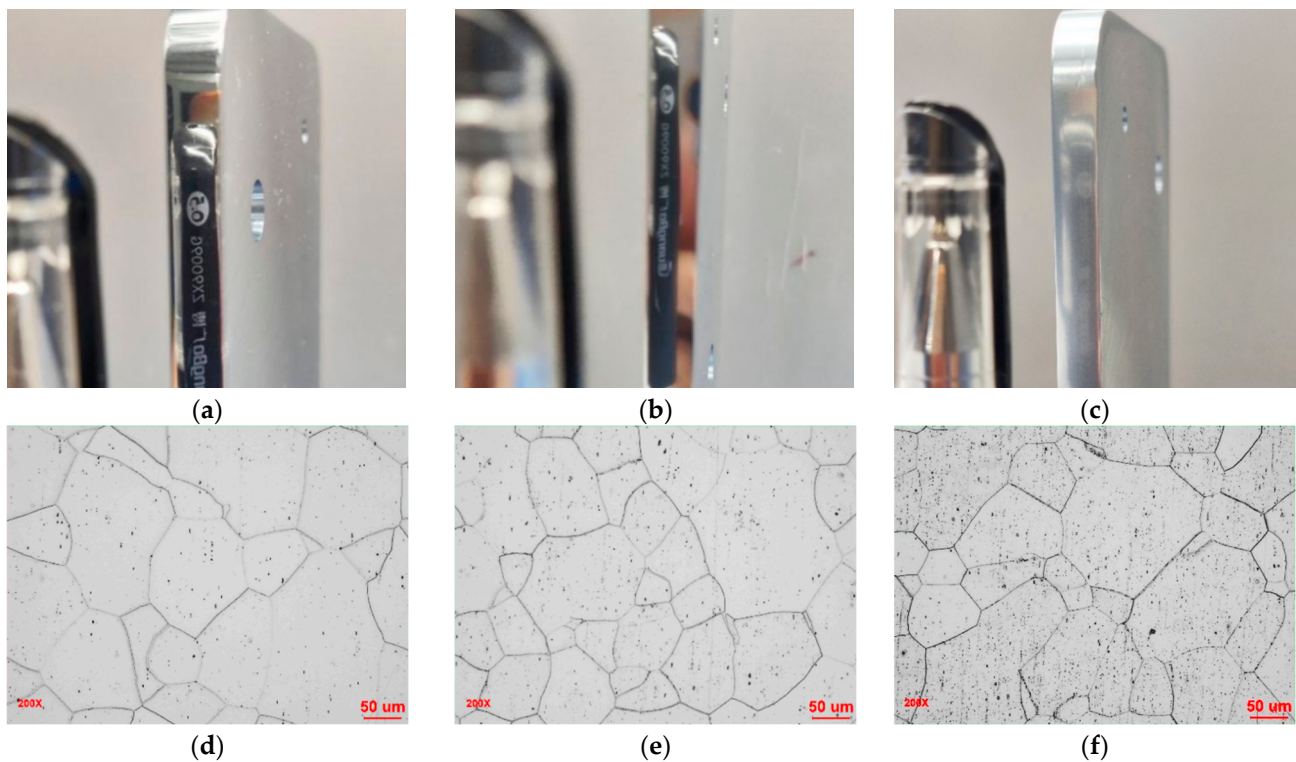
The objective of our research is to develop an automatic metallographic image grading system. We anticipate that this system will be more efficient and reliable than manual grading for quality assessment of 3C aluminum alloy profiles, achieving a metallographic grading accuracy exceeding 90%, with image grading time controlled within 10 s to enable high-quality automated detection. Further, the program package and deployment are completed, and the metallographic rating system is developed, which lays a solid foundation for efficient and large-scale metallographic analysis and quality control. It is anticipated that the metallographic grading system will reduce labor costs.

## 2. Materials and Methods

### 2.1. Metallographic Sample Making and Metallographic Analysis

Samples with different anodic “mirror” effects were taken from the customer’s production line, such as a–c in Figure 1. These samples were made of EN AW-6013 (EN 573-3:2019+A2:2023) aluminum alloy, and its composition is shown in Table 1. The samples were measured using a gloss meter, as illustrated in Figure 2, yielding distinct gloss values. For the specimens shown in Figure 1a–c, the measured gloss values were 559 GU, 509 GU, and 411 GU, which were classified as premium, qualified, and substandard samples, respectively. We measured the glossiness values of all anodized specimens and classified them into five grades (ranging from A to E) according to the ranked glossiness values, as presented in Table 2. Grade A signifies the optimum anodization quality. Grade B denotes the anodization quality that is acceptable to customers. Grade C represents the anodization quality for which customers need to grant a waiver for acceptance, and those below this grade are deemed non-conforming products. Representative samples from each grade were selected for metallographic preparation, forming a dataset to support subsequent supervised machine-learning tasks.

Metallographic samples were cut from the same position on each sample and ground successively with 320 and 1200 mesh silicon carbide sandpaper. The samples were then polished using a 1  $\mu\text{m}$  polishing disc and silica polishing liquid on an automatic polishing machine under controlled parameters ( $300 \pm 5$  rpm,  $60 \pm 2$  N pressure). After cleaning, they were etched with Keller’s reagent ( $\text{HF}:\text{HCl}:\text{HNO}_3:\text{H}_2\text{O} = 2:3:5:190$  by volume, prepared from analytical-grade solutions with densities 1.15 g/mL, 1.19 g/mL, 1.40 g/mL, and deionized water, respectively) at  $26\text{--}30$  °C for 75–80 s. To ensure the stability of metallographic image quality, all preparation steps were conducted in an ambient temperature-controlled environment ( $26 \pm 1$  °C), with critical parameters verified before each experiment. The sample was re-cleaned, wiped with alcohol, and mounted on a Leica DMI8A metallographic microscope. Select the appropriate field of view and adjust the light intensity to 70% of the maximum output intensity to capture the d-e image in Figure 1. A one-to-one mapping between anodic “mirror” effects and corresponding metallographic images was established to construct the dataset.



**Figure 1.** Different “mirror” anode effect samples and corresponding metallography: (a) Optimal “mirror” effect; (b) Qualified “mirror” effect; (c) Failed “mirror” effect; (d) Optimum sample metallography; (e) Qualified sample metallography; (f) Failed sample metallography.

**Table 1.** Standard chemical composition of alloy EN AW-6013, mass%.

Si	Mg	Fe	Cu	Mn	Cr	Zn	Ti	Al
0.50~0.70	0.80~1.20	<0.50	0.50~0.70	<0.80	<0.005	<0.10	<0.25	Bal.



**Figure 2.** Gloss meter.

**Table 2.** Anodized Specimen Evaluation.

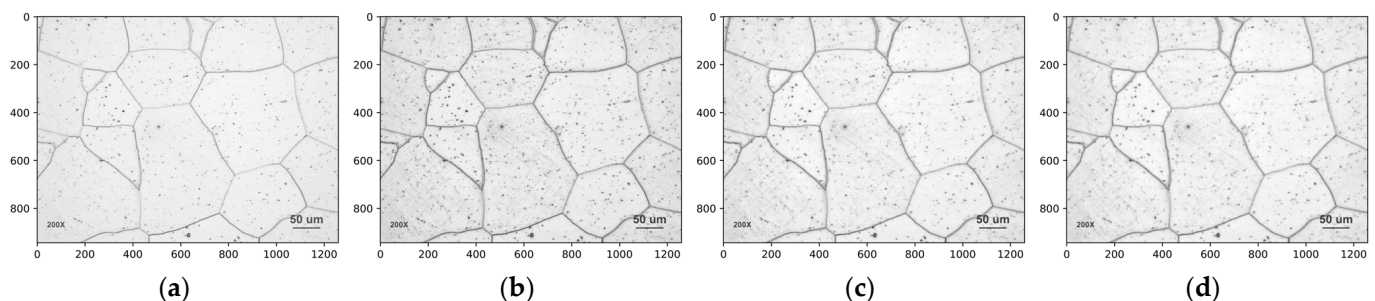
y-Gloss Value (GU)	Sample Grade
$y \geq 530$	A
$500 < y \leq 530$	B
$470 < y \leq 500$	C
$440 < y \leq 470$	D
$y \leq 440$	E



As shown in Figure 1, the metallographic photos of samples with different anode effects are obviously different. In the metallography of qualified samples, there are many grains with high brightness (corrosion-resistant) and low second-phase content. We define these as “high-quality grains”, while the metallography of unqualified samples shows the opposite. Therefore, it is a reasonable assumption that the higher the proportion of high-quality grains, the better the anodizing effect of the sample. Hence, the study takes a single grain as the basic unit to identify high-quality grains and calculate the proportion of high-quality grains.

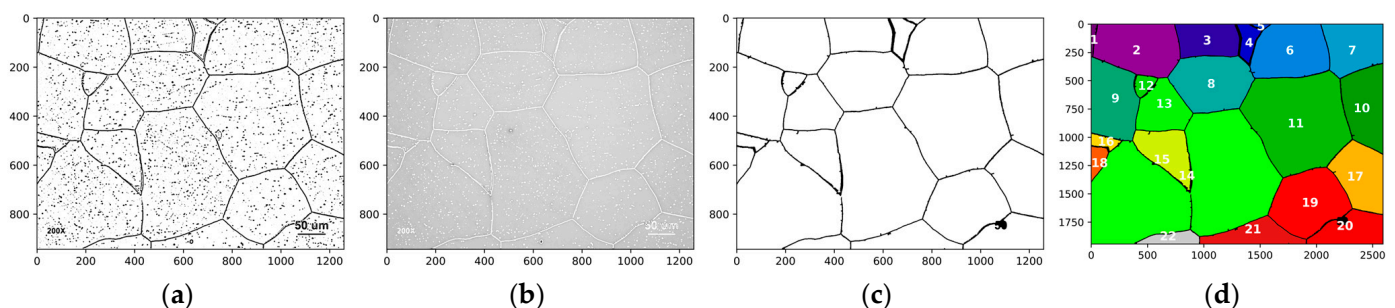
## 2.2. Picture Processing

Python 3.8.8 is used to read the grayscale image, as shown in Figure 3a. To eliminate the influence of uneven illumination during the camera’s capture of the metallographic images, the grayscale image is processed using contrast-limited adaptive histogram equalization (CLAHE) [16], as shown in Figure 3b. In order to remove image noise and effectively retain clarity and detail of grain boundaries, median filtering (Figure 3c) and bilateral filtering (Figure 3d) are used to process images [17].



**Figure 3.** Image preprocessing: (a) original image; (b) after treatment by CLAHE; (c) median filtering; (d) bilateral filtering.

The preprocessed image is binarized using the adaptive thresholding method, with two different parameters set to obtain two types of second-phase distributions. The first type represents the impurity distribution, as shown in Figure 4a, while the background image is obtained by subtracting the impurity distribution from the original image, as shown in Figure 4b. After performing a series of morphological operations and connected domain analysis on the second type of second-phase distribution image, the grain boundaries are extracted, as shown in Figure 4c, and the grains are calibrated, as shown in Figure 4d. In Figure 4d, different colors represent different grains.



**Figure 4.** Grain division: (a) Distribution of the second phase; (b) Background image; (c) Grain boundary extraction; (d) Grain calibration.

### 2.3. Feature Extraction

Due to the difference in the materials studied, the morphology of the secondary phase and size are not taken as the main research objects. Instead, we are more inclined to take a series of statistical values related to the grain and gray value as the representative by evaluating the advantages and disadvantages of the local area and then calculating the proportion, the overall metallographic evaluation. In order to better distinguish the good and bad grains, several features are selected to offset the subjective influence brought by naked-eye observation. Combined with material science knowledge and computer vision knowledge, 17 features of grain, as shown in Table 3, are listed, and the features are selected.

**Table 3.** Feature extraction.

Image	Feature	Meaning
original	Brightness	Average grain gray value
	Brightness_Dev	Gray value difference
	Entropy_Original	The richness of grain gray value
	Maximum	Maximum gray value
	Minimum	Minimum gray value
	Median	The median of the gray value
	Mode	The most gray values in the grain
	Mode_proportion	The proportion of the Mode in the grain
	Over_180_rate	The proportion of gray value greater than 180 in the grain
	Kurtosis	The sharpness of the gray value distribution
	Skewness	The pattern of gray value distribution
	area	The sum of intra-grain pixels
	perimeter	The sum of grain boundary pixels
Impurity	Impurities_Percentage	The amount of impurities in the grain
	Entropy_Impurity	The richness of grain gray value in impurity image
Background	Grayscale_Ave_Back	Average grain gray value in background image
	Grayscale_Dev_Back	Gray value difference in background image

#### 2.3.1. Feature Calculation

For the original image, 13 features are extracted. Brightness is the average gray value of the image, and Brightness\_Dev can measure the difference in the degree of the gray level of the image. Brightness and Brightness\_Dev are calculated as follows:

$$\text{Brightness} = \frac{\sum_{i=1}^N x_i}{N} \quad (1)$$

$$\text{Brightness\_Dev} = \sqrt{\frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}} \quad (2)$$

where  $N$  represents the number of grayscale values in the grain, and  $x_i$  is the  $i$ -th grayscale value.

The Maximum is the largest grayscale value in the grain, and the Minimum is the same. The grayscale value of the grain in sequence in the middle is the Median. Mode, Mode\_proportion, and Over\_180\_rate have a certain effect on the overall brightness of the image. The formula for calculating Mode\_proportion and Over\_180\_rate is as follows:

$$\text{Mode\_proportion} = \frac{X_{mode}}{N} \quad (3)$$

$$\text{Over\_180\_rate} = \frac{X_{180}}{N} \quad (4)$$

where,  $X_{mode}$  represents the number of modes,  $X_{180}$  represents the number of grayscale values greater than 180, and  $N$  represents the number of grayscale values in the grain.

The Kurtosis of grayscale histogram can evaluate the concentration of grayscale inside the grain, and the skewness can reflect the asymmetry of grayscale distribution. Skewness and Kurtosis are calculated as follows:

$$\text{Skewness} = \frac{1}{n} \sum_{i=0}^{255} \left[ \left( \frac{X_i - \mu}{\sigma} \right)^3 \right] \quad (5)$$

$$\text{Kurtosis} = \frac{1}{n} \sum_{i=0}^{255} \left[ \left( \frac{X_i - \mu}{\sigma} \right)^4 \right] - 3 \quad (6)$$

where  $n$  is 256, representing the number of different grayscale values;  $X_i$  is the frequency of occurrence of the  $i$ -th grayscale value in the  $[0, 255]$  gray range;  $\mu$  is the mean value of frequency, and  $\sigma$  is the standard deviation of frequency.

The greater the entropy of grain grayscale, the more complex the grain structure or the more small changes. Entropy is calculated by the following formula:

$$\text{Entropy\_Original} = - \sum_{i=0}^{255} p(x_i) \log p(x_i) \quad (7)$$

where,  $x_i$  is the  $i$ -th grayscale value,  $p(x_i)$  is the frequency of occurrence of the  $i$ -th grayscale value.

For the impurity image, the Impurities\_Percentage and the Entropy\_Impurity are calculated. The Impurities\_Percentage directly reflects the impurity content of the grain, and the lower Impurities\_Percentage means that the grain is purer. Impurities\_Percentage and Entropy\_Impurity are calculated as follows:

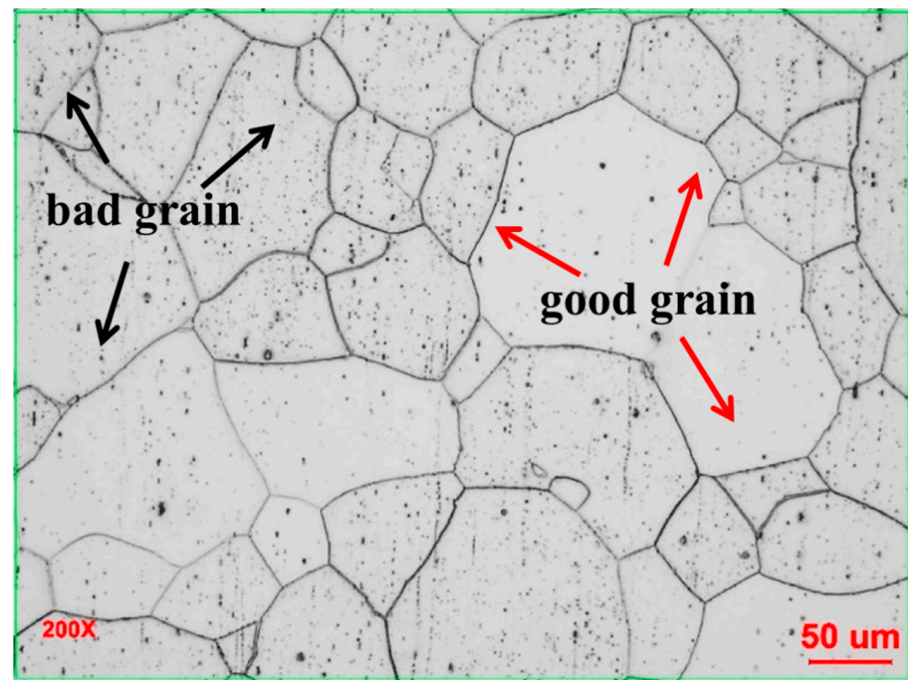
$$\text{Impurities\_Percentage} = \frac{X_0}{N} \quad (8)$$

$$\text{Entropy\_Impurities} = -p(x_0) \log p(x_0) - p(x_{255}) \log p(x_{255}) \quad (9)$$

where  $N$  represents the number of grayscale values in the grain, and  $x_0$  represents the number of grayscale values 0 in the grain. Different from Entropy\_Original, the Entropy\_Impurities calculates only 0 and 255.

For the background image, the Grayscale\_Ave\_Back and Grayscale\_Dev\_Back are calculated to analyze the grain background color.

Through the above steps, a grain dataset containing 17 features is obtained. Experts empirically defined grains with higher brightness and fewer second-phase particles as "good" grains, labeled as 1 (as shown by the red arrows in Figure 5), while regions with darker brightness and more second-phase particles were defined as "bad" grains, labeled as 0 (as shown by the black arrows in Figure 5). This process resulted in a dataset with classification labels.



**Figure 5.** Dataset construction.

### 2.3.2. Feature Selection

Area and perimeter do not need to participate in grain classification. The dataset contains 15 features in addition to the area and perimeter. It is foreseeable that not all features contribute to grain classification, and some redundant features may even hinder model performance. In order to reduce model complexity and improve model generalization ability, feature selection is needed.

The main methods of feature selection include embedding, filtering, and packaging [18]. The embedding is carried out simultaneously in model construction and feature selection. By fitting some tree models (DT, GBDT, XGBoost, RF), the embedding method is used to select features, use the model to rank the importance of features, and the features with high contribution to the model are selected and compared comprehensively, as shown in Table 4. The Entropy\_Impurities, Brightness\_Dev, Skewness, and Impurities\_Percentage are considered to be more important in most models. At the same time, it can also be clearly seen that the three models—RF, GBDT, and DT—think that the contribution of Impurities\_Percentage is the largest. Only XGBoost thinks that the contribution of Impurities\_Percentage is the smallest and that the Entropy\_Impurities contribution is the largest. This may be because the Entropy\_Impurities has a large linear correlation with the Impurities\_Percentage; hence, other methods are needed to select features. We rank the contribution of the features to the four models and remove the features with small contributions, such as the Median, Mode, and Brightness, which may inhibit the performance of the model.

Secondly, the variance method can be used to filter the remaining features. In order to unify the dimensions, all features are normalized, and then the variance of each feature is calculated. The results are shown in Table 5. Features with small variances, such as Grayscale\_Ave\_Back, can be eliminated because of their small fluctuations, and the effect of good and bad grain differentiation is not obvious.

Finally, a correlation analysis was conducted on each feature, and the heatmap of correlation coefficients is shown in Figure 6. The correlation between the features selected by the embedding method and the variance method was analyzed. In the heatmap, the darker or lighter the color, the stronger the negative or positive correlation between the



features. As shown in Figure 6, for example, the correlation between the entropy of the binary image and the proportion of impurities reaches 0.98, indicating a strong positive correlation. Features with high correlation describe different aspects of the same thing. To reduce model complexity and eliminate redundant features, one of the two highly correlated features is removed.

**Table 4.** Feature Importance Ranking.

Feature	RF	XGBoost	GBDT	DT	Average Sorting
Entropy_Impurities	2	1	2	4	2.25
Brightness_Dev	5	3	3	2	3.25
Skewness	3	2	4	5	3.5
Impurities_Percentage	1	15	1	1	4.5
Mode_proportion	6	4	6	3	4.75
Grayscale_Dev_Back	8	5	5	6	6
Kurtosis	4	7	7	7	6.25
Entropy_Original	7	10	8	15	10
Maximum	10	9	11	11	10.25
Over_180_rate	9	14	10	8	10.25
Minimum	11	13	9	10	10.75
Grayscale_Ave_Back	13	11	12	9	11.25
Brightness	12	6	14	13	11.25
Mode	15	8	15	12	12.5
Median	14	12	13	14	13.25

**Table 5.** Features and Variance.

Features	Variance
Brightness	0.0699
Median	0.0832
Grayscale_Ave_Back	0.0909
Mode	0.1050
Grayscale_Dev_Back	0.1258
Minimum	0.1261
Brightness_Dev	0.1579
Maximum	0.1700
Skewness	0.1852
Kurtosis	0.1863
Impurities_Percentage	0.1882
Entropy_Impurities	0.1929
Mode_proportion	0.1931
Entropy_Original	0.1937
Over_180_rate	0.2935

The 15 features were selected by embedding method, variance method, and correlation coefficient method, and the final features were Impurities\_Percentage, Brightness\_Dev, Mode, Kurtosis, and Over\_180\_rate.

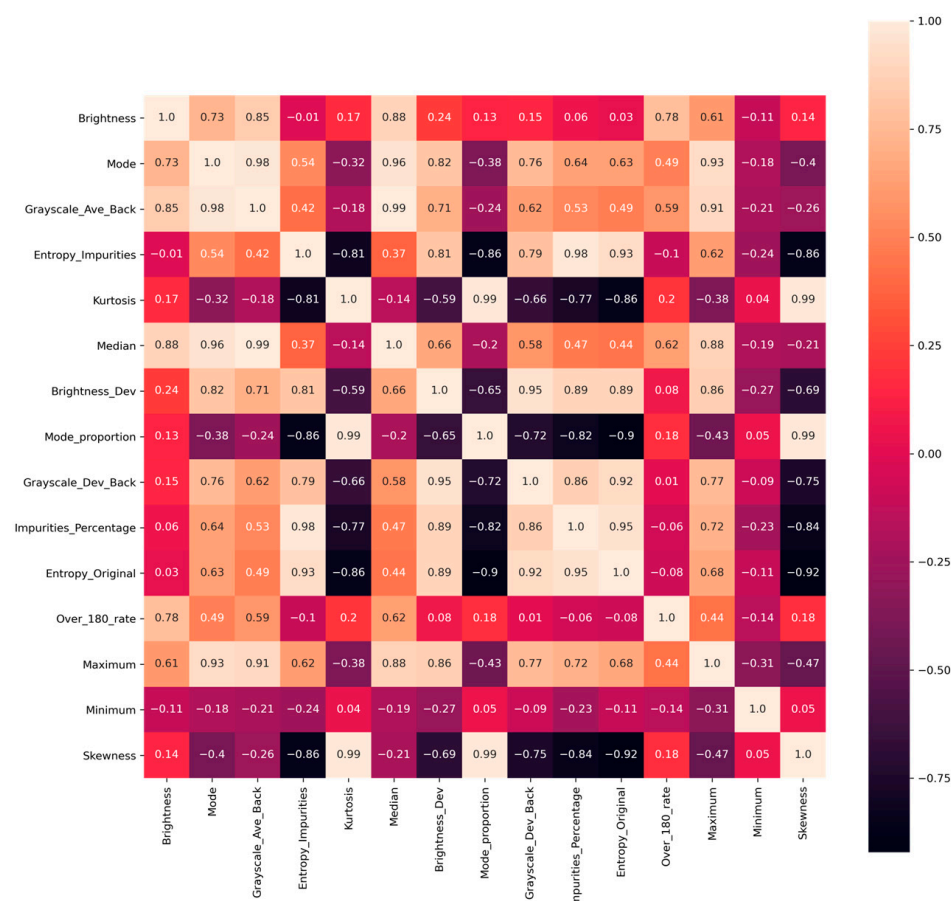


Figure 6. Correlation coefficient heat map.

## 2.4. Model Building

Grain classification is needed, and grain datasets and machine-learning classification algorithms can achieve this purpose. For a long time, machine-learning classification algorithms have developed rapidly in various fields. Çetin Necati et al. used four machine-learning classification algorithms (RF, SVM, NB, and MLP) to classify soybean varieties based on the shape, size, and quality attributes of soybean seeds [19]. By integrating multiple machine-learning classification models, Mahajan M et al. achieved remarkable results in ECG signal classification [20].

The grain dataset is split into a training set and a test set in an 80:20 ratio. The machine-learning classification model is trained on the training set, and its performance is evaluated on the test set. To improve the model, defects in the dataset are addressed, several classification models are fitted, and appropriate evaluation metrics are selected to assess the models' performance.

### 2.4.1. Algorithm Description

#### Decision Tree

The Decision Tree (DT) is composed of a root node, several internal nodes, and leaf nodes, which correspond to the grain features calculated previously. The root node and internal nodes are divided by impurity measures, such as entropy and the Gini index. Every time the tree grows, nodes are divided in the direction of minimum impurity [21]. We selected the Gini index as the impurity measure, imposed no restrictions on the maximum tree depth, set no limits on the maximum number of leaf nodes, and assigned equal weights to all classes.

## Ensemble Learning

(1) Random Forest (RF): The Random Forest consists of Decision Trees, and its randomness is reflected in each Decision Tree built. The samples used by these Decision Trees are randomly selected from the grain training set, and the final prediction results are determined by the Decision Trees through voting [22]. To facilitate comparison, we trained Random Forest models with 50, 100, and 500 trees, uniformly selecting the Gini index as the impurity measure.

(2) Gradient Boosting Decision Tree (GBDT): Unlike the Bagging approach used in Random Forests (RF), each iteration of Boosting increases the weights of misclassified grain samples based on the results from the previous iteration. Each new model built by the GBDT algorithm attempts to correct the errors of the previous model [23]. The loss function employs log\_loss (logarithmic loss) for probabilistic classification, with a learning rate of 0.1, 100 trees, a maximum depth of 3 for individual trees, and early stopping disabled.

(3) eXtreme Gradient Boosting (XGBoost): XGBoost, as an enhanced version of the GBDT (Gradient Boosting Decision Tree) algorithm, is designed to prevent model overfitting through the application of loss functions and regularization. The logistic loss function is selected as the model's loss criterion, which not only calculates the base loss but also effectively mitigates overfitting through the integration of first-order and second-order derivatives [24]. Logistic regression regularization is implemented to control model complexity further. Specifically, L1 regularization is characterized by penalties proportional to the absolute values of model weights, which is commonly employed to precisely zero out partial weights for feature selection, while L2 regularization is defined by penalties proportional to the square of model weights, enabling all features to be preserved. Since feature screening has already been conducted, preference is given to the L2 regularization approach. In the regularization framework, larger weights are penalized more severely. To avoid excessive model complexity, the weight coefficient for the L2 regularization term is set to 1, ensuring smaller weights are maintained. A learning rate of 0.3 is configured, where the adjustment magnitude during model weight updates is controlled through smaller iterative steps, enabling slower and more stable convergence to reduce overfitting risks. A total of 100 decision trees are utilized, and uniform sample weights are assigned to each class.

(4) Stacking: Different from the ensemble of tree models, Stacking can integrate different kinds of models. By integrating predictions from multiple models for grain prediction, a logistic regression model is then constructed as a meta-model using these predictions. The output of the meta-model serves as the final prediction result for grains [25]. We employed two stacked ensembles:

① A three-model ensemble comprising RF ( $n_{\text{estimators}} = 50$ ), XGBoost, and LR (L2 regularization, solver = 'lbfgs').

② A five-model ensemble including RF ( $n_{\text{estimators}} = 50$ ), XGBoost, LR (L2 regularization, solver = 'lbfgs'), Multilayer Perceptron (MLP), and SVC (kernel = 'linear').

For both ensembles, the predictions from the base models were used to train a meta-model (logistic regression). The entire process was validated through 5-fold stratified cross-validation.

(5) Voting: Voting is an ensemble learning method that integrates diverse types of models. The final grain prediction is determined by a majority vote of the predictions from these models. We implemented soft voting with two distinct ensembles:

① A three-model ensemble using RF ( $n_{\text{estimators}} = 50$ ), XGBoost, and LR (L2 regularization, solver = 'lbfgs').

② A five-model ensemble incorporating RF ( $n_{\text{estimators}} = 50$ ), XGBoost, LR (L2 regularization, solver = 'lbfgs'), Multilayer Perceptron (MLP), and SVC (kernel = 'linear').

By averaging the predicted probabilities from these models, the final grain prediction becomes more robust and probability-driven.

### Logistic Regression

Logistic Regression (LR) uses the sigmoid function to map the output of linear regression between 0 and 1, essentially predicting the probability of the class [26]. We employed logistic regression to predict the probability of grain classes, where a grain is assigned to the class with the highest predicted probability. To improve classification performance, we explored different regularization methods, including L1 (Lasso) and L2 (Ridge) regularization. The following loss function is formulated:

$$L = E(\omega) + \lambda ||\omega||^q \quad (10)$$

The original linear loss function is defined as  $E(\omega)$ , where  $\omega$  denotes the weights obtained through training. The regularization term is represented by  $\lambda ||\omega||^q$ , with  $\lambda$  being the regularization coefficient and  $q$  being the model's order. When  $q$  is set to 1, first-order regularization is implemented, and when  $q$  is set to 2, second-order regularization is established. The difference between L1 and L2 is mainly in the penalty term. L2 imposes a square-level penalty on features, which can retain all features, while L1 can reduce the weight of some features to 0, which can realize feature selection. When the regularization coefficient  $\lambda$  is set to a larger value, the constraints imposed on the parameters are strengthened. During the training process, excessively large parameter values are automatically suppressed by the regularization term, thereby reducing overfitting. The optimizer selected by LR(L1) is saga, and the optimizer selected by LR(L2) is lbfgs, with the same category weight.

### Naive Bayes

Naive Bayes (NB) uses feature probabilities to predict classification. For an unclassified grain sample, the probabilities of it belonging to each class are computed under the given conditions. The grain is then assigned to the class with the highest predicted probability [27]. Without specifying the prior probability of the class, the prior probability is automatically calculated based on the class distribution in the training data.

### Support Vector Machine

Support Vector Machines (SVM) separate two classes of grain samples by constructing an optimal hyperplane that maximizes the margin (distance) between the two classes in the feature space [28], where the following constraints must be satisfied.

$$\min (C + \frac{1}{2} \sum_{i=1}^n \theta_j^2) \quad (11)$$

In this context,  $C$  is referred to as the regularization parameter, while  $\frac{1}{2} \sum_{i=1}^n \theta_j^2$  is defined as the regularization term, where  $\theta$  represents the model parameters. When  $C$  is assigned a larger value, the training error is reduced; however, overfitting may be induced. Conversely, when  $C$  is assigned a smaller value, the regularization term's role is enhanced, thereby improving the model's generalization capability. The regularization parameter  $C$  is set to 1, the Gaussian (rbf) kernel is selected for non-linearly separable data, the linear kernel is used for linearly separable cases, and equal class weights are applied across all categories.

## K-Nearest Neighbor

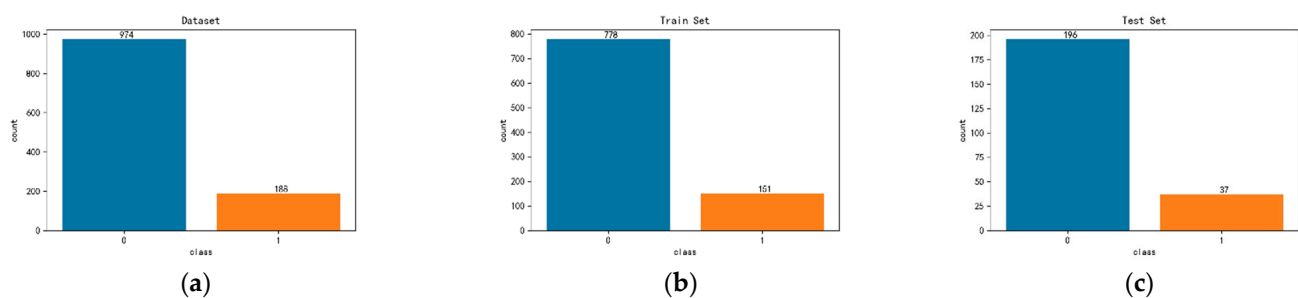
The K-Nearest Neighbors (KNN) algorithm measures the proximity between grain samples using Euclidean distance. It selects the K closest grain samples to the unclassified grain and predicts its class through majority voting [29]. Five and seven sample points were selected, respectively, to compare the predicted results. All neighbor samples have the same voting weight.

## Artificial Neural Network

An Artificial Neural Network (ANN) consists of an input layer, one or more hidden layers, and an output layer. Each hidden layer is composed of fully connected nodes. Grain feature information is propagated from the input layer through the hidden layers to the output layer, with activation functions applied at each layer. Finally, the output layer uses a sigmoid activation function to generate probabilities for grain classification [30]. The neural network architecture uses a single hidden layer with 100 neurons, ReLU activation for the hidden layer, and the Adam optimizer. The Adam optimizer is configured with exponential decay rates of 0.9 (first moment) and 0.999 (second moment), L2 regularization ( $\alpha = 0.0001$ ), a fixed learning rate of 0.001, and early stopping is disabled.

### 2.4.2. Model Evaluation Metrics

The number of grain samples of the dataset labeled by experts and the divided training set and test set are shown in Figure 7.



**Figure 7.** Dataset sample situation: (a) Dataset; (b) Training set; (c) Test set.

It is evident that the two sample types are not in a 1:1 ratio. Specifically, the ratio of good grains to poor grains is approximately 1:5.2, resulting in a class-imbalanced dataset. This imbalance renders methods relying solely on ‘accuracy’ as an evaluation metric ineffective [31,32]. When the ratio of Category 0 to Category 1 samples is 9:1, even if the model predicts all samples as Class 0 samples, the accuracy can reach 90%. However, the model does not have any recognition ability for Class 1 samples. Therefore, the accuracy cannot be used only as the evaluation metric of the model.

For imbalanced datasets, in addition to the commonly used accuracy metrics, recall is also used as the evaluation metric of the model [33]. When more good grain samples were correctly predicted, the good grain recall was higher. Our rating method is based on the area of good grains, which requires prioritizing the identification of good grains. Simultaneously, we aim to maximize the correct classification of poor grains as poor. Therefore, we calculate the recall rate for poor grains—the higher the recall, the poorer grains are accurately predicted. In order to further judge the prediction ability of the model under non-equilibrium conditions, the AUC (Area Under the ROC Curve) evaluation metrics were also used. AUC can simultaneously test the classification ability of the classifier for both bad-grain and good-grain samples. While good grains are judged as good grains, it can also test whether the model misjudges bad grains as good grains less.



The calculation methods of accuracy, recall (good grain recall rate is recall1, bad grain recall rate is recall0), and AUC are described in Table 6.

**Table 6.** Confusion matrix to calculate the evaluation metrics of the model.

True value	Predicted Value	
	Positive	Negative
Positive	TP (11)	FN (10)
Negative	FP (01)	TN (00)

TP (True Positive): The number of grains predicted by the model to be good grains that are actually good grains. It is represented by 11. FN (False Negative): The number of grains predicted by the model to be bad grains that are actually good grains. It is represented by 10. FP (False Positive): The number of grains predicted by the model to be good grains that are actually bad grains. It is represented by 01. TN (True Negative): The number of grains predicted by the model to be bad grains that are actually bad grains. It is represented by 00.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

$$recall1 = \frac{TP}{TP + FN} \quad (13)$$

$$recall0 = \frac{TN}{FP + TN} \quad (14)$$

$$TPR = \frac{TP}{TP + FN} \quad (15)$$

$$FPR = 1 - \frac{TN}{FP + TN} = \frac{FP}{FP + TN} \quad (16)$$

The vertical coordinate of the ROC curve is the True Positive Rate (TPR), and the horizontal coordinate is the False Positive Rate (FPR). The curve takes values of different classification thresholds successively to obtain multiple groups of TPR and FPR, which are drawn successively in the image, that is, the ROC curve. The closer the ROC curve is to the top left corner (0,1), the better the model performs. By calculating the area under the ROC curve (AUC), the performance of the model ROC curve is measured in a more comparable numerical way.

#### 2.4.3. Dataset's Problems and Processing Methods

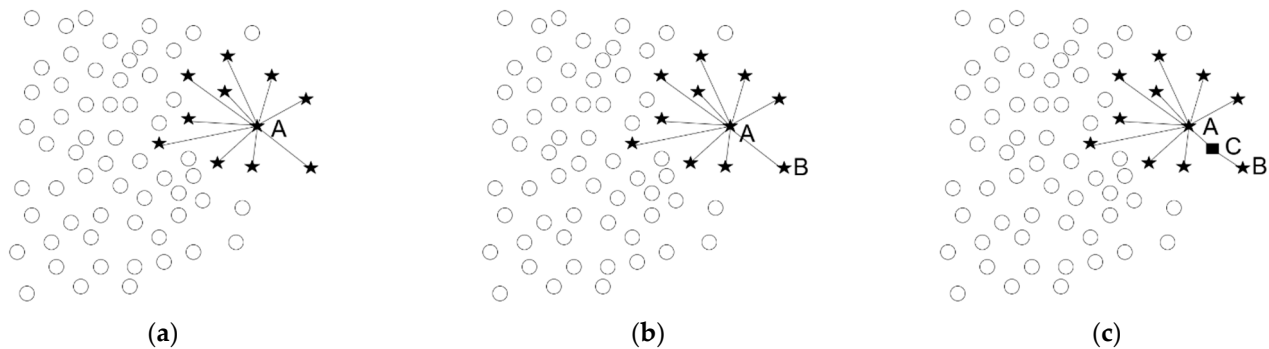
The quality of the dataset is directly related to the quality of the established model. When the dataset is small, models struggle to learn generalizable patterns from the data, leading to suboptimal performance on new data and weaker generalization. Models trained on imbalanced datasets may exhibit bias toward the majority class, performing better on the majority class while failing to learn sufficient features from the minority class samples. This results in poor generalization capability for the minority class. Additionally, commonly used evaluation metrics like accuracy may produce misleading results. If the proportion difference between the two types of samples in the grain dataset is too large and the number of samples is too small, the following methods are used.

#### SMOTE

Currently, there is a significant imbalance in the quantity of the two types of samples in the dataset. Due to the insufficient sample size, we should not balance the dataset by undersampling the majority class. Instead, we should use oversampling the minority class to increase the number of training samples. If the expert were to re-label the grain samples, it would require considerably more time. In order to optimize the model and reduce the influence of an imbalanced dataset, the Synthetic Minority Oversampling Technique

(SMOTE) was used to increase the number of good grains [34]. SMOTE's idea is to have a line between a minority class sample and its  $K$  neighbor samples and, between the lines, create a new sample of the same class as the sample in which the line was placed [35]. The specific process is the following:

1. For each Class 1 sample  $A$ , calculate the Euclidean distance from  $A$  to all other Class 1 samples (neighbors) (Figure 8a).



**Figure 8.** SMOTE oversampling diagram: (a) Class 1 sample  $A$ ; (b)  $A$ 's neighbors  $B$ ; (c) New sample  $C$ .

2. According to the imbalanced proportion of samples, the sampling rate is set. For each Class 1 sample  $A$ , neighboring Class 1 sample  $B$  is randomly selected from its neighbors (Figure 8b).

3. For each neighbor  $B$ , according to Formula (17) and the original minority class sample, build a new Class 1 sample  $C$  (Figure 8c).

$$C = A + \text{rand}(0,1) \times d_{AB} \quad (17)$$

where  $d_{AB}$  is the Euclidean distance between Class 1 samples  $A$  and  $B$ , and  $\text{rand}(0,1)$  indicates that the value is randomly taken between 0 and 1.

Use the Python 3.8.8 third-party library SMOTE to sample the training set, and do not process the test set. The sample number of good and bad grains in the training set after SMOTE sampling is shown in Table 7.

**Table 7.** Sample status after and before SMOTE sampling.

	Number of Class 0 Samples (PCS)	Number of Class 1 Samples (PCS)
Original training set	778	151
SMOTE	778	778

### 10-Fold Cross-Validation

In addition, it is easy to find that the sample size of the dataset is relatively small, and the trained model may not fully learn the features of the dataset. Therefore, the 10-fold cross-validation method is adopted. This method can make full use of the existing dataset and divide the training set into ten parts on average. Each time the model is fitted, nine parts of the data are used, and the remaining part is used for model verification to fully train the model, improve the generalization ability of the model, and reduce the contingency [36].

For the Random Forest model RF ( $n = 50$ ) with 50 trees, we first use subsets 1–9 as the training set and subset 10 as the validation set to train and evaluate the model. Next, subsets 1–8 and 10 are used for training, with subset 9 as the validation set. This process is repeated until every subset has once served as the validation set. Finally, we compute the model's average performance metrics (e.g., accuracy, recall). The same cross-validation

method is applied to other models to calculate their average metrics. By comparing these averaged metrics across models, the optimal model is selected.

The model's performance is ensured through 10-fold cross-validation, which involves multiple rounds of training and validation on different subsets of the training data. This methodology enables a more comprehensive and robust evaluation of model performance by minimizing dependency on specific data partitions. Stability is quantified using the standard deviation, where a smaller deviation indicates that consistent predictive performance is maintained across diverse data subsets with reduced sensitivity to data fluctuations. Consequently, enhanced generalization capability and improved robustness are achieved, as the model is guided to learn universal patterns within the data rather than overfitting to idiosyncratic noise.

### 3. Results

After the above aluminum alloy metallographic image processing is used to divide the grain, modeling is used to judge the good and bad grain and can calculate the proportion of good grain area. The following are some experimental results that are shown and explained.

#### 3.1. Grain Classification Result

##### 3.1.1. Raw Dataset

As shown in Table 8, several classification models were fitted to classify grains. Accuracy, recall, AUC, and ROC curves were used to judge the performance of the model on the test set. From the perspective of accuracy and recall, most classifiers can achieve an accuracy of 90%, while the recall of most classifiers is about 80%. From the perspective of the ROC curve and AUC, some classifiers have a weak ability to identify good grains.

**Table 8.** Performance of different models.

Classifier	Accuracy	Good Grain Recall	Bad Grain Recall	AUC
RF1 (n = 50)	0.9442	0.8919	0.9541	0.9854
RF2 (n = 100)	0.9356	0.8649	0.9490	0.9868
RF3 (n = 500)	0.9313	0.8378	0.9490	0.9864
XGBoost (n = 100)	0.9356	0.8378	0.9541	0.9814
LR1 (kernel = L2)	0.9528	0.8649	0.9694	0.9891
LR2 (kernel = L1)	0.9399	0.6757	0.9898	0.9690
NB	0.7296	0.9459	0.6888	0.9210
DT	0.9185	0.7297	0.9541	0.8419
SVC1 (kernel = linear)	0.9485	0.8378	0.9694	0.9885
SVC2 (kernel = rbf)	0.8412	0.0000	1.0000	0.9459
GBDT	0.9399	0.8378	0.9592	0.9819
ANN (MLP)	0.9056	1.0000	0.8876	0.9891
KNN1 (K = 5)	0.8927	0.6216	0.9439	0.9002
KNN2 (K = 7)	0.8884	0.5405	0.9541	0.9013
Stacking (RF1, XGB, LR1)	0.9485	0.8108	0.9745	0.9866
Voting (RF1, XGB, LR1)	0.9442	0.8378	0.9643	0.9857
Stacking (RF1, XGB, LR1, MLP, SVC1)	0.9442	0.8108	0.9694	0.9872
Voting (RF1, XGB, LR1, MLP, SVC1)	0.9442	0.8378	0.9643	0.9869

The recognition effect of the model fitted with an imbalanced dataset on good grains can be reflected by the recall, which refers to the ratio of the number of samples correctly predicted as good grains to the actual number of good grains. The higher the recall rate for good grains, the more good grains are correctly predicted.

ROC curves of different models are shown in Figure 9.

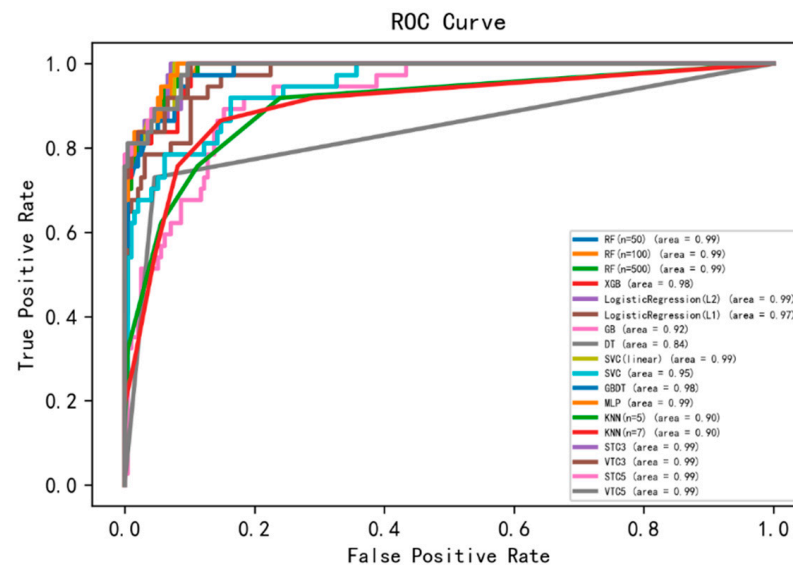


Figure 9. ROC curves of different models.

In Table 8, a specific subset of the data reveals an unusual pattern: the SVC(rbf) model achieves an accuracy of 84.12%, but its recall rate for good and bad grains is 0 and 1, respectively. This indicates that the model predicts all samples in the test set as poor grains, demonstrating zero ability to identify good grains. In this case, however, its accuracy is still as high as 84.12%. Therefore, in the imbalanced dataset, accuracy cannot be used as the only model evaluation metric, and recall and AUC should also be considered. Notably, the SVC model with a linear kernel achieves a recall rate of 0.8378 for good grains and 0.9694 for poor grains, demonstrating significantly better performance compared to the SVC (rbf) model. The Gaussian (rbf) kernel is designed for non-linearly separable data, while the linear kernel is suited for linearly separable data. This divergence in their performance on the same dataset suggests that the underlying data patterns may align more closely with linear separability.

As outlined in Section 2.4.1, we configured all class weights to be equal for the Decision Tree (DT) and Logistic Regression (LR) models and assigned uniform voting weights to neighboring samples in K-Nearest Neighbors (KNN). However, this uniform weighting approach may negatively impact performance on imbalanced datasets. Table 8 shows that the recall rates for good grains are suboptimal across these models: 0.7297 (DT), 0.5405 (KNN with  $K = 7$ ), 0.6216 (KNN with  $K = 5$ ), and 0.6757 (LR with L1 regularization). In contrast, the LR model with L2 regularization achieves a significantly higher recall rate of 0.8649 for good grains. We hypothesize that this discrepancy stems from differences in optimizers: the L1-regularized LR uses the saga optimizer, which is designed for large datasets, whereas the L2-regularized LR employs the lbfgs optimizer, optimized for small-to-medium datasets. Since our dataset falls into the latter category, lbfgs may be more appropriate. Notably diverging from other models, the Naive Bayes (NB) classifier attains a recall rate of 0.9459 for good grains and 0.6888 for poor grains. This superior performance likely arises because NB calculates prior probabilities directly from the class distribution in the training data, thereby enhancing its ability to identify good grains.

The remaining models generally achieve accuracies above 0.9, with recall rates for good grains exceeding 0.8, recall rates for poor grains surpassing 0.85, and AUC scores above 0.9. The Random Forest (RF) model delivers the best performance when the number of trees is set to 50, achieving an accuracy of 0.9442, a recall rate of 0.8919 for good grains, 0.9541 for poor grains, and an AUC of 0.9854.

### 3.1.2. SMOTE-Sampled Dataset

Compared with before and after SMOTE sampling, it can be seen that the recall has significantly increased, and the model after SMOTE has a stronger ability to identify good grain. Although most models fitted after SMOTE exhibit slightly reduced accuracy, the recall rates for good grains and AUC are significantly improved. This demonstrates that model performance after SMOTE sampling is superior to pre-sampling performance. It can also be seen from the ROC curves of Figures 9 and 10 that the ROC curve of the model after SMOTE sampling was closer to the upper left corner, and the recognition ability of good grain was significantly improved.

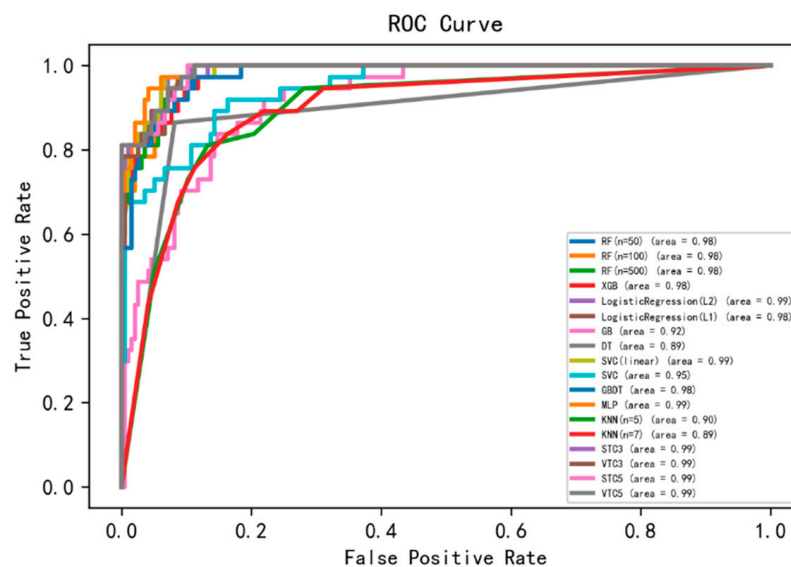


Figure 10. ROC curve of the model after SMOTE sampling.

As shown in Table 9, without altering hyperparameters and solely applying SMOTE to balance the dataset, we observed notable improvements in recall rates for good grains compared to the non-sampled case across DT, KNN, LR (L1), and SVC (rbf), with values largely exceeding 0.8. In contrast, the Naive Bayes (NB) model exhibited a reduced recall rate for poor grains post-SMOTE. This degradation likely stems from NB's inherent assumption of feature independence, which makes it highly sensitive to data distribution. SMOTE sampling may disrupt the original distribution, violating this assumption and thus impairing performance. The Artificial Neural Network (ANN) achieved a recall rate of 0.8929 for poor grains after SMOTE, outperforming the non-sampled scenario. Notably, its recall rate for good grains remained at 1.0 under both sampling conditions, indicating robust capture of Class 1 features. However, this perfection may signal overfitting. The increased sample diversity introduced by SMOTE likely enhanced ANN's generalization capability, thereby improving its recall for poor grains.

Voting (RF1, XGB, LR1, MLP, SVC1) is a model obtained by voting on the prediction results of five different types of models. It holistically evaluates predictions across multiple models and emerges as the top performer among the benchmarked approaches, achieving the highest accuracy, AUC, and an elevated recall rate for good grains. The Voting ensemble (RF1, XGB, LR1, MLP, SVC1) achieved an accuracy of 0.9313, a recall rate for good grains of 0.9459, a recall rate for poor grains of 0.9286, and an AUC of 0.9870.

After SMOTE sampling, the ROC curves for different models are as shown in Figure 10.



**Table 9.** Performance of different models after SMOTE sampling.

Classifier	Accuracy	Good Grain Recall	Bad Grain Recall	AUC
RF1 (n = 50)	0.9270	0.9459	0.9235	0.9837
RF2 (n = 100)	0.9270	0.9459	0.9235	0.9833
RF3 (n = 500)	0.9313	0.9459	0.9286	0.9828
XGBoost (n = 100)	0.9227	0.8378	0.9388	0.9800
LR1 (kernel = L2)	0.9270	0.9730	0.9184	0.9868
LR2 (kernel = L1)	0.8798	1.0000	0.8571	0.9850
NB	0.7082	0.9459	0.6633	0.9160
DT	0.9099	0.8649	0.9184	0.8916
SVC1 (kernel = linear)	0.9142	0.9730	0.9031	0.9868
SVC2 (kernel = rbf)	0.8112	0.9189	0.7908	0.9458
GBDT	0.9142	0.9189	0.9133	0.9786
ANN (MLP)	0.9099	1.0000	0.8929	0.9903
KNN1 (K = 5)	0.8584	0.8108	0.8673	0.8955
KNN2 (K = 7)	0.8369	0.8378	0.8367	0.8949
Stacking (RF1, XGB, LR1)	0.9270	0.8919	0.9337	0.9852
Voting (RF1, XGB, LR1)	0.9270	0.8919	0.9337	0.9851
Stacking (RF1, XGB, LR1, MLP, SVC1)	0.9270	0.8919	0.9337	0.9861
Voting (RF1, XGB, LR1, MLP, SVC1)	0.9313	0.9459	0.9286	0.9870

### 3.1.3. Ten-Fold Cross-Validation

10-fold cross-validation was conducted on the SMOTE data. The average accuracy and average recall of the model after ten fits and their respective standard deviations are shown in Table 10. The bold part is the model with the highest average accuracy, average recall, lower standard deviation of accuracy, and standard deviation of recall.

**Table 10.** Ten-fold cross-validation results.

Classifier	Average Accuracy	Accuracy Standard Deviation	Average 1 Class Recall	Recall Standard Deviation
RF1 (n = 50)	0.9550	0.0155	0.9794	0.0193
RF2 (n = 100)	0.9544	0.0136	0.9794	0.0175
RF3 (n = 500)	0.9550	0.0129	0.9794	0.0175
<b>XGBoost (n = 100)</b>	<b>0.9621</b>	<b>0.0139</b>	<b>0.9807</b>	<b>0.0155</b>
LR1 (kernel = L2)	0.9293	0.0198	0.9499	0.0285
LR2 (kernel = L1)	0.9068	0.0183	0.9434	0.0361
NB	0.8239	0.0306	0.9910	0.0115
DT	0.9357	0.0210	0.9459	0.0357
SVC1 (kernel = linear)	0.9313	0.0206	0.9563	0.0252
SVC2 (kernel = rbf)	0.8053	0.0209	0.8419	0.0402
GBDT	0.9473	0.0166	0.9717	0.0206
ANN (MLP)	0.8977	0.0457	0.9152	0.1108
KNN1 (K = 5)	0.9222	0.0135	0.9858	0.0203
KNN2 (K = 7)	0.9171	0.0123	0.9781	0.0224
Stacking (RF1, XGB, LR1)	0.9576	0.0180	0.9755	0.0168
Voting (RF1, XGB, LR1)	0.9537	0.0154	0.9794	0.0166
Stacking (RF1, XGB, LR1, MLP, SVC1)	0.9569	0.0168	0.9730	0.0147
Voting (RF1, XGB, LR1, MLP, SVC1)	0.9435	0.0169	0.9666	0.0252

As demonstrated in Table 10, XGBoost delivers optimal performance on the test set, achieving the highest accuracy, recall rate for good grains, and the lowest standard deviation. XGBoost achieved an accuracy of 0.9621 and a recall rate for good grains of 0.9807, with standard deviations of 0.0139 and 0.0155 for accuracy and good grain recall rate, respectively, demonstrating greater robustness and lower overfitting risk compared to other models.

Simultaneously, the Stacking ensemble (RF1, XGB, LR1, MLP, SVC1) demonstrates competitive recall performance, achieving an accuracy of 0.9569 and a recall rate for good grains of 0.9730, with standard deviations of 0.0168 and 0.0147 for accuracy and good grain recall rate, respectively. The Stacking ensemble (RF1, XGB, LR1, MLP, SVC1) demonstrates strong robustness with no evident signs of overfitting observed. However, its final recall performance falls short of XGBoost's, likely due to suboptimal recall contributions from the integrated LR1 model (kernel = L2). The Naive Bayes (NB) model achieves a recall rate for good grains of 0.9910 (standard deviation: 0.0115) and an accuracy of 0.8239 (standard deviation: 0.0306). However, the accuracy and recall rate metrics suggest that the model tends to overpredict grains as good, exhibiting suboptimal performance in identifying bad grains. The KNN (K = 5) and KNN (K = 7) models exhibit relatively small standard deviations for accuracy (0.0135 and 0.0123, respectively) but achieve lower accuracy scores of 0.9222 and 0.9171. Their standard deviations for good grain recall rates are comparatively larger at 0.0203 and 0.0224. In contrast, the ANN model shows higher standard deviations for both accuracy (0.0457) and good grain recall rate (0.1108), along with lower performance metrics (accuracy: 0.8977; good grain recall: 0.9152). The RF models demonstrate strong performance, with accuracy consistently exceeding 0.95 and a good grain recall rate of 0.9794 across all configurations. As the number of trees increases, the standard deviations for both accuracy and good grain recall decrease. For example, RF3 (n = 500) achieves standard deviations of 0.0129 (accuracy) and 0.0175 (good grain recall). Nevertheless, even the best RF model still underperforms compared to XGBoost.

### 3.2. Metallographic Rating

The metallographic image was segmented into grains through a series of operations, including grayscale conversion, denoising, binarization, and morphological operations, such as opening and closing, followed by connected domain analysis. To classify the grains, 15 features were extracted, and the grains were labeled as good or bad to create the grain dataset. Three feature selection methods were applied, and 18 categorical models were fitted. To enhance the model's generalization and robustness, SMOTE sampling and 10-fold cross-validation were performed. Finally, the proportion of the good grain area was calculated.

By observing many metallographic images and the proportion of good grain area, it is found that the higher the grade of metallographic, the higher the proportion of good grain area. By measuring the gloss values of anodized products using a gloss meter and comparing these measurements with the calculated percentage of good grain area from metallographic analysis, we identified a distinct one-to-one correspondence between gloss values and the proportion of good grains. By setting thresholds of different good grain area proportions, better rating results can be obtained, as shown in Table 11.

**Table 11.** Metallurgical image threshold rating.

Y-Gloss Value(GU)	X-Proportion of Good Grain Area	Metallographic Rating
$y \geq 530$	$x \geq 70$	A
$500 < y \leq 530$	$55 \leq x < 70$	B
$470 < y \leq 500$	$35 \leq x < 55$	C
$440 < y \leq 470$	$20 \leq x < 35$	D
$y \leq 440$	$x < 20$	E

### 3.3. Metallographic Rating System

By dividing the threshold of good grain area proportion to determine the aluminum alloy metallographic image grade, the above image processing to divide the grain, establish

grain classification models, calculate the good grain area proportion, and then grade the metallographic image, this series of steps by using the Python 3.8.8 third-party library “pyinstall” packaged into a .exe executable program, that is, the metallographic rating system, which is used for the actual metallographic rating. The metallographic grading system achieved a processing time of only 8.79 s per image when evaluating test metallographic samples. The system rating results are shown in Figure 11. In Figure 11, different colors represent different grains.

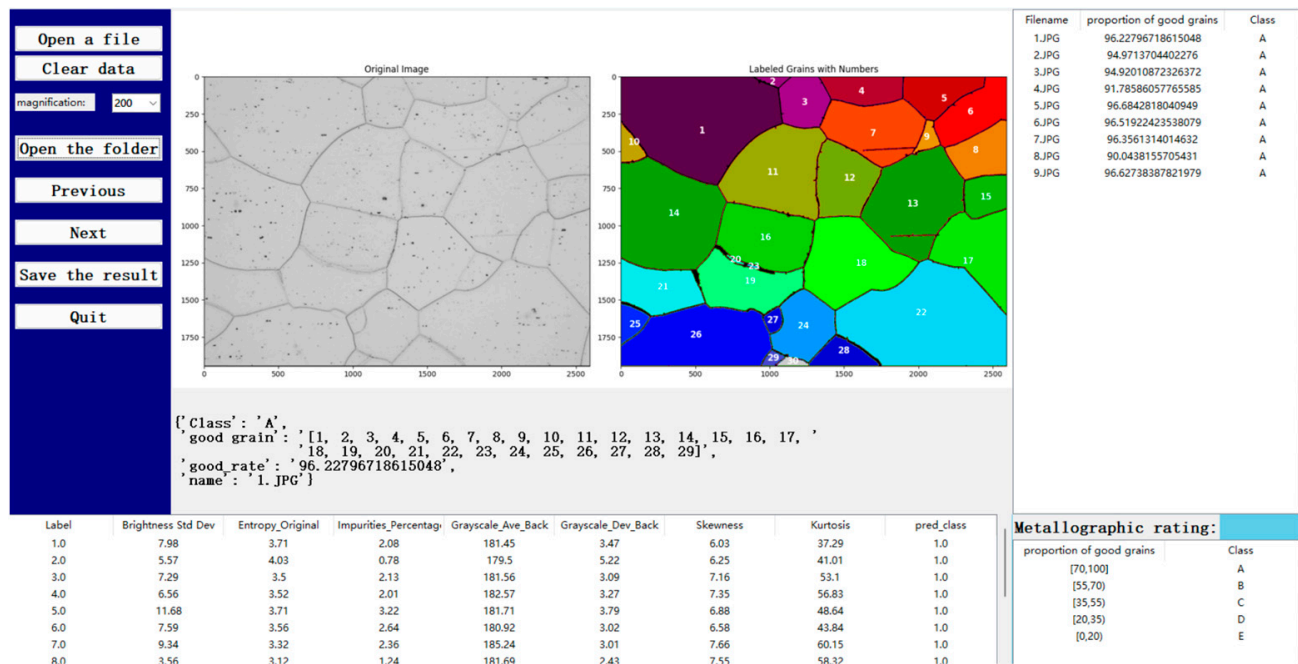


Figure 11. Interface of the metallographic rating system.

## 4. Discussion

### 4.1. Image Processing Results Are Discussed

In the image processing process, from the initial image preprocessing to the final grain boundary extraction and grain division, every step is carefully designed and repeatedly verified. CLAHE processing, median filtering, and bilateral filtering used in image preprocessing are aimed at effectively removing noise and enhancing image contrast and sharpness, which lays a good foundation for subsequent analysis. In practice, by comparing the images before and after processing, it is found that CLAHE processing makes the grayscale distribution of the images more uniform, and the grain region, which was blurred by uneven illumination, becomes clearly distinguishable. The median filter performs an excellent job of removing salt and pepper noise, while the bilateral filter further smooths the image while maintaining grain boundary detail.

In the process of grain boundary extraction and grain segmentation, connected domain analysis, as well as opening and closing operations of morphological processes, are applied repeatedly. The thresholds for these operations are optimized through extensive experimentation. Taking a typical set of metallographic images as an example, in the initial attempt, due to improper threshold setting, the grains appear over-segmented or merged. After many tests and adjustments of different metallographic samples, the final threshold can accurately identify the grain boundaries and completely divide the grains, ensuring the accuracy of grain feature extraction.

#### 4.2. The Results of Grain Classification Are Discussed

In the task of grain classification, most of the classifiers have shown some effectiveness on the original dataset, but the problem of the number of good grain samples in the dataset being too small cannot be ignored. The use of the SMOTE oversampling method has significantly improved the ability of most classifiers to identify good grain. Taking the tree classifiers as examples, the recall of good grains is about 80% before oversampling. After SMOTE oversampling, the recall increased to about 90%. However, there is blindness in the SMOTE method. From the perspective of data distribution, synthesized new samples may change the distribution characteristics of the original data, resulting in bias in the learning process of the model. For example, in some cases, the synthesized sample may be concentrated in a local region of the original data, making the model less able to generalize data to other regions.

After a series of processing, the XGBoost classifier performs well. Through the verification of different batches of test sets, the accuracy is always stable at more than 96%, the recall is about 98%, and the standard deviation is less than 0.02, which indicates that the model has a high degree of reliability and stability. Compared with other models, such as SVC (rbf), the recall is 0 when the data imbalance problem is not addressed, and accuracy is high but lacks practical significance. After the same processing steps, XGBoost is far superior to other models in terms of comprehensive performance and can accurately classify grains, meeting the strict requirements for grain classification in actual metallographic rating.

Since equal class weights were assigned to multiple model parameters—a suboptimal approach for imbalanced datasets—weaker identification of good grains was exhibited by models trained without sampling. However, after balancing the dataset via SMOTE sampling, most models showed significant improvement in recognizing good grains. Notably, the Naive Bayes (NB) model trained on the SMOTE-balanced dataset displayed no enhancement in good grain identification; instead, its performance in detecting poor grains degraded. This is likely due to SMOTE altering the original data distribution, which conflicts with the NB model's sensitivity to distributional assumptions, thereby violating its theoretical prerequisites and reducing model efficacy.

XGBoost demonstrated superior performance, likely due to its ability to enhance overall prediction accuracy by combining multiple weak learners and incorporating regularization terms that control model complexity. However, as a black-box model, it exhibits limited interpretability of predictions. In practical scenarios, model performance may further be influenced by data noise, feature selection, and parameter tuning.

#### 4.3. Metallographic Rating Results Discussed

In the process of exploring metallographic rating methods, the deep learning network model cannot reach the ideal accuracy due to the limitation of the small amount of data, even if the data enhancement technology is adopted. For example, when using a certain deep learning architecture, after data enhancement of the training set, the accuracy of the model on the verification set is only about 70%, and the prediction results on the actual test images are significantly different from the expert annotations.

After adopting the rating method based on grain classification and area proportion, it was found that there is an obvious correlation between different grades of metallographic images and the proportion of good grain area through the experimental analysis of a large number of metallographic images. After statistical analysis of 200 different grades of metallographic images, it is found that the average proportion of good grain area of grade A metallographic images is more than 70%, grade B is between 55 and 70%, grade C is between 35 and 55%, grade D is between 20 and 35%, and grade E is less than 20%. Based on this, the metallographic rating system has excellent performance in practical applications,

with an average time of less than 9 s. Compared with traditional manual rating, the efficiency is greatly improved, and the accuracy is effectively guaranteed, providing an efficient and reliable solution for metallographic analysis and quality control of 3C profiles.

## 5. Conclusions

1. This study correlated microstructural characteristics with anodizing effects in high-brightness aluminum extrusion profiles for mobile phones via experimental and metallographic analysis. A rating system using high-quality grain proportion as a key indicator classified anodizing performance into five grades (A–E). Results indicated that glossiness  $\geq 500$  GU (qualified) was achieved when the high-quality grain (grains with high brightness and low second-phase content) area exceeded 55%.

2. In the feature extraction stage, multiple feature values were calculated for the original image, impurity image, and background image. Feature screening was performed using the embedding method, variance method, and correlation coefficient method. Finally, key features—Impurities\_Percentage (percentage of impurities), Brightness\_Dev (brightness deviation), Mode (mode), Kurtosis, and Over\_180\_rate (rate of gray value greater than 180 in the grain)—were identified and used for subsequent analysis.

3. In model construction, the SMOTE oversampling technique was applied to balance the dataset to address data imbalance and 10-fold cross-validation was used to optimize the model's generalization ability. With accuracy, recall, and AUC as evaluation metrics, 18 classification models were compared and tested. Experimental verification showed that the XGBoost model performed best. On the test set, it achieved a grain classification accuracy of 96.21%, a recall rate of good grains of 98.07%, a single-image rating time of less than 9 s, and standard deviations of both accuracy and recall less than 0.02, demonstrating high precision and strong robustness.

4. This study successfully developed a metallographic rating system for ultra-bright aluminum profiles for mobile phones by constructing a dataset and integrating computer vision with machine-learning methods. The system automatically determines the metallographic grade based on the proportion of high-quality grain areas, effectively reducing evaluation costs for metallographic analysis. It lays a solid foundation for efficient and large-scale metallographic analysis and quality control, contributing to improved production quality and control levels of aluminum profiles for mobile phones.

### 5. Limitations and Future Research Directions:

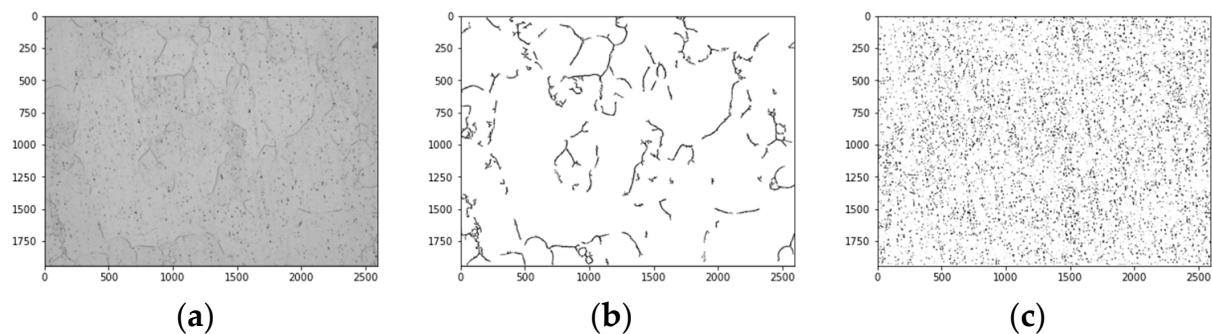
(a) During model fitting, some models lacked fine-tuning of parameters, potentially limiting their full predictive potential. Future work will conduct in-depth parameter optimization for each model to leverage their strengths and enhance system prediction accuracy.

(b) The metallographic grading system investigated in this paper was developed using training data from EN AW-6013 aluminum alloy, with its applicability to other alloys remaining unvalidated. Notably, high-gloss anodized surfaces are primarily applied to alloys such as EN AW-6063 (EN 573-3:2019+A2:2023), EN AW-6061 (EN 573-3: 2019+A2:2023), and EN AW-7003 (EN 573-3: 2019+A2:2023), which exhibit significant compositional variations in their primary alloying elements. Additionally, given the influence of recrystallization-inhibiting elements like Mn, Cr, and Zr on microstructural evolution, future research expansions may require designing alloy-specific metallographic preparation protocols and establishing corresponding datasets to systematically evaluate correlations between alloy compositions and anodization performance.

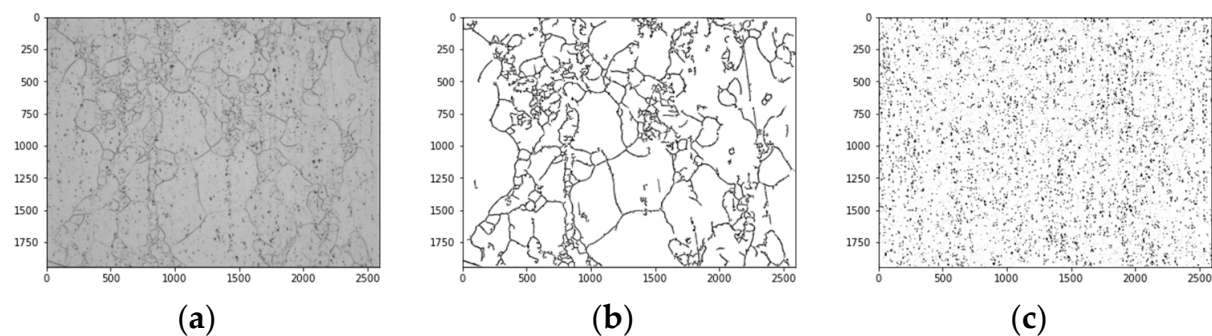
Research on 6061 (EN 573-3) aluminum alloy has been initiated. Samples with varying gloss values have been collected, and metallographic inspection along with image processing is currently underway, with selected results shown in Figures 12 and 13. Subsequent studies will involve feature extraction, dataset construction, and model training. Looking



ahead, this metallographic grading system will undergo continuous improvement based on the findings of this paper to achieve automated metallographic grading capabilities across diverse alloy types.



**Figure 12.** 6061 (EN 573-3) with 451GU gloss value. (a) Original image; (b) Grain boundary extraction; (c) Distribution of the second phase.



**Figure 13.** 6061 (EN 573-3) with 323GU gloss value. (a) Original image; (b) Grain boundary extraction; (c) Distribution of the second phase.

**Author Contributions:** Conceptualization, X.X. and F.J.; methodology, X.X.; software, L.L. and F.Y.; validation, X.X., F.J. and C.J.; formal analysis, X.X. and H.H.; investigation, X.X. and H.H.; resources, X.X. and H.H.; data curation, X.X. and H.H.; writing—original draft preparation, X.X.; writing—review and editing, X.X. and L.L.; visualization, L.L. and F.Y.; supervision, F.J.; project administration, X.X.; funding acquisition, H.H. and F.J. All authors have read and agreed to the published version of the manuscript.

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

**Data Availability Statement:** The original contributions presented in the study are included in the article. Further inquiries can be directed to the corresponding author.

**Conflicts of Interest:** Xuda Xu and Lurong Li were employed by Guangdong Hoshion Aluminium Co., Ltd., and Chunli Jiang was employed by the Guangdong Institute of Special Equipment Inspection and Research Zhongshan Branch. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## References

1. Hu, L.; Chen, D.; Shi, F.; Chen, S. Effect of AlSiFe on the anodizing process of 6063 aluminum. *Surf. Interface Anal. Sia* **2016**, *48*, 1299–1304. [[CrossRef](#)]
2. Wang, G.; Song, D.; Zhou, Z.; Klu, E.E.; Liu, Y.; Liang, N.; Jiang, J.; Sun, J.; Ma, A. Effect of Ultrafine Grains on the Coating Reaction and Anticorrosion Performance of Anodized Pure Aluminum. *Coatings* **2020**, *10*, 216. [[CrossRef](#)]
3. Tamadon, A.; Pons, D.J.; Sued, K.; Clucas, D. Development of Metallographic Etchants for the Microstructure Evolution of A6082-T6 BFSW Welds. *Metals* **2017**, *7*, 423. [[CrossRef](#)]

4. Julián, L.; Raúl, M.; Iván, S.; David, C.; Adrián, P.; Marta, F.; Pablo, M.; Francisco, H. A tutorial on the segmentation of metallographic images: Taxonomy, new MetalDAM dataset, deep learning-based ensemble model, experimental analysis and challenges. *Inf. Fusion* **2022**, *78*, 232–253. [\[CrossRef\]](#)
5. Naik, D.L.; Sajid, H.U.; Kiran, R. Texture-Based Metallurgical Phase Identification in Structural Steels: A Supervised Machine Learning Approach. *Metals* **2019**, *9*, 546. [\[CrossRef\]](#)
6. Dali, C.; Dingpeng, S.; Jun, F.; Shixin, L. Semi-Supervised Learning Framework for Aluminum Alloy Metallographic Image Segmentation. *IEEE Access* **2021**, *9*, 30858–30867. [\[CrossRef\]](#)
7. Matan, R.; Ofer, B.; Gal, O. An end-to-end computer vision methodology for quantitative metallography. *Sci. Rep.* **2022**, *12*, 4776. [\[CrossRef\]](#)
8. Katika, H.; Maharajan, J.D.; Gottim, D.R.; Kasagani, V.V.N. Overcoming optical image challenges in automatic grain size measurement using a novel computer vision algorithm applied to hot deformation of Al-Zn-Mg Powder metallurgy alloy. *Mater. Lett.* **2024**, *357*, 135743. [\[CrossRef\]](#)
9. Majumdar, S.; Sau, A.; Biswas, M.; Sarkar, R. Metallographic image segmentation using feature pyramid based recurrent residual U-Net. *Comput. Mater. Sci.* **2024**, *244*, 113199. [\[CrossRef\]](#)
10. Germain, L.; Sertucha, J.; Hazotte, A.; Lacaze, J. Classification of graphite particles in metallographic images of cast irons—Quantitative image analysis versus deep learning. *Mater. Charact.* **2024**, *217*, 114333. [\[CrossRef\]](#)
11. Shi, W.; Zhao, H.; Zhang, H.; Song, L.; Chen, K.; Zhang, B. Wire melted mark metallographic image recognition and classification based on semantic segmentation. *Expert Syst. Appl.* **2024**, *238*, 13. [\[CrossRef\]](#)
12. Shen, Y.; Ma, F.; Tan, L. Development status and prospect of artificial intelligence recognition in metallographic inspection. *Phys. Exam. Test.* **2024**, *42*, 59–63. [\[CrossRef\]](#)
13. Wang, S.; Guo, R.; Hu, H.; Zhang, Y.; Li, X. A deep learning-based method for grading the grain size of steel metallographic images. *J. Optoelectron. Laser* **2023**, *34*, 1075–1083. [\[CrossRef\]](#)
14. Yue, S.; Zhiguo, A.; Lijuan, B.; Xiurui, G.; Wenjin, Y.; Lijun, L. Automatic ferrite grain size rating method based on deep learning. *Hebei Metall.* **2023**, 73–77. [\[CrossRef\]](#)
15. Su, C. Deep Learning-Based Segmentation and Grading of Metallographic Organization of Carburized Gears. Master's Thesis, Jiangnan University, Wuxi, China, 2023. [\[CrossRef\]](#)
16. Reza, A.M. Realization of the Contrast Limited Adaptive Histogram Equalization (CLAHE) for Real-Time Image Enhancement. *J. Vlsi Signal Process.* **2004**, *38*, 35–44. [\[CrossRef\]](#)
17. Justusson, B.I. Median filtering: Statistical properties. In *Two-Dimensional Digital Signal Processing II*; Springer: Berlin/Heidelberg, Germany, 1981; pp. 161–196. [\[CrossRef\]](#)
18. Nirbhav; Malik, A.; Maheshwar; Jan, T.; Prasad, M. Landslide Susceptibility Prediction based on Decision Tree and Feature Selection Methods. *J. Indian Soc. Remote Sens.* **2023**, *51*, 771–786. [\[CrossRef\]](#)
19. Necati, Ç. Machine Learning for Varietal Binary Classification of Soybean (*Glycine max* (L.) Merrill) Seeds Based on Shape and Size Attributes. *Food Anal. Meth.* **2022**, *15*, 2260–2273. [\[CrossRef\]](#)
20. Mahajan, M.; Kadam, S.; Kulkarni, V.; Gujar, J.; Naik, S.; Bibikar, S.; Ochani, A.; Pratap, S. ECG signal classification via ensemble learning: Addressing intra and inter-patient variations. *Int. J. Inf. Technol.* **2024**, *16*, 1–9. [\[CrossRef\]](#)
21. Dai, Q. Research of decision tree classification algorithm in data mining. *Int. J. Database Theory Appl.* **2016**, *5*, 1–8. [\[CrossRef\]](#)
22. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [\[CrossRef\]](#)
23. Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [\[CrossRef\]](#)
24. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [\[CrossRef\]](#)
25. Wolpert, D.H. Stacked generalization. *Neural Netw.* **1992**, *5*, 241–259.
26. Bender, R.; Ziegler, A.; Lange, S. Logistische Regression. *Dtsch. Med. Wochenschr.* **2007**, *132*, 33–35. [\[CrossRef\]](#)
27. Bhowmik, T.K. Naive Bayes vs Logistic Regression: Theory, Implementation and Experimental Validation. *Intel. Artif.* **2015**, *18*, 14–30. [\[CrossRef\]](#)
28. Karlsen, R.E.; Gorsich, D.J.; Gerhart, G.R. Target classification via support vector machines. *Opt. Eng.* **2000**, *39*, 704–711. [\[CrossRef\]](#)
29. Zhongheng, Z. Introduction to machine learning: K-nearest neighbors. *Ann. Transl. Med.* **2016**, *4*, 218. [\[CrossRef\]](#)
30. Gardner, M.W. Artificial neural networks (the multilayer perceptron)—A review of applications in the atmospheric sciences. *Atmos. Environ.* **1998**, *14–15*, 2627–2636. [\[CrossRef\]](#)
31. Quanshang, S. Research on Imbalanced Data Sets Classification Method. *J. Sci. Educ.* **2013**, *39*, 92–93. [\[CrossRef\]](#)
32. Lin, Z.; Hao, Z.; Yang, X. Effects of Several Evaluation Metrics on Imbalanced Data Learning. *J. South China Univ. Technol. (Nat. Sci. Ed.)* **2010**, *38*, 147–155. [\[CrossRef\]](#)
33. Jianhong, Y. Optimization boosting classification based on metrics of imbalanced data. *Comput. Eng. Appl.* **2018**, *54*, 128–132. [\[CrossRef\]](#)
34. Yang, S.; Luo, L.; Liu, H. Research on gradient lifting Algorithm of unbalanced Data Sets. *Microcomputers* **2024**, *3*, 67–69.

35. Elreedy, D.; Atiya, A.F.; Kamalov, F. A theoretical distribution analysis of synthetic minority oversampling technique (SMOTE) for imbalanced learning. *Mach. Learn.* **2023**, *113*, 4903–4923. [[CrossRef](#)]
36. Zhang, H.Y.; Liu, R.Y. Application of cross-validation method in model comparison. *Adv. Appl. Math.* **2023**, *4*, 1866–1873. [[CrossRef](#)]

**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.