The Mathematical Magic Behind Image Recognition

Yiyan Chi [1], Wanle Chi [2, 3, a]

[1]Grade 12, Class 9, Wenzhou No.2 Senior High School, Zhejiang Province, China

[2]College of artificial intelligence, Wenzhou Polytechnic , Wenzhou, Zhejiang 325035, China

[3]Faculty of Information and Communication Technology, University Teknikal Malaysia Melaka (UTeM), Malacca 76100, Malaysia

[a] chiokchi@163.com

Abstract

From the facial recognition technology that enables effortless unlocking of mobile phones to the automatic photo classification in albums that effectively organizes personal memories, from the autonomous driving systems that smoothly navigate around road obstacles to the supermarket scanners that accurately identify products at the checkout counter—these image recognition technologies, which have seamlessly integrated into our daily lives, often evoke a profound sense of "magic" among people. Whether it is the instant photo tagging function of mobile phone AI, which is empowered by advanced image recognition technology, or the real-time calorie recognition of food by smart watches, which uses sophisticated algorithms to analyze dietary intake, these convenient features are underpinned by the profound yet ingenious application of mathematical knowledge. This paper takes real-life scenarios as the starting point. It adopts simple and easily understandable language to comprehensively expound on the mathematical principles underlying core algorithms such as convolution, gradient descent, backpropagation, and non-maximum suppression, and supplements them with simplified and accessible formula derivations. The aim is to comprehensively and systematically uncover the mathematical mysteries behind image recognition technology, enabling readers to intuitively recognize that abstract mathematical theories have long been the core driving force for artificial intelligence to "understand the world". Moreover, it further promotes the popularization of mathematical knowledge in the field of artificial intelligence and the in-depth understanding of image recognition technology.

Keywords

Image Recognition; Mathematical Foundations; Convolution; Gradient Descent; Backpropagation.

1. Introduction

In the digital age, image recognition technology has pervaded diverse facets of daily life and emerged as an essential constituent of contemporary society. Specifically, mobile phone artificial intelligence (AI) systems can instantaneously identify and label key elements within images, thus facilitating efficient album organization and management. Self-service scanners in supermarkets can rapidly recognize product-related information to streamline the customer payment process and enhance the overall experience. Community security surveillance systems can issue early warnings upon detecting strangers loitering in critical areas. Moreover, the latest smart watch models launched in 2025 are equipped with real-time food calorie recognition functionality, enabling them to promptly calculate calorie content by capturing food images through built-in cameras. All the above-mentioned application scenarios are supported by image recognition technology.

Image recognition technology endows computers with "intelligent visual perception capabilities," enabling them to interpret image content in a way similar to human visual cognition. Nevertheless, unlike humans who directly perceive images through their visual senses, computers lack innate visual perception capabilities and can only process and analyze data in digital form. The core mechanism that transforms vivid images into analyzable digital data and generates accurate recognition judgments is based on mathematical principles. Throughout the entire image recognition process, from the conversion of image information into digital matrices, the extraction of image features through convolution operations, the optimization of recognition models via gradient descent and backpropagation algorithms, the elimination of overlapping recognition results using non-maximum suppression, to the reduction of uncertainties in the recognition process based on probability theory and the improvement of model efficiency and accuracy through optimization theory, each step heavily depends on mathematical tools, which constitute the fundamental foundation of image recognition technology.

The research on image recognition originated from the simulation of human visual perception, and the evolution of algorithm frameworks has driven the continuous improvement of technical performance. The transition from traditional manual feature extraction to deep learning-based automatic feature learning has been a pivotal breakthrough. Residual neural networks (He et al., 2016) [1] solved the gradient vanishing problem in deep network training, while the introduction of Transformer architecture (Dosovitskiy et al., 2021) [3] broke the limitations of convolutional neural networks and opened up new ideas for global feature extraction in image recognition.

The rise of deep learning has driven a revolutionary leap in image recognition technology. Compared with traditional manual feature extraction methods, deep learning models can automatically learn effective features from data. The attention mechanism proposed in "Attention is All You Need" (Vaswani et al., 2017) [2] has become a core component of modern image recognition models, enabling adaptive focus on key image regions. On this basis, vision Transformers (Dosovitskiy et al., 2021) [3] have realized end-to-end image recognition by treating image patches as sequence tokens, and recent CLIP-adapter optimization methods (Ye et al., 2025) [4] have further improved the fine-grained perception ability of pre-trained models.

With the continuous expansion of the application fields of image recognition, its role in industrial production, medical care, and remote sensing has become increasingly prominent. In industrial scenarios, multi-sensor fusion-based anomaly detection methods (Li et al., 2025) [5] have solved the problem of incomplete product information capture in automatic quality inspection, while physics-grounded dynamic anomaly detection (Li et al., 2025) [6] can effectively identify functional defects of products. In the medical field, AI-driven diagnostic tools (Smith et al., 2025) [10] have shown great potential in improving the efficiency and accuracy of radiological diagnosis, and advanced segmentation models (Chen et al., 2026) [7] have provided strong support for precise clinical treatment planning. This paper delves into the mathematical magic that makes these technological marvels possible.

2. Mathematical Principles Underlying Image Feature Extraction

In the domain of image processing, linear algebra assumes a crucial position in extracting hierarchical features via operations like convolution, padding, and stride. These methodologies utilize linear transformations and matrix manipulations to conduct image analysis and processing, facilitating the extraction of detailed information at diverse levels of abstraction.

From a computational standpoint, colorful images encountered in daily situations are represented in a fundamentally distinct form; they are essentially a sequence of numerically - arranged data. As a fundamental discipline in mathematics, linear algebra serves as the core "linguistic tool" for transforming image information into digital representations. Among its significant applications in image recognition, the convolution operation (integrating padding and stride mechanisms) is considered the "core algorithmic component," tasked with the vital function of extracting hierarchical key features from images.

First, the digital representation of images is expounded here. A grayscale image is essentially a two - dimensional digital matrix. Taking a standard 100×100 - pixel grayscale image as an example, each element within the matrix corresponds to a specific pixel in the image, with numerical values spanning from 0 to 255. In the context of digital image processing, grayscale images consist of pixels ranging from 0, denoting pure black, to 255, signifying pure white. Intermediate values correspond to various gray tones, and the entire grayscale image is perceived by human observers as a continuous spectrum of these tones. Color images are based on the RGB color model, which encompasses three primary colors: Red (R), Green (G), and Blue (B). Each color channel corresponds to a grayscale matrix that captures luminance information. Collectively, these matrices combine to form the full - color images we observe.

Notably, the digital matrix of an image only encodes the luminance information of individual pixels and does not directly contain the key features necessary for recognition tasks, such as the contour of a cat's ears, the linearity of lane markings, or the geometric profile of product packaging. To enable computational systems to extract these key features from large - scale pixel datasets, the convolution operation (aided by padding and stride techniques) functions as an essential core process.

2.1. Convolution Operation

The detailed description of the digital representation of images is as follows: a grayscale image is essentially a two - dimensional (2D) digital matrix. Taking a standard 100×100 - pixel grayscale image as an example, each element in the matrix corresponds to a specific pixel in the image. The numerical values range from 0 (representing pure black) to 255 (representing pure white), and the intermediate values correspond to different gray levels.

Regarding color images, the underlying structure can be processed by means of techniques such as image segmentation and registration, as well as color quantization using superpixels. The principle is similar but more complex: three independent grayscale matrices, corresponding to the Red (R), Green (G), and Blue (B) color channels respectively, are superimposed. Each matrix encodes the luminance information of its corresponding channel, and the combined effect of these three matrices forms the colorful images perceived visually.

It should be noted that the digital matrix of an image only encodes the luminance information of individual pixels and does not inherently capture the essential features necessary for recognition tasks (e.g., the contour of cat ears, the straight lines of lane markings, the geometric shapes of product packaging). Therefore, convolution operations, combined with padding and stride techniques, are indispensable for computational systems to extract these crucial features from large - scale pixel datasets.

Simplified convolution formula:

$$
\begin{aligned}
(\mathrm{I} * \mathrm{K})_{\mathrm{i,j}} = {} & \mathrm{I}_{\mathrm{i,j}} * \mathrm{K} + \mathrm{I}_{\mathrm{i,j+1}} * \mathrm{K}_{1,2} + \mathrm{I}_{\mathrm{i,j+2}} * \mathrm{K}_{1,3} + \mathrm{I}_{\mathrm{i+1,j}} * \mathrm{K}_{2,1} + \mathrm{I}_{\mathrm{i+1,+1j}} * \mathrm{K}_{2,2} + \mathrm{I}_{\mathrm{i+1,j+2}} * \mathrm{K}_{1,3} \\
& + \mathrm{I}_{\mathrm{i+1,j+2}} * \mathrm{K}_{2,3} + \mathrm{I}_{\mathrm{i+2,j}} * \mathrm{K}_{3,1} + \mathrm{I}_{\mathrm{i+2,j+1}} * \mathrm{K}_{3,2} + \mathrm{I}_{\mathrm{i+2,j+2}} * \mathrm{K}_{3,3}
\end{aligned}
$$

In the formula:

I represents the original image matrix, and I_{ij} represents the pixel value at the position (i, j) in the image matrix;

K represents the 3×3 convolution kernel, and $K_{m,n}$ (m, n = 1, 2, 3) represents the element value at the position (m, n) in the convolution kernel;

$(I*K)_{i,j}$ represents the feature value at the position (i, j) in the feature map obtained after the convolution operation.

To illustrate, consider the application of the convolution kernel $\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$. This kernel is characterized by a specialized design, where the central element is assigned a value of 8, and all peripheral elements are set to -1. In the convolution process, the central pixel in the target image region is notably emphasized, as reflected by the corresponding weight in the convolution matrix. Conversely, the surrounding pixels are suppressed due to their relatively lower weights. The Laplacian convolution kernel, a computational model, magnifies the intensity difference between the central pixel and its neighboring pixels, thus effectively enhancing the image edge features. This specific convolution kernel, commonly employed in image edge detection tasks, is based on a unique design principle that has been demonstrated to be effective in identifying edges and object contours in images.

The histogram of oriented gradients (HOG) algorithm (Dalal & Triggs, 2005) [8] is a classic texture feature extraction method that captures local gradient information of images, and it has been widely used in object detection due to its strong robustness to illumination and shadow changes. This method lays a foundation for the subsequent development of multi-modal feature fusion technologies (Li et al., 2025) [5].

2.2. Padding

A critical challenge emerges during the convolution operation: as the convolution kernel slides across the image matrix, edge pixels of the original image participate in fewer computational iterations compared to central pixels. This phenomenon results in a reduction in the dimensions of the generated feature map and the loss of edge feature information. To address this issue, the "padding" technique is introduced. Specifically, padding involves pre-convolutionally appending a layer of pixels (typically zero-padding) around the original image matrix, ensuring that edge pixels of the original image participate in convolution computations with the same frequency as central pixels.

Padding size calculation formula :

$$P = (K - 1)/2$$

Where:

P denotes the number of pixel layers appended around the image (i.e., padding size);

K represents the size of the convolution kernel (assuming K is an odd number, which is the predominant scenario in image recognition tasks).

For instance, when a 3×3 convolution kernel (K=3) is employed, the padding size P is calculated as (3-1)/2 = 1. This implies appending a single layer of zero pixels around the original image. Post-padding, the dimensions of the feature map derived from convolution are consistent with those of the original image, thereby effectively preserving edge feature information.

2.3. Stride

Stride is defined as the number of pixels traversed by the convolution kernel in each movement during the sliding process. A stride of 1 implies a one - pixel displacement per iteration, while a stride of 2 denotes a two - pixel displacement. In convolutional neural networks, the stride size directly impacts the resolution of the resultant feature maps. A smaller stride yields a larger feature map containing more detailed information; however, this comes at the expense of increased computational complexity. Conversely, a larger stride generates a smaller feature map, which reduces computational requirements but may lead to the loss of some fine - grained details.

Feature map size calculation formula (after padding and stride):

$$H_{out} = \mathrm{Floor}(\frac{H_{in} + 2 * P - K}{S} + 1)$$

$$W_{out} = \mathrm{Floor}(\frac{W_{in} + 2 * P - K}{S} + 1)$$

Where:

H_{out} and W_{out} denote the height and width of the output feature map, respectively;

H_{in} and W_{in} represent the height and width of the input image, respectively;

P denotes the padding size;

K represents the convolution kernel size;

S denotes the stride;

Floor() denotes the floor function, which truncates the fractional part of the computed result to retain the integer component.

For illustrative purposes, consider a 100×100 input image (H_{in}=100, W_{in}=100) processed with a 3×3 convolution kernel (K=3), a padding size of P=1, and a stride of S=1. The output feature map dimensions are calculated as (100+2*1-3)/1+1=100, which is consistent with the input image dimensions. When the stride is increased to S=2, the output feature map dimensions become (100+2*1-3)/2+1=50, resulting in a halved resolution.

2.4. Application Case

The "AI food calorie recognition" watches that gained popularity rely heavily on convolution operations (integrated with padding and stride) for their core functionality. When utilizing the smartwatch for food recognition, the device's integrated camera captures an image of the food to analyze its nutritional content. Subsequently, the embedded algorithm executes convolution operations on the food image using multiple sets of convolution kernels with varying sizes and strides. Specifically, the first convolutional layer utilizes a 3×3 kernel with S=1 and P=1 to extract low-level features such as food edges and textures; the second The layer utilizes a 5×5 kernel with a stride (S) of 2 and padding (P) of 2 to extract mid-level features, such as local food shapes; the upper layers employ a 7×7 kernel with S=1 and P=3 to capture high-level features, like the overall food morphology. These convolution kernels are specifically tailored to distinct food features: some are designed for extracting shape features (e.g., the circular bread contour of hamburgers, the irregular shape of French fries), while others are for extracting color features (e.g., the red hue of apples, the brown shade of chocolate, the green tint of vegetables), and some for texture feature extraction (e.g., the granular texture of cookies, the smooth surface of eggs). Following the extraction of these key features via convolution operations, The smartwatch quickly cross-references the food's characteristics with its extensive internal food feature database, aiming to estimate the calorie content within a second, although the accuracy of such estimations can vary. It is thus evident that convolution operations, augmented by padding and stride techniques, The smartwatch serves as a key facilitator for rapidly integrating food features and striving for precise calorie recognition.

3. Image Recognition's Mathematical Cornerstones

3.1. Calculus & Backpropagation

After the extraction of key image features via convolution operations, computational systems proceed to the critical subsequent stage of generating accurate inferences based on these features. To achieve this, a learning-based model is requisite. This model is inherently dynamic;

it can iteratively augment its inferential accuracy through continuous learning. The core algorithms enabling the model's iterative enhancement are gradient descent and backpropagation, which are grounded in the derivative concept in calculus.

3.1.1. Loss Function-Quantifying Prediction Errors

At the onset of model training, the model possesses limited experiential knowledge, akin to a novice reader, and thus tends to make errors. In the realm of machine learning, loss functions play a pivotal role in quantifying the disparities between model predictions and actual results. For instance, misclassifying a cat as a dog, an apple as an orange, or a cup as a bowl. To enable the model to identify and rectify errors, a loss function is indispensable as it functions as a quantitative measure to precisely gauge the discrepancy between the model's predictive outputs and the ground truth. In the field of machine learning, this metric is termed the "loss function."

A loss function is a mathematical construct that quantifies the error between the model's predictions and actual outcomes, serving as a crucial element in machine learning and deep learning. It guides the optimization process by offering a measure of the model's predictive accuracy, facilitating the adjustment of parameters to minimize this error and enhance performance. Its core function is to assess the model's predictive performance: a higher value of the loss function indicates a greater divergence between the predicted outputs and the ground truth, corresponding to less reliable inferences; conversely, a lower value of the loss function signifies a closer congruence between the predictions and the ground truth, representing more accurate inferential results.

In the context of image recognition models, the Mean Squared Error (MSE) loss is a commonly used loss function for regression tasks, providing a straightforward method to quantify the error between predicted and actual values. The log_loss function, suitable for regression tasks such as calorie prediction, and the Cross-Entropy (CE) loss function, applicable to classification tasks like object recognition, are widely used due to their effectiveness in evaluating model predictions.

Mean Squared Error (MSE) loss function:

$$\mathrm{L} = \frac{1}{2} * (\mathrm{y} - \hat{\mathrm{y}})^2$$

Where:

y denotes the ground truth (label);

ŷ denotes the model's predicted value;

L represents the error score.

Cross-Entropy (CE) loss function:

$$\mathrm{L} = -\sum(\mathrm{y}_\mathrm{i} * \ln(\hat{\mathrm{y}}_\mathrm{i}))$$

Where:

y_i denotes the ground truth label for the i-th category (assigned 1 if the object belongs to the i-th category, and 0 otherwise);

$\hat{\mathrm{y}}_\mathrm{i}$ denotes the probability that the model predicts the object to belong to the i-th category;

ln() denotes the natural logarithm function.

3.1.2. Gradient Descent-Finding the Optimal Parameter Adjustment Direction

To iteratively enhance the inferential accuracy of the model, it is essential to adjust the internal parameters of the model according to the error metric provided by the loss function, thus minimizing the value of the loss function. A crucial question emerges: how to determine the optimal direction for parameter adjustment? This is precisely the situation where the gradient descent algorithm showcases its core functionality.

A heuristic analogy can facilitate a deeper understanding of gradient descent. Consider the loss function as a topographic landscape with valleys (representing minimum values). The adjustment of the model's parameters, such as the weights in convolution operations, is akin to navigating towards these valleys to minimize the prediction error. It is analogous to an individual situated at an arbitrary position on this surface. The primary goal is to guide this individual to the depression (the minimum of the loss function) as efficiently as possible, as this minimum corresponds to the smallest prediction error and the highest inferential accuracy of the model.

The "gradient" in gradient descent refers to the derivative of the loss function with respect to the parameter ω, denoted as $\frac{dL}{d\omega}$. Functioning as a "directional indicator," the gradient precisely delineates the adjustment direction of parameter ω that enables the most rapid reduction of the loss function, thereby minimizing the model's prediction error with optimal efficiency.

Core update formula of gradient descent:

$$\omega_{new} = \omega_{old} - \eta * (\frac{dL}{d\omega})$$

Where:

ω_{old} denotes the original parameter value of the model prior to adjustment;

ω_{new} denotes the updated parameter value of the model post-adjustment;

η represents the "learning rate," analogous to the step size of the individual descending the topographic surface, governing the magnitude of each parameter adjustment;

$\frac{dL}{d\omega}$ denotes the gradient (i.e., the derivative of the loss function with respect to parameter ω), which determines the direction of parameter adjustment.

A concrete illustrative example is provided as follows:

Assume the model is tasked with classifying whether an object is a cat. The ground truth y = 1 (indicating the object is a cat), and the model's predicted value ŷ = 0.6. Utilizing the The gradient of the Mean Squared Error (MSE) loss function is calculated by taking the partial derivative of the loss with respect to each model parameter. For instance, in the context of a linear regression model with parameters w (weights) and b (bias), the gradient descent algorithm updates these parameters in the direction opposite to the gradient of the loss function with respect to w and b.

$$\frac{dL}{d\omega} = -(y - \hat{y}) * \frac{d\hat{y}}{d\omega}$$

For simplification, assum $\frac{d\hat{y}}{d\omega} = 1$; accordingly, $\frac{dL}{d\omega} = -(1 - 0.6) = -0.4$ .Given $\eta = 0.1$, substituting these values into the update formula yields:

$$\omega_{new} = \omega_{old} - 0.1 * (-0.4) = \omega_{old} + 0.04$$

This parameter adjustment facilitates the model's subsequent predictions to be more aligned with the ground truth.

3.1.3. Backpropagation-Efficiently Calculating Gradients for Deep Models

For deep convolutional neural networks (DCNNs) with multiple layers, directly calculating the gradient of each parameter using the chain rule is extremely complex. The backpropagation algorithm addresses this issue by computing the gradient in reverse, from the output layer back to the input layer, thereby significantly enhancing the efficiency of gradient computation.

For a DCNN consisting of L layers, each layer is equipped with trainable parameters (weights ω_l and biases b_l) and generates an output feature map described by $a_l = f(z_l)$. Here, $z_l = \omega_l * a_{l-1} + b_l$ (where * denotes the convolution operation) and $f(.)$ represents the activation function.

The loss function L is defined based on the final output a_L. To update the model parameters via gradient descent optimization, it is essential to compute the gradient $\frac{\partial_L}{\partial_{\omega i}}$ for each layer l.

The efficiency of backpropagation is particularly critical in deep networks. For example, ResNet models (He et al., 2016) [1] address the vanishing gradient problem through residual connections, and their parameter updates rely on the combination of gradient descent and backpropagation, improving training efficiency by more than three times compared to traditional networks.

3.1.4. Learning Rate Scheduling

The learning rate η is a crucial hyperparameter. If it is too large, the model may oscillate around the minimum value; if it is too small, the training speed will be extremely slow. In addition to the cosine annealing strategy mentioned earlier, the "step decay" strategy is also widely used:

$$\eta = \eta_0 * \gamma^{\mathrm{Floor(epoch/s)}}$$

In the formula:

η_0 represents the initial learning rate;

γ represents the decay rate (usually 0.1);

epoch represents the number of training rounds;

s represents the decay step (usually 10 or 20).

For example, if η_0=0.1,γ=0.1,s=10, then the learning rate will be reduced to 0.01 after 10 rounds of training and 0.001 after 20 rounds, ensuring fast convergence in the early stage and stable convergence in the later stage.

The “photo correction” function of the AI error notebook leverages gradient descent and backpropagation to continuously enhance accuracy. In the event that the function misclassifies a multiple - choice question as a fill - in - the - blank question, the user's manual correction serves as the ground truth label. The system computes the cross - entropy loss between the model's prediction and the ground truth label. Subsequently, backpropagation is employed to calculate the gradient of each layer's parameters from the output layer to the input layer. Ultimately, the parameters are updated in accordance with the gradient descent formula featuring a step - decay learning rate. After tens of thousands of iterations, the model has undergone fine - tuning to precisely recognize diverse question types and a broad spectrum of handwriting styles, as evidenced by its performance on datasets such as MNIST and DBRHD.

3.2. Probability Theory & Non-Maximum Suppression

In practical image recognition scenarios, numerous uncertain factors typically arise, such as image blurriness and special shooting perspectives. Consequently, the model may generate multiple overlapping recognition boxes for the same object target. Probability theory facilitates the management of uncertainty, whereas non - maximum suppression (NMS) refines the recognition outcomes by removing redundant boxes.

3.2.1. Softmax Function

Following the model's processing of image features, the initial output consists of a set of arbitrary real numbers (logits). The Softmax function, a mathematical activation function, converts a set of real numbers into a probability distribution where each number is transformed into a value between 0 and 1, and the sum of these probabilities is equal to 1.

Softmax function formula :

$$p_i = \frac{e^{z_i}}{(e^{z_1} + e^{z_2} + \ldots + e^{z_n})}$$

In the formula:

z_i represents the original output of the i-th category;

e is the base of the natural logarithm (≈2.71828);

p_i represents the probability of the i-th category.

For example, if the original outputs for "cat", "dog", and "rabbit" are 5, 3, and 1, the probabilities are:

$$p_{cat} = e^5/(e^5 + e^3 + e^1) \approx 86.7\%, p_{dog} \approx 11.7\%, p_{rabbit} \approx 1.6\%$$

The model selects the category with the highest probability as the result.

3.2.2. Gaussian Distribution & Gaussian Filtering

Most image noises follow a Gaussian distribution, whose The probability density function is:

The formula for the Gaussian distribution probability density is:

$$f(x) = \frac{1}{\sigma * \sqrt{2\pi}} * e^{(-(x-\mu)^2/(2\sigma^2))}$$

In the formula:

μ represents the mean (usually 0 for image noise);

σ represents the standard deviation (controlling the spread of noise);

x represents the noise pixel value.

Gaussian filtering, which employs a Gaussian kernel, leverages the Gaussian distribution to conduct a weighted average on image pixels. This kernel assigns greater weights to pixels that are nearer to the center, effectively smoothing the image while preserving edges. For a 3×3 Gaussian kernel with μ=0 and σ=1:

$$K = \frac{1}{16} * \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

Convolving the image with this kernel suppresses noise while preserving image details, improving recognition accuracy by about 20%.

3.2.3. Non-Maximum Suppression (NMS)

When recognizing objects (such as pedestrians in autonomous driving), the model may output multiple overlapping boxes for the same target. NMS eliminates redundant boxes by the following steps:

Sort all recognition boxes in descending order of probability (confidence score);

Select the box with the highest probability as the current optimal box, and calculate the intersection over union (IoU) between this box and all other boxes;

Delete all boxes with IoU greater than a preset threshold (usually 0.5);

Repeat steps 2-3 for the remaining boxes until no redundant boxes are left.

IoU calculation formula:

$$IoU = \frac{Area(Box1 \cap Box2)}{Area(Box1 \cup Box2)}$$

In the formula:

Box1 and Box2 represent two recognition boxes;

Area(Box1 ∩ Box2 represents the area of the intersection of the two boxes;

Area(Box1 ∪ Box2) represents the area of the union of the two boxes.

For example, if two overlapping boxes have an IoU of 0.6 (greater than 0.5), the box with the lower probability is deleted, retaining only the optimal box. This ensures that each target has only one accurate recognition box.

In the real-time semantic segmentation of autonomous driving, the model first uses Gaussian filtering to denoise the road image, then uses the Softmax function to calculate the probability of each pixel belonging to pedestrians, vehicles, and lane lines. For pedestrian recognition, multiple overlapping boxes may be generated. Non-Maximum Suppression (NMS) is a technique widely used in object detection to filter out overlapping bounding boxes. By calculating the Intersection over Union (IoU) between these boxes, NMS identifies and removes redundant ones, ensuring only the most accurate and non-overlapping boxes are retained. This ensures that the autonomous driving system can clearly identify each traffic participant and make timely decisions.

3.3. Optimization Theory

The ultimate goal of model training is to find optimal parameters with good generalization ability. Optimization theory provides Methods to address overfitting and enhance training speed, including regularization, AdamW, and Mini-Batch Gradient Descent

3.3.1. L2 Regularization: Preventing Overfitting

Overfitting occurs when the model memorizes training data details. L2 regularization incorporates a squared parameter term into the loss function to constrain parameter magnitude.

L2 regularization formula :

$$\mathrm{L}_{\mathrm{regularization}} = \mathrm{L} + \lambda * \sum \omega_{\mathrm{i}}{}^{2}$$

In the formula:

L represents the original loss function;

λ represents the penalty coefficient (controlling regularization strength);

$\sum \omega_{\mathrm{i}}{}^{2}$ represents the sum of squares of model parameters.

For instance, with a regularization strength λ of 0.01, the model not only reduces prediction errors but also constrains parameter values, ensuring it concentrates on essential features, such as the outline of a cat, while disregarding less significant details, such as fur color.

3.3.2. AdamW Algorithm: An Enhanced Version of Adam with Weight Decay

AdamW combines momentum, adaptive learning rate, and independent weight decay, with the following parameter update formula:

AdamW parameter update formula :

$$\mathrm{m}_{\mathrm{t}} = \beta_1 * \mathrm{m}_{\mathrm{t}-1} + (1-\beta_1) * \mathrm{g}_{\mathrm{t}}$$

$$\mathrm{v}_{\mathrm{t}} = \beta_2 * \mathrm{v}_{\mathrm{t}-1} + (1-\beta_2) * \mathrm{g}_{\mathrm{t}}{}^{2}$$

$$\widehat{\mathrm{m}}_{\mathrm{t}} = \frac{\mathrm{m}_{\mathrm{t}}}{1-\beta_1^{\mathrm{t}}}$$

$$\hat{\mathrm{v}}_{\mathrm{t}} = \frac{\mathrm{v}_{\mathrm{t}}}{1-\beta_2^{\mathrm{t}}}$$

$$\omega_{\mathrm{t}} = \omega_{\mathrm{t}-1} - \eta * \frac{\widehat{\mathrm{m}}_{\mathrm{t}}}{\sqrt{\hat{\mathrm{v}}_{\mathrm{t}}} + \varepsilon} - \eta * \lambda * \omega_{\mathrm{t}-1}$$

In the formula:

m_{t} represents the first moment (momentum);

v_{t} represents the second moment (adaptive learning rate);

g_{t} represents the gradient of the loss function at time t;

β_1 (≈0.9), β_2 (≈0.999) are exponential decay rates;

ε (≈1e-8) is a small constant to avoid division by zero;

λ represents the weight decay coefficient.

AdamW's independent weight decay solves the problem that Adam's adaptive learning rate weakens regularization, making the model more robust. Its training speed is 3 times faster than traditional algorithms.

3.3.3. Mini-Batch Gradient Descent: Balancing Speed and Stability

Mini-Batch Gradient Descent divides training data into small batches (100-200 samples) for parameter updates. The gradient of each batch is:

Mini-Batch gradient formula:

$$g_t = (\frac{1}{B}) * \sum_i^B \nabla_\omega L(\omega, x_i, y_i)$$

In the formula:

B represents the batch size;

x_i, y_i represent the i-th sample and its label;

$\nabla_\omega L$ represents the gradient of the loss function with respect to parameters.

This method balances computational efficiency (faster than Batch Gradient Descent) and stability (more stable than Stochastic Gradient Descent), and is widely used in large-scale model training.

Baidu's Plant Recognition APP uses L2 regularization and AdamW for model training. During training, 200 plant images are selected as a mini-batch each time, and the AdamW algorithm is used to update parameters with β_1=0.9, β_2=0.999, λ=0.001. L2 regularization, by penalizing larger weights and thus reducing the magnitude of model parameters, helps in preventing the model from overfitting to specific plant photos. This technique, combined with the AdamW optimizer, ensures a fast and stable training process. The final model can accurately recognize thousands of plant species with a size reduced by 10 times, suitable for mobile phone deployment.

The design of modern image recognition models draws on the advantages of different algorithm frameworks, and the integration of convolutional and Transformer structures has become a mainstream trend. Residual neural networks introduced skip connections to realize deep network training, while the self-attention mechanism (Vaswani et al., 2017) [2] enabled models to capture long-range dependencies. The combination of these two technologies has promoted the development of hybrid models, and the latest point-level prompting methods (Ai et al., 2025) [9] have further improved the robustness of 3D point cloud image analysis models.

4. Image Recognition in Daily Life

4.1. AI Food Calorie Recognition Watch

Convolution & Padding & Stride: Uses 3×3 kernels with P=1, S=1 to extract low-level features, 5×5 kernels with P=2, S=2 to extract middle-level features, and 7×7 kernels with P=3, S=1 to extract high-level features.

Softmax & Gaussian Filtering: Gaussian filtering denoises images, and Softmax calculates food category probabilities.

Gradient Descent & Backpropagation: Uses mean squared error loss to optimize calorie prediction, with backpropagation calculating gradients and step decay learning rate ($\eta\_0 = 0.1, \gamma = 0.1, s = 10$).

4.2. Smart Supermarket Self-Service Checkout Counter

3D Convolution + Convolution: 3D convolution extracts product shape features, and 2D convolution extracts logo and color features.

Mini-Batch Gradient Descent + L2 Regularization: Batch size B=150, λ=0.001, training model to recognize deformed and occluded products.

NMS + Semantic Segmentation: Semantic segmentation separates products and hands, and NMS eliminates redundant recognition boxes (IoU threshold=0.4).

4.3. AI Medical Imaging Screening

Multi-layer Convolution: The bottom layer uses 3×3 kernels to extract lesion edges, while the upper layer employs 7×7 kernels to extract lesion shapes.

Gaussian Distribution + Probability Calculation: It models the features of benign/malignant lesions using Gaussian distribution and calculates classification probabilities.

L2 Regularization: With $\lambda=0.01$, it prevents the model from missing atypical lesions.

In the medical field, image recognition technology plays an important role in disease diagnosis and medical image analysis. The DAF-Mamba model proposed by Chen et al. (2026) [7] achieves high-precision segmentation of cardiac images by integrating dynamic multi-scale selection and adaptive feature fusion modules, which effectively solves the problem of blurred boundaries in traditional segmentation methods. In addition, the evaluation of AI-driven diagnostic tools (Smith et al., 2025) [10] provides a structured guide for the clinical application of image recognition technology, ensuring the reliability and utility of these tools in radiological diagnosis.

4.4. Remote Sensing and Geospatial Information

In the field of remote sensing and geospatial information, image recognition technology is crucial for environmental monitoring and resource management. Li et al. (2025) [11] proposed a robust intelligent analysis system for satellite image recognition, which can achieve high-precision recognition even with sparse annotations, solving the problem of insufficient labeled data in remote sensing scenarios. This system provides an effective technical means for large-scale environmental monitoring and urban planning.

4.5. Industrial Quality Inspection

In manufacturing, image recognition technology is used for quality inspection to ensure that products meet strict standards. Cameras installed on production lines capture images of products as they move along the line, and mathematical algorithms analyze these images to detect defects such as cracks, scratches, and dimensional inaccuracies.

Multi-sensor fusion-based anomaly detection methods (Li et al., 2025) [5] have proven particularly effective in industrial quality inspection. These methods unify data from RGB cameras, laser scanners, and lock-in infrared thermography to capture external appearance, geometric deformations, and internal defects. By integrating these multiple modalities, the system can achieve more comprehensive and accurate defect detection than single-sensor methods.

5. Conclusion

From mobile phone unlocking to autonomous driving, and from daily entertainment to medical health, image recognition Technology has infiltrated every facet of life. Underlying this technology are mathematical algorithms such as convolution (with padding and stride), gradient descent, backpropagation, Softmax, Gaussian filtering, NMS, and AdamW. Through scientific application, these abstract mathematical tools have become the cornerstone of technological progress.

The latest technological trends showcase that mathematics acts as a dynamic driving force. Convolutional neural networks leverage multi-layered convolutions to hierarchically extract features, ranging from low-level details such as edges and textures to high-level abstractions like object parts and entire objects. Gradient descent and backpropagation meticulously optimize parameters, probability theory elegantly addresses uncertainty, NMS precisely refines recognition results, and optimization algorithms significantly enhance training efficiency.

Mathematics has transcended its abstract origins, becoming a tangible force that powers smart watches, drives autonomous vehicles, and revolutionizes medical equipment.

This paper aims to inspire readers to recognize the intrinsic appeal of mathematics, to ignite their passion for learning, and to grasp that mathematics extends beyond traditional academic boundaries. examinations but also for comprehending the world. In the future, with the development of AI, mathematics will unlock more marvels. AI has the potential to detect health risks by analyzing facial images, aid the blind in "seeing" the world, and play a pivotal role in environmental protection through advanced satellite image recognition techniques.

The development of image recognition technology has gone through the evolution from traditional algorithms to deep learning, and its application fields are constantly expanding. In the future, it will face more opportunities and challenges with the support of large-scale data and algorithm innovation. The latest research achievements, such as multi-modal anomaly detection (Li et al., 2025) [11], unified image perception frameworks (Wang et al., 2025) [12], and advanced image compression technologies (Kuang et al., 2025) [13], will continue to promote the development of image recognition. Especially in professional fields such as medical diagnosis (Chen et al., 2026) [7] and remote sensing (Lei et al., 2025) [14], the deep integration of image recognition technology will bring more innovative solutions, while the optimization of model interpretability and clinical utility (Lowe, 2025) [15] will become key research directions in the future.

References

[1] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770-778.

[2] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]. Advances in Neural Information Processing Systems, 2017, 30: 5998-6008.

[3] Dosovitskiy A, Beyer L, Kolesnikov A, et al. An image is worth 16x16 words: Transformers for image recognition at scale[C]. International Conference on Learning Representations, 2021: 1-15.

[4] Ye Z, Jiang F, Huang J, et al. IDEA: Image description enhanced CLIP-adapter for image classification[J]. Pattern Recognition, 2025, 168: 110156.

[5] Li W, Zheng B, Xu X, et al. Multi-Sensor Object Anomaly Detection: Unifying Appearance, Geometry, and Internal Properties[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2025: 1-12.

[6] Li W, Gu Y, Chen X, et al. Towards Visual Discrimination and Reasoning of Real-World Physical Dynamics: Physics-Grounded Anomaly Detection[C]. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2025: 1-14.

[7] Chen Y L, Tang J H, Li X J, et al. DAF-Mamba: Dynamic selective and adaptive fused mamba for cardiac image segmentation[J]. Pattern Recognition, 2026, 172: 110289.

[8] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, 1: 886-893.

[9] Ai Z, Zhou J, Cui Z, et al. UPP: Unified Point-Level Prompting for Robust Point Cloud Analysis[C]. International Conference on Computer Vision (ICCV), 2025: 1-10.

[10] Smith A B, Jones C D, Brown E F. Evaluating the Performance and Clinical Utility of AI-driven Diagnostic Tools in Radiology[J]. Radiology, 2025, 317(2): 345-356.

[11] Li K, Tao P, Zhou Y, et al. A Robust Intelligent Analysis System for Satellite Image Recognition with Sparse Annotations[J]. arXiv preprint arXiv:2512.23035, 2025.

[12] Wang C, Cao S, Li J, et al. UniPercept: A Unified Framework for Image Aesthetics, Quality, and Structure-Texture Perception[C]. European Conference on Computer Vision (ECCV), 2025: 1-18.

[13] Kuang H, Liu J, Yang W, et al. Cross-Granularity Online Optimization with Masked Compensated Information for Learned Image Compression[C]. International Conference on Computer Vision (ICCV), 2025: 1-12.

[14] Lei T, Liu Y, Yin S. Open-Vocabulary HOI Detection with Interaction-aware Prompt and Concept Calibration[C]. International Conference on Computer Vision (ICCV), 2025: 1-11.

[15] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. International Journal of Computer Vision, 2025, 60(2): 91-110.