Received: 19 August 2025 | Revised: 7 January 2026 | Accepted: 28 January 2026 | Published online: 10 March 2026

REVIEW

Artificial Intelligence and Applications
2026, Vol. 00(00) 1–17
DOI: 10.47852/bonviewAIA62027265

A Short Review on Computer Vision: Visualizing the World Through Machine

Mohammad Mehedi Hassan[1,*], Stephen Karungaru[2] and Rezaul Bashar[3]

[1] *Department of Integrated Information Systems, The Kyoto College of Graduate Studies for Informatics, Japan*

[2] *Computer Science and Mathematical Science Program, Tokushima University, Japan*

[3] *Global Circle for Scientific, Technological and Management Research, Australia*

Abstract: Computer vision is an important part of artificial intelligence. It helps machines interpret and make decisions based on visual data. As machines take on more responsibility in decision-making, vision is a key way for them to understand their surroundings. The ability for machines to see and understand the world through visual input raises the question of whether they can truly understand complex situations based on the objects and interactions around them. This paper explores the main concepts and algorithms behind computer vision, beginning with its early development. It discusses foundational techniques and how they have changed over time, including innovations that are shaping the field. The paper also looks at the limitations of these foundational concepts and how they have impacted the growth of current technologies. It includes a critical examination of present-day technologies, pointing out their challenges and shortcomings despite many improvements. Finally, the paper covers the various applications of machine vision in different fields and the promising future for further advancements in computer vision technologies.

Keywords: computer vision, deep learning, machine learning, artificial intelligence, AI applications

1. Introduction

What is artificial intelligence (AI)? To understand AI, we must first define intelligence. In short, intelligence is making beneficial decisions by perceiving the environment [1]. Humans are naturally intelligent and learn from decisions over time. Artificial refers to what is human-made rather than natural. AI is a created model or machine that makes decisions based on the environment and its objects [2]. Alan Turing [3] asked whether machines can think. Machines may not be able to think like humans, but they can make positive decisions based on feedback from the environment. Today, machines can see patterns and images and make decisions; however, it is debatable whether they have humanlike intelligence.

Human decision-making is a very complex process and depends on observing the environment through human senses, especially vision [4]. Vision helps humans learn and gain intelligence. Computer vision [5] was inspired by human vision and uses cameras to support machine decision-making. Although its development began early, computer vision continues to evolve and remains essential. Vision enables the visualization of environmental objects through light reflection [6]. Human eyes see objects when light reflects off them. Without light or minimal reflection, the human eye cannot visualize objects. Computer vision enables machines to identify objects, distance, and movement using cameras, data, and algorithms instead of biological vision. This system analyzes thousands of products or processes per minute, surpassing human ability. Computer vision began in the late 1960s to mimic the human visual system and enable robots to behave intelligently. It aimed to extract three-dimensional structures from images for full scene understanding. Early studies in the 1970s formed the foundations for many computer vision algorithms today. The next decade saw more rigorous mathematical analysis and quantitative aspects of computer vision, such as scale-space, shape inference, and contour models. By the 1990s [5], research focused on projective 3D reconstructions, camera calibration, sparse 3D reconstructions, dense stereo correspondence, and image segmentation. The field of computer graphics and computer vision has also increased, leading to the resurgence of feature-based methods and the advancement of deep learning techniques.

Computer vision is an important tool in many areas, such as medicine [7, 8], industry [9], military [10], and autonomous vehicles [11, 12]. In medicine, it helps diagnose patients, improve images for human interpretation, and support research. In industry, it is used for quality control, inspecting final products, and agricultural processes. In military applications, it helps detect enemy soldiers or vehicles and guides missiles. More advanced systems use image data to target specific areas, which reduces complexity and improves reliability. Autonomous vehicles, including submarines, land vehicles, aerial vehicles, and unmanned

***Corresponding author:** Mohammad Mehedi Hassan, Department of Integrated Information Systems, The Kyoto College of Graduate Studies for Informatics, Japan. Email: m_hassan@kcg.edu

aerial vehicles, rely on computer vision for navigation, obstacle detection, and specific tasks.

In this paper, we will discuss the basic concepts of computer vision, algorithms, and recent developments. Computer vision is a vast domain, and it is difficult to discuss all aspects of it. Here, we will discuss basic concepts such as edge detection, Histogram of Oriented Gradients (HOG), feature extraction, etc., along with their limitations. These are computationally efficient, interpretable, and applicable to small-scale problems, which are good for real-time applications, especially where resources are limited. However, they are not robust, scalable, or applicable to real-world conditions. Edge detection is more susceptible to noise, changes, and parameters. HOG is good for handling small geometric changes, illumination changes, etc., but it is not good for handling large viewpoint changes, occlusions, and high complexity scenes. Moreover, it is not scalable as it involves manual feature extraction, which is expensive, time-consuming, and inflexible. The traditional approaches are not good for real-world applications as they are not robust, scalable, or applicable to real-world conditions, etc. Here, we will also discuss future trends, ethics, etc.

2. Review Approach or Methodology

In this work, we reviewed computer vision by first examining traditional techniques and key challenges such as occlusion and scalability. Recent developments were then discussed, with emphasis on significant algorithms and developments, followed by real-life applications. Comparative analysis was also carried out to observe patterns and trends. The review then ended with future directions, which included significant discussion points, giving an overview of the evolution and current status of the subject. Literature was collected from various prominent databases such as IEEE Xplore, Scopus, Web of Science, PubMed, Google Scholar, among others, ranging from early to recent literature. Selection was based on defined criteria, focusing on books, peer-reviewed journals, and conference papers that applied computer vision methods and reported quantitative evaluations.

3. Overview of Computer Vision: Enabling Visual Perception in Machines

A computer system has visual data that helps it recognize objects such as an image of a cat. The image is processed and analyzed to identify edges, shapes, or colors. The system is trained with many images to identify patterns. It then analyzes the image and predicts the class of the image. The system shows the result by highlighting the image of the cat. To analyze an image, the system must understand what an image is. An image is a set of pixels that have different shades of color. For black-and-white images, the squares are either black or white. For grayscale images, each square is a shade of gray, ranging from black to white. Each square has a mix of three colors for color images: red, green, and blue (RGB) [13]. In short, we can say that an image is a 2-dimensional function $f(x, y)$, where x and y are spatial (plane) coordinates and h represents the amplitude (e.g., intensity or color) at each point. There are two types: analog images, represented by continuous values, and digital images, defined as discrete integers with quantized amplitudes, such as photographs and bitmaps. Computer vision processes digital images using algorithms. Digital image processing uses digital computers to process images composed of finite elements called pixels. Analog images must be converted into digital data for processing. Figure 1 explains an image with the original image, its 2D visualization of grayscale image, 3D visualization of intensity, and the matrix of pixel values of the digital image.

Computerized processes can be classified as low-, mid-, or high-level [14]. Low-level processes involve image preprocessing to reduce noise and enhance contrast. Mid-level processes include segmentation, description, and object classification. High-level processing interprets recognized objects and performs cognitive vision functions. An example is automated text analysis, which acquires an image, preprocesses it, extracts characters, describes them, and recognizes them. Digital image processing includes all these processes.

3.1. Foundational techniques

Computer vision relies on math concepts and algorithms that handle and understand visual data. Here are some basic techniques, their math principles, and examples.

3.1.1. Edge detection

Edges are the lines where colors or brightness change sharply in an image, such as the outline of a building against the sky. Detecting these edges helps computers understand the shapes of objects in an image [15]. For instance, Figure 2 shows the outline of a statue. In this image, it is clear to see the edges of the statue, the edges of the beads, and the edges of the pot put in front of the feet of the statue.

Edge detection identifies regions with high-intensity gradients using:

$$G(x, y) = \sqrt{\left(G_x^2 + G_\gamma^2\right)} \tag{1}$$

Where:

$$G_x = \partial f/\partial x, \text{ and } G_\gamma = \partial f/\partial y \tag{2}$$

In the equation,

1) G_x and G_γ are partial derivatives of the image function f(x, y) with respect to x and y, respectively. These derivatives measure how much the pixel intensity changes:

$G_x = \partial f/\partial x$: measures intensity changes in the horizontal direction (left/right).

$G_\gamma = \partial f/\partial y$: measures intensity changes in the vertical direction (up/down).

2) The gradient magnitude G(x, y) is computed by combining the horizontal and vertical gradients using:

$$G(x, y) = \sqrt{\left(G_x^2 + G_\gamma^2\right)}$$

This gives a measure of the overall change in intensity at each point. The higher the value of $G(x, y)$, the more likely that point is an edge.

3) G_x and G_γ measure image changes in x and y directions, combining them to calculate intensity at each pixel, aiding in detecting significant brightness or color changes.

Figure 1
Simple representation of a digital image and grayscale of the original image, 3D visualization of intensity, and the matrix overlay of pixel values of the image

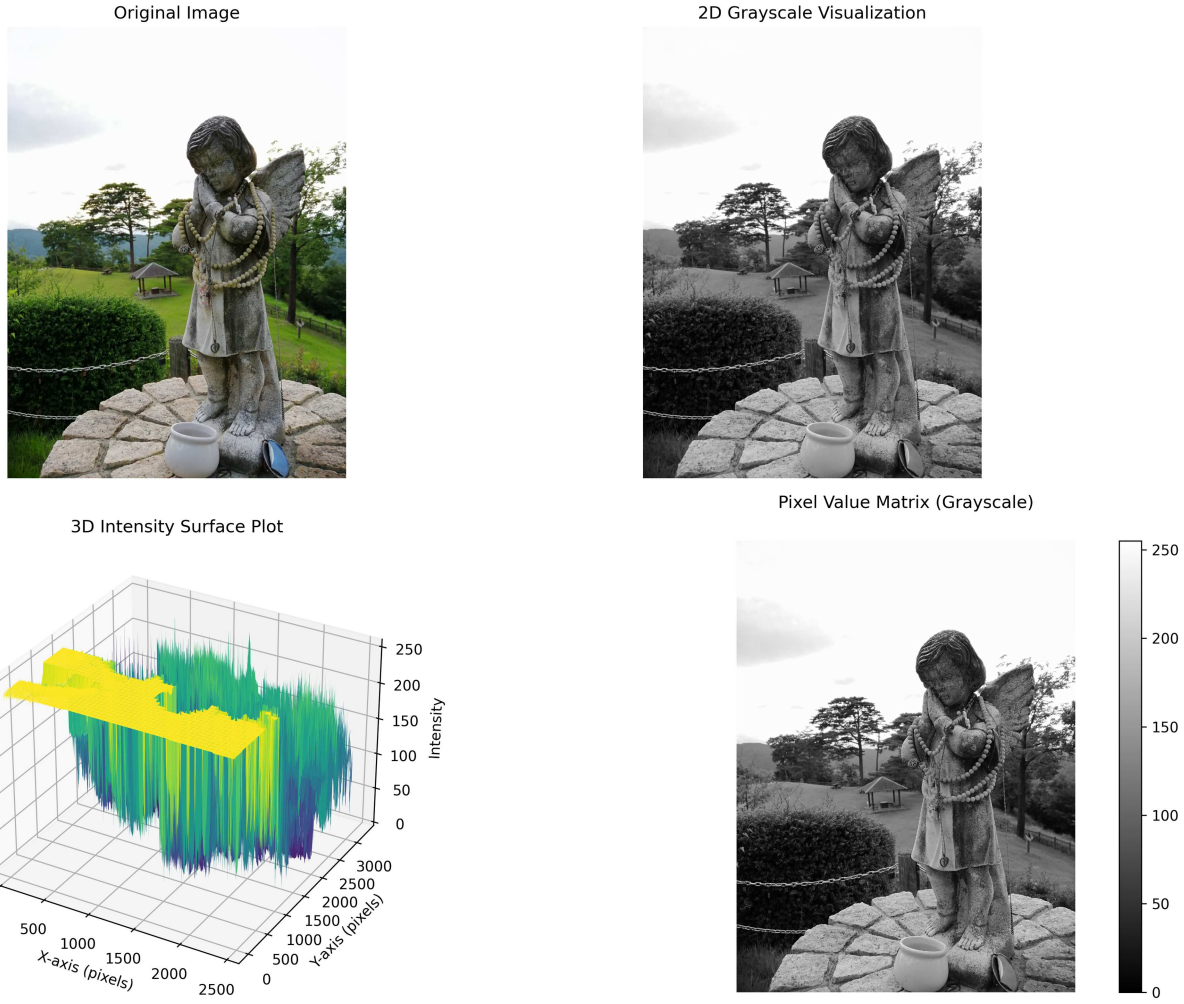


Figure 2
Visualizing the edges of a digital image by Canny edge detection

3.1.2. HOG (Histogram of Oriented Gradients)

HOG is a feature extraction technique in computer vision to detect objects in images [16]. If there is a photo of a face or human, then this feature captures the shape of objects based on pixel intensities, focusing on edges and contours. The process involves dividing the image into small regions, calculating the gradient in each region, and summarizing these gradients in a histogram.

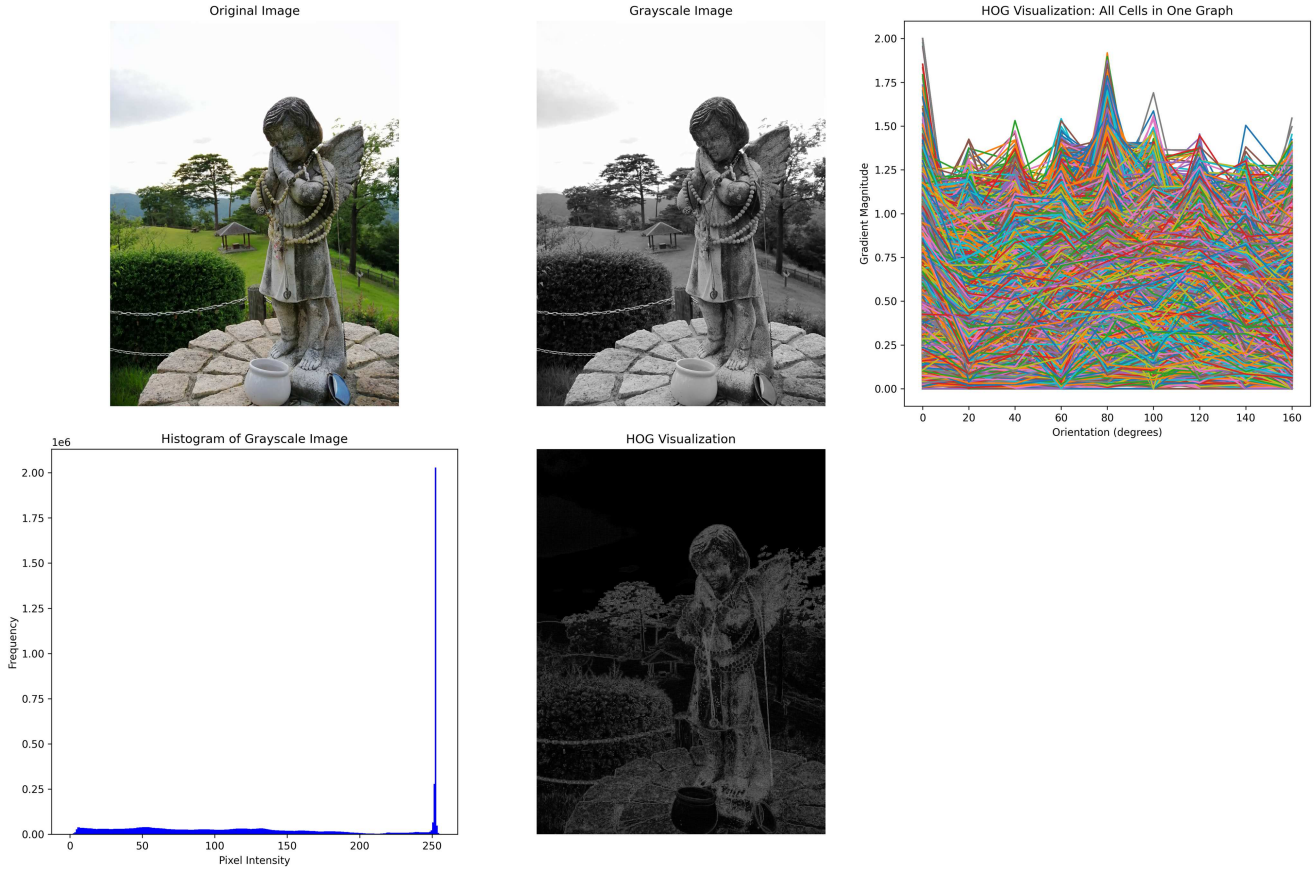
We know there is an edge if the change is large (e.g., from dark to light). In simpler terms, it tells us how quickly the image becomes brighter or darker. There is no edge if the change is small (e.g., from light to slightly darker). The gradient is calculated for each pixel using the Sobel [17] operator or other methods:

$$G_x(x, y) = f(x + 1,\ y) - f(x - 1, y) \text{ (horizontal gradient)} \quad (3)$$

$$G_\gamma(x, y) = f(x,\ y + 1) - f(x,\ y - 1) \text{ (vertical gradient)} \quad (4)$$

Where f(x, y) is the intensity value at the pixel (x, y). G_x and G_y represent the horizontal and vertical changes in intensity.

Figure 3
Network teaching representation of HOG features showing the original and grayscale images, gradient orientation and magnitude of image cells, and the grayscale histogram with pixel intensity versus frequency



Once we have the gradients, we calculate the magnitude and direction of the gradients for each pixel:

$$\text{Magnitude}\,(x,y) = \sqrt{G_x\,(x,y)^2 + G_y\,(x,y)^2} \tag{5}$$

$$\text{Orientation}\,(x,y) = \text{atan2}\left(G_y\,(x,y)\,, G_x\,(x,y)\right) \tag{6}$$

In a face photo, HOG would work by calculating the brightness of each pixel, dividing the image into cells, typically 8 × 8 pixels. Then, compute a histogram of the edge directions in each section. The histogram will have bins corresponding to different angles (e.g., 0°, 45°, 90°, etc.), and each pixel contributes its gradient magnitude to the appropriate bin based on its orientation. To make the descriptor robust to changes in lighting, each cell's histogram is normalized with respect to its neighboring cells (this is called block normalization). A block typically consists of multiple cells (e.g., 2 × 2 cells). Figure 3 illustrates the HOG features.

Normalization ensures that variations in lighting don't affect object detection.

Once histograms from all cells and blocks are combined into a feature vector, a machine learning model such as a support vector machine (SVM) [18] uses it to recognize whether an image contains a face. HOG is effective for object detection because it captures key visual features, is robust to lighting changes and small distortions, and performs well in detecting pedestrians, vehicles, and faces in images or videos.

3.1.3. Feature extraction

Feature extraction is a key computer vision operation that enables a computer to identify objects in an image. Features are unique image patterns [19] such as corners, lines, or textures that computers use for recognition. For instance, table corners or zebra stripes assist in identifying objects, just as hairstyles or glasses assist in identifying humans. Features reduce complexity by avoiding pixel-by-pixel comparison and instead rely on distinctive features. A corner represents the intersection of object faces, and a line represents an edge. A common approach is gradient-based feature extraction, which detects intensity changes. Harris corner detection [20] is a widely used method that identifies corners by estimating a response value at each image position. The equation for Harris corner detection is:

$$R = \det(M) - k(trace(M))^2 \tag{7}$$

Where:

1) M is a 2 × 2 matrix of image gradients at a particular point in the image.
2) $det(M)$ refers to the determinant of the matrix M, which provides information about the area's cornerness.
3) $trace(M)$ is the sum of the diagonal elements of the matrix M, which indicates how much the pixel values change in different directions. A high trace value means there's a significant intensity change.
4) k is a sensitivity factor, typically a small constant (e.g., k = 0.04) that helps adjust the calculation. A smaller k results

in more points being detected as corners, while a larger k makes the detection stricter, only identifying strong corners.

5) The final value R is used to determine whether a pixel is a corner. A high value of R indicates that the point is a corner, while a lower value suggests the point is not.
6) The matrix M is built by means of horizontal and vertical direction gradients (intensity differences) (typically computed by means of Sobel operators). The matrix allows us to measure the intensity and direction of the intensity change at a point.

$$M = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix} \tag{8}$$

Where:

I_x is the gradient in the x-direction (horizontal), and I_y is the gradient in the y-direction (vertical). Consider a photo of a desk with a coffee mug. The mug edges where it meets the desk form corners. The algorithm examines gradients to find strong pixel intensity changes. The line where the desk meets the mug is an edge, another detectable feature. Desk patterns, such as wood grain, also help distinguish objects. Using methods like Harris corner detection, the computer detects corners and edges to understand the image layout and recognize the desk and mug in other images.

3.2. Challenges in computer vision

Though edge detection, HOG, and feature extraction are some of the most commonly used techniques in computer vision, they are also limited. These techniques are used for the detection of important features of the image. However, real-world scenarios are affected by some challenges that make the detection of features more difficult. Among the challenges, the issues of scalability and occlusion are of great concern.

3.2.1. Occlusion

Occlusion [21] (Figure 4) is when one portion of the object is hidden from view by another object. This is a very common phenomenon when we are dealing with real-world vision problems, as objects are often partially visible due to occlusion. Occlusion is a problem that can affect the algorithms as it causes incomplete features, incorrect gradients, and tracking problems. Missing or truncated corners and edges make recognition difficult. Occlusion can also distort gradients, leading to false edge detection. For HOG, occluded regions produce incomplete gradients, reducing recognition accuracy. In dynamic scenes, occlusion can confuse tracking algorithms and cause object loss.

The visibility of a pixel or object can be modeled as a binary mask $V(x, y)$, where:

$$V(x, y) = \begin{cases} 1 & if\ pixel(x, y)\ is\ visible, \\ 0 & if\ pixel(x, y)\ is\ occluded. \end{cases} \tag{9}$$

When an object $O(x, y)$ is occluded, only the visible parts are captured in image $I(x, y)$:

$$I(x, y) = V(x, y) \cdot O(x, y) \tag{10}$$

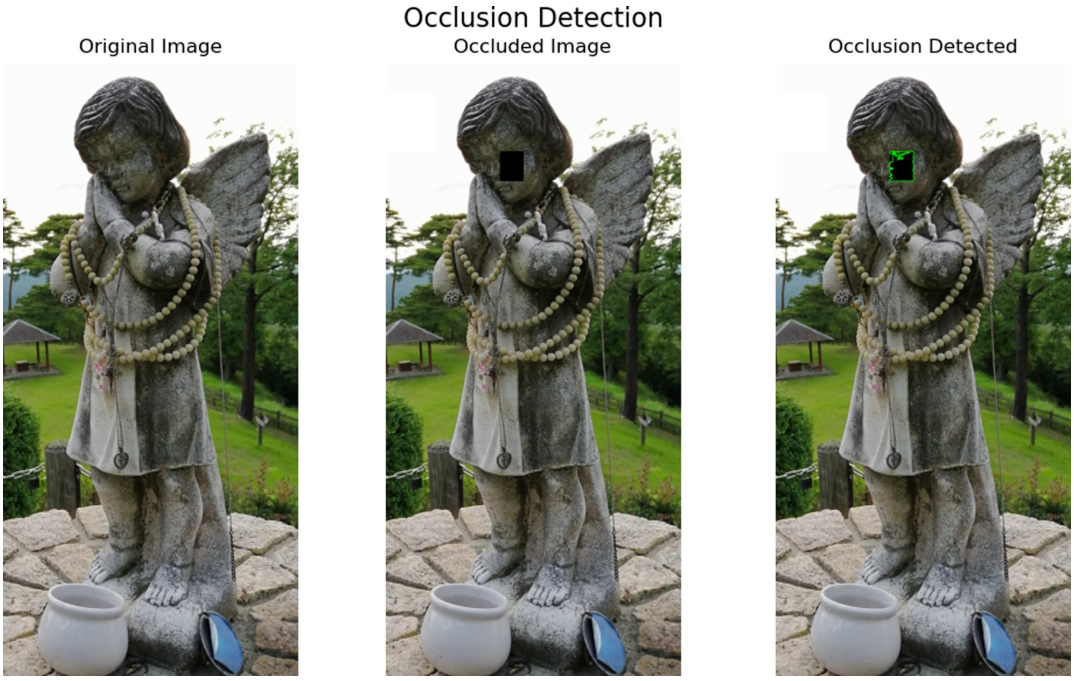
This masking operation leads to partial or missing data in the image. For example:

1) In edge detection, gradients $\nabla I(x, y)$ in occluded regions may be incomplete or distorted.
2) In feature extraction, feature descriptors such as Scale-Invariant Feature Transform (SIFT) or HOG rely on the visible region and cannot account for missing parts.

When detecting a car on a busy street, other vehicles may block parts of it, so edge detection or HOG captures only visible parts, leading to incomplete recognition or misclassification.

Occlusion affects object recognition by creating incomplete feature sets. Consider an object O represented by a set of features

Figure 4
An occlusion was created and detected. In this image, there is an original image, followed by an occluded image, and occlusion detection in that image



($\{f_1, f_2, \ldots, f_n\}$). When some features are occluded, only a subset ($S \subset \{f_1, f_2, \ldots, f_n\}$) is available.

The recognition task then becomes one of matching S to the known feature set under uncertainty:

$$\text{Match:}colon\ arg,\ \max_i\ \text{Similarity}\,(S, O_i) \tag{11}$$

where O_i represents a known object. The similarity score may drop below the recognition threshold if the occlusion is significant. In HOG, occlusion can disrupt the computation of gradient histograms. If a person's torso is occluded, gradients in those regions will not contribute to the histogram. Mathematically, if a gradient $g(x, y)$ is part of the occluded region $(V(x, y) = 0)$, it is excluded from the histogram calculation.

3.2.2. Scalability

Scalability [22] refers to an algorithm's ability to handle images of varying size, resolution, and complexity. Computer vision methods must perform well across different image scales and resolutions.

In the real world, objects are of varying sizes depending on the distance or viewing angle, and hence, feature extraction techniques such as HOG are difficult to implement. For high-resolution images, more information is available, but the computation is more intensive, and hence, the edge detection is slow or inaccurate. For low-resolution images, the information is reduced, and hence the feature extraction is poor. As the size of the image increases, the computation required for edge detection, HOG, and feature extraction increases significantly, thereby increasing the cost and the processing time, which is critical in applications such as autonomous driving or surveillance. The multi-scale problem is another challenge, as the HOG-based techniques assume that the objects are of similar sizes.

In real-world images, the same object O can appear at different scales s:

$$O_{s(x,\,y)} = O\,(sx,\ sy) \tag{12}$$

where $s > 1$ represents upscaling, and $s < 1$ represents downscaling.

When detecting an object at different scales, the feature space F_s extracted from O_s changes with s. Features such as edges or keypoints may disappear or change orientation at extreme scales.

To address this, multi-scale processing involves creating an image pyramid:

$$I_{\{s\}(x,\,y)} = I\left(\frac{x}{s}, \frac{y}{s}\right) \tag{13}$$

where I_s represents the image scaled by s. Algorithms then process each level of the pyramid to find scale-invariant features. For an image of size $H \times W$, the computational cost of many vision algorithms increases with the number of pixels HW. For example:

1) Edge Detection: Computing gradients $\nabla I(x, y)$ involves operations for each pixel, leading to $O(HW)$ complexity.
2) HOG: Constructing histograms across overlapping blocks increases complexity to approximately $O(HW.B)$, where B is the number of blocks.
3) Feature extraction and matching also need to be scale-invariant. For example, the similarity between two objects O and O' can be computed using a transformation-invariant distance d:

$$d(O,\ O') = \min_T\ \text{Distance}\,(T(O), O') \tag{14}$$

where T is a transformation (e.g., scaling, rotation).

Without proper scaling mechanisms, the distance d increases as the object size changes, leading to poor recognition results.

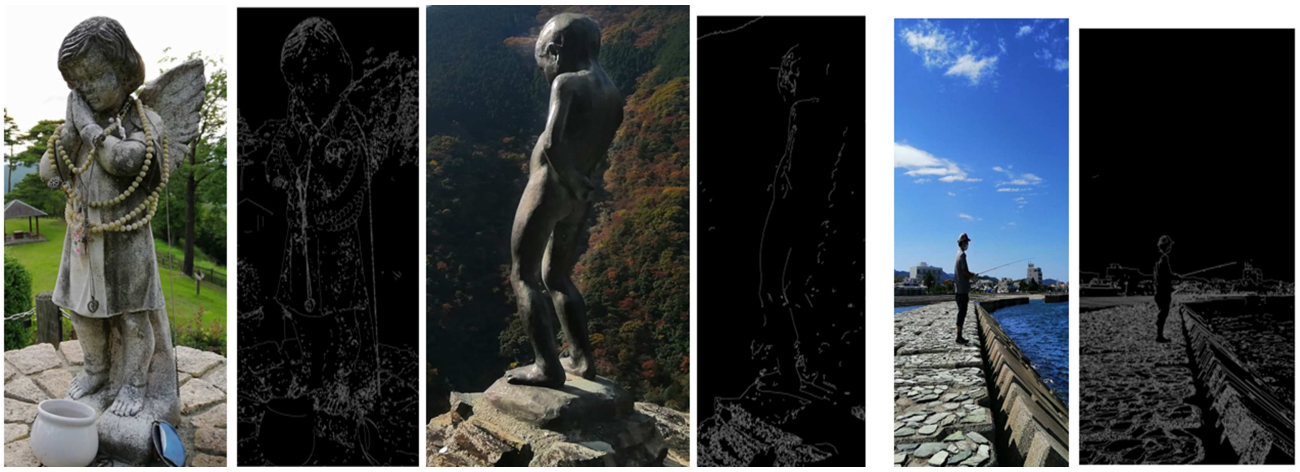
A HOG detector may work well when a person is close and large in the image but may fail when the person is far away and the features are small. Similarly, large or high-resolution images increase computational cost and slow processing. Figure 5 shows the digital image and its edges.

3.2.3. Additional challenges in foundational techniques

Noise sensitivity: Sobel or Canny edge [23] detection operators are sensitive to noise, which may cause incorrect or missed edges.

Weak boundaries: It is difficult to detect objects that have fuzzy or low-contrast boundaries.

Figure 5
Simple representation of a digital image



Restricted to visible features: Since the HOG operator is based on the detection of edge gradients, it is difficult to detect blurred or shadowed objects.

Pose variance sensitivity: The HOG operator may fail to detect patterns when the object is skewed, thereby reducing the accuracy.

Feature mismatches: It is difficult to perform feature extraction on complex scenes.

Feature diversity: Different objects may require different features, and hence, it is difficult to develop a universal feature extraction mechanism.

Occlusion and scalability are major issues for basic computer vision techniques such as edge detection, HOG, and feature extraction. Occlusion hides key features, making object detection difficult. Scalability involves handling varying image sizes and resolutions, which is computationally expensive.

4. Evolution of Computer Vision: Key Algorithms and Advances

Classical approaches such as SIFT, HOG, and Haar cascades were based on explicit coding and handcrafted features. Classical approaches had many disadvantages, such as being task-specific, incapable of handling complex patterns, computationally expensive for big data, sensitive to noise, and lacking end-to-end learning capabilities. Handcrafted features were usually designed for specific tasks or databases, which were not transferable. These approaches were also poor performers for tasks such as object detection or semantic segmentation, which require contextual understanding of images. Another disadvantage of handcrafted features was that they were sensitive to changes in illumination, rotation, occlusion, and scale, which made it difficult to recognize images.

Deep learning techniques were developed to solve the limitations of traditional approaches in computer vision and other fields. The era of neural networks began in the 1943–1980s [24] with the foundations being laid by McCulloch and Pitts. The perceptron was developed by Rosenblatt in 1958 [25], and the multi-layer perceptron was suggested in 1998 [26]. Convolutional Neural Networks (CNNs) [27] were initially proposed in 1998 by Yann LeCun but were computationally hampered and faced the vanishing gradient problem. AlexNet [28] and ImageNet [29] were proposed in 2012, with revolutionary performance on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) competition. Modern architectures introduced deeper and leaner networks such as VGGNet [30], ResNet [31], Inception [32], and EfficientNet [33], and advancements in object detection and segmentation. In 2021, Vision Transformers (ViTs) [34] and Generative AI [35] were adopted in computer vision applications. Deep learning [36] is a strong technique for the automatic selection of task-related features from data. It is a way of representing complex high-dimensional data, where low-level features are learned in shallow layers, while higher-level features are learned in deeper layers. Deep learning is particularly effective for large-scale image datasets, such as ImageNet, where AlexNet outperformed traditional computer vision techniques. The efficiency of training deep learning models is due to the advent of hardware, backpropagation, rectified linear unit (ReLU), dropout, and batch normalization. Deep learning is a paradigm shift from traditional computer vision techniques from a manual to a data-driven approach, owing to the advances in data, hardware, and algorithms.

4.1. Convolutional Neural Networks (CNNs)

CNNs are the backbone of most deep learning computer vision methods. They process grid-structured data such as images and are effective for classification, detection, and segmentation by learning spatial feature hierarchies. They extract edge, texture, and shape features using learnable filters. Key constituents of CNNs include:

1) Convolutional Layer: The convolutional layer is the core unit of a CNN. It applies convolution operations between the input and learnable filters to extract local features such as edges, textures, and patterns. Filters slide over the input, computing dot products to produce feature maps, and multiple filters learn different features. A convolution operation between an input X (e.g., an image) and a kernel (or filter) W produces a feature map F:

$$F(i,j) = \sum_{m=1}^{M} \sum_{n=1}^{N} W(m,n) \cdot X(i+m-1, j+n-1) + b \tag{15}$$

Here:
X: Input matrix (image of size $H \times W$)
W: Filter (of size $M \times N$)
b: Bias term
i, j: Coordinates of the output feature map

The convolution slides the filter over the input to compute dot products, capturing local patterns such as edges or textures. For instance, a 3×3 filter W is applied to a 5×5 image X. The result is a 3×3 feature map after valid padding.

2) Pooling Layer: The pooling layer reduces the spatial dimensions of the feature maps while retaining a lot of information. The operation of the pooling layer is the summarization of the values within a certain region. The most common pooling layers are Max Pooling and Average Pooling.

Given a window size $k \times k$, for max pooling:

$$P(i,j) = \max_{m,n \in k \times k} X(i+m, j+n) \tag{16}$$

The proposed solution aims to decrease computational complexity, enhance network resilience to translations or distortions, and prevent overfitting by reducing the number of parameters.

3) Activation Layer: The activation layer applies a non linear function to feature maps. Without nonlinearity, the model becomes a linear system and cannot learn complex patterns. Since convolutional and fully connected layers are linear operations, activation layers are essential for representing complex relationships.

Common Activation Functions:

a. ReLU (Rectified Linear Unit): $f(x) = max(0, x)$, the graph is a piecewise linear function with zero for negative inputs and linear for positive inputs, offering efficient computation, reducing vanishing gradient, and sparse activation.
b. Sigmoid: $f(x) = 1/(1 + e^{-x})$, the S-shaped graph offers a probabilities interpretation and smooth gradients but has a vanishing gradient problem, slowing training, and is not zero-centered.
c. Tanh: $f(x) = (e^{x} - e^{-x}) / (e^{x} + e^{-x})$, the S-shaped curve offers advantages such as zero-centered optimization and a steeper

gradient than a sigmoid, but also suffers from the vanishing gradient problem for large $|x|$.

d. The activation layer is crucial for introducing the nonlinear capabilities of CNNs, enabling them to learn and model complex data relationships.

e. Fully Connected Layer: After convolution and pooling, the learned feature maps are flattened and fed through one or more fully connected layers:

$$y = W \cdot x + b \tag{17}$$

Here, x is the flattened input, W is a weight matrix, and b is a bias vector.

The fully connected layer connects neurons in one layer to the next, enabling predictions based on learned features. It aggregates features from previous layers and maps them to output classes or regression values at the network's end.

4) Loss Function and Backpropagation: The loss function and backpropagation enable neural network training by optimizing parameters to reduce prediction errors. The loss function measures the difference between predicted outputs and true labels to guide optimization. A few popular loss functions include mean squared error for regression tasks, which is:

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2 \tag{18}$$

and binary cross-entropy for binary classification tasks:

$$L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y_i}) + (1 - y_i)\log(1 - \hat{y_i})] \tag{19}$$

The loss function can also include regularization such as *L1* and *L2* to avoid overfitting. The objective is to minimize the loss function, and this is done using methods such as gradient descent. The gradients of the loss function are calculated using the chain rule in Backpropagation. The gradients are propagated backward through the layers of the network. For example, for a weight W at the output layer, the update rule is:

$$W := W - \eta\frac{\partial L}{\partial W} \tag{20}$$

where η is the learning rate. The major difficulties in backpropagation include the vanishing gradients, where gradients get too small, and the exploding gradients, where gradients get too large, disrupting the training process. The ReLU and gradient clipping techniques address these difficulties. The loss functions and backpropagation help neural networks learn complex patterns and improve their performance in image recognition and classification.

5) Output Layer: The output layer produces final predictions from extracted features. For classification, a softmax function converts logits into probabilities. Mathematically, for C classes of output, the probability for class j is given by the softmax formula:

$$\hat{y_j} = \frac{e^{z_j}}{\sum_{i=1}^{C} e^{z_i}} \tag{21}$$

where $z_j = W_j . h + b_j$ is the raw score for class j, h is the output from the last hidden layer, W_j is the weight for class j, and b_j is the bias. The predicted class is the one with the highest probability, which is computed as $\hat{y} = arg\ max_j\ (\hat{y}_j)$. For example, in a three-class classification problem, the output layer produces a vector of probabilities, and the class with the highest probability is selected as the predicted label. In the example of a CNN for image classification into three classes (cat, dog, and bird), the output layer has three neurons, one for each class. After processing an image, the network produces a vector of features, $h = [2.5, 1.2, -0.5]$, which is passed through the output layer. The raw scores (logits) for each class are computed as:

$$[z_1 = 2.11,\ \ z_2 = -0.01,\ \ z_3 = -0.9]$$

These logits are then converted into probabilities using the softmax function:

$$[\hat{y_1} \approx 0.855,\ \ \hat{y_2} \approx 0.103,\ \ \hat{y_3} \approx 0.043]$$

The predicted class has the highest probability, class "cat," with a probability of *0.855*. Thus, the CNN classifies the input image as "cat."

CNNs share parameters, reducing model size, and detect features regardless of position. They learn low-level features in early layers and complex patterns in deeper layers. Advantages include sparse connectivity, weight sharing, and reduced computation through pooling.

4.2. Object detection

Object detection involves the identification and localization of objects in the image by classifying the objects and determining the bounding box for the object. The most commonly used object detection techniques include the R-CNN [37] family and the YOLO [38] family. The R-CNN family uses CNN for classifying the regions, while the YOLO family uses the entire image for object detection by dividing the image into a grid. The YOLO loss function is composed of three components:

$$L = L_{coord} + L_{conf} + L_{class} \tag{22}$$

where L_{coord} is the loss for bounding box regression, L_{conf} is the loss for object confidence (whether an object is present in the predicted box), and L_{class} is the loss for classification (classifying the object within the bounding box).

Object detection has many real-world applications, such as the use of pedestrian detection in autonomous vehicles, where it is used in real time, and the use of anomaly detection in surveillance systems. In general, object detection has many automation and safety-related applications.

4.3. Semantic segmentation

In semantic segmentation [39], the class of each pixel in the image is specified. Thus, all the pixels of the same object are of the same class. Unlike object detection, where the bounding box is used for labeling, all the pixels are labeled for fine-grained scene understanding. It is typically achieved using CNNs with an encoder–decoder structure. The encoder is utilized to extract the features, and the decoder is utilized to upsample the features for pixel-wise prediction. The most commonly utilized model is the U-Net [40] model.

Mathematically, the goal is to assign a label to each pixel in the image. Let y_i represent the true label of the i-th pixel, and $\hat{y}_i$ represent the predicted label. The loss function used for training semantic segmentation models is typically the cross-entropy loss,

which measures the difference between the predicted probability distribution and the true distribution:

$$L = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_i^c \log(\hat{y}_i^c) \tag{23}$$

Where:

N is the number of pixels,

C is the number of classes,

y_i^c is the ground truth label for class c at pixel I (1 if the pixel belongs to class c, 0 otherwise), and

$\hat{y}_i^c$ is the predicted probability of class c at pixel i.

For multi-class segmentation, the network produces a probability for each pixel and applies cross-entropy loss to penalize incorrect predictions. Semantic segmentation is used in medical imaging, autonomous driving, and remote sensing. By dividing an image into meaningful regions, it provides detailed scene understanding for high-level image analysis.

4.4. Generative models

Generative models learn to represent the data distribution in order to generate new data similar to the original data. Unlike discriminative models, which predict the label, generative models learn the joint probability distribution $P(x,y)$ to generate data. The most common generative models are Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Gaussian Mixture Models (GMMs).

4.4.1. Generative Adversarial Networks (GANs)

These two neural networks used in GANs are called discriminator and generator. The discriminator neural network is used for distinguishing real and fake data. On the other hand, the generator neural network is used for generating data. The generator's objective is to trick the discriminator. This can also be viewed as a minimax game. The goal function of GAN is:

$$\min_G \max_D E_{x\sim p_{dt}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{24}$$

Where:

$G(z)$ is the generated data from random noise z,

$D(x)$ is the discriminator's estimate of the probability that x is real,

$p_{data}(x)$ is the real data distribution, and

$p_z(z)$ is the distribution of the latent noise input to the generator.

4.4.2. Variational Autoencoders (VAEs)

VAEs are another class of generative models that combine variational inference and autoencoders. The model learns an approximate posterior distribution over the latent code z for data x, and it is trained for maximizing the Evidence Lower Bound (ELBO) on the log-likelihood of the data. The ELBO can be expressed as:

$$L(\theta, \phi; x) = E_{q_\phi(z|x)}[\log p_\theta(x|z)] - D_{KL}\left(q_\phi(z|x)\,|p(z)\right) \tag{25}$$

Where:

$p_\theta(x|z)$ is the likelihood of the data given the latent variable,

$q_\Phi(z|x)$ is the approximate posterior distribution (the encoder),

$p(z)$ is the prior distribution on the latent variables, and

D_{KL} is the Kullback–Leibler divergence between the approximate posterior and the prior.

VAEs learn to generate data by sampling from the learned latent space and decoding the samples into data points that resemble the original data distribution.

4.4.3. Gaussian Mixture Models (GMMs)

GMMs assume data is drawn from a mixture of Gaussian distributions. Each component is defined by a mean and covariance, and the final distribution is a weighted sum of these components. The likelihood of the data is represented as:

$$p(x) = \sum_{i=1}^{K} \pi_i N(x|\mu_i, \Sigma_i) \tag{26}$$

Where:

π_i is the weight of the i-th Gaussian component,

$(N(x|\mu_i, \Sigma_i))$ is the Gaussian distribution with mean μ_i and covariance Σ_i, and

K is the number of Gaussian components.

GMMs are used in generative modeling, clustering, and density estimation. GMMs can generate new samples by sampling from the fitted mixture distribution.

Generative models are used in various domains, such as realistic image generation (GANs, VAEs), synthetic data creation for training, anomaly detection, drug discovery, and text or music creation. Their ability to generate realistic data makes them useful when labeled data is limited or creative generation is required.

4.5. Vision Transformers (ViTs)

"Vision Transformers" is a class of deep learning architectures in computer vision that make use of the transformer architectures, which were first introduced in the field of natural language processing (NLP). Unlike CNNs, which rely on convolution and pooling, ViTs divide an image into fixed-size, non-overlapping patches and treat each patch as a token.

The image is first divided into patches, such as 16×16 or 32×32 pixels. These patches are linearly projected and combined with positional encodings, and the resulting sequence is passed through a transformer.

Mathematically, let the input image be of size $H \times W \times C$, where H is the height, W is the width, and C is the number of channels (e.g., 3 for RGB images). The image is divided into N patches, where $N = (H \times W) / P^2$ and P is the size of each patch. Each patch x_i of size $P \times P \times C$ is flattened and linearly projected to a vector of size D, where D is the embedding dimension. The patch embeddings are then combined with positional encodings PE_i to form the input sequence:

$$z_0 = [x_1 + PE_1, x_2 + PE_2, \ldots, x_N + PE_N] \tag{27}$$

This sequence is passed through the transformer encoder, which includes multi-head self-attention and feed-forward layers. The final output is passed through the classification or regression head. In the transformer encoder, the multi-head self-attention mechanism allows the model to weigh the importance of the patches. The feed-forward layers are also applied.

The primary advantage of ViTs over CNNs is their ability, through self-attention, to effectively model long-range dependencies between distant image regions. This makes ViTs particularly effective for tasks such as large-scale image classification, object detection, segmentation, etc. The primary disadvantages of ViTs are their requirement for large amounts of data and heavy

computational requirements, owing to their quadratic complexity. ViTs are also different from CNNs in their treatment of spatial relationships. ViTs split images into patches of a certain size and then flatten them before processing them as sequences using self-attention. Self-attention allows ViTs to effectively model global relationships from an early stage, whereas CNNs need many layers to achieve a large receptive field. While self-attention reduces the computational cost of ViTs, their lack of strong inductive biases such as spatial locality makes them data-inefficient. This results in poor performance on small datasets and scalability issues with large images.

ViTs have demonstrated superior performance to CNNs on a range of image classification tasks, particularly those with large-scale training or pretraining datasets. The ability of ViTs to effectively model long-range dependencies and handle large resolutions makes them particularly effective for a range of computer vision tasks.

5. Traditional Methods versus Deep Learning–Based Methods

Conventional computer vision techniques such as SIFT, HOG, Speeded-Up Robust Features (SURF), etc., are used for simple tasks but are more sensitive to noise and changes. These techniques perform well on small-scale data under specific conditions but show poor performance on large-scale and varying data. In comparison with these techniques, deep learning techniques such as CNN are more efficient and robust for dealing with complex tasks such as classification, detection, and segmentation. Deep learning techniques show robust performance on large-scale and varying data and are invariant to scale, orientation, and noise but demand more data and computational resources. Despite these requirements, deep learning techniques show better performance compared to traditional techniques, especially under varying conditions. Table 1 shows traditional methods versus deep learning–based methods.

6. Computer Vision Applications

6.1. Medical imaging

In medical images, computer vision has a significant role to play in the automation and precise diagnosis of images. Object detection and image segmentation are used to classify images such as X-rays, MRIs, CT scans, and ultrasounds. For instance, deep learning helps in the diagnosis of diseases such as cancer and other brain-related diseases by identifying tumors and lesions in the images. CNNs are best used to detect hidden patterns in images.

6.2. Autonomous driving and robotics

In self-driving cars, computer vision plays a central role in enabling cars to perceive their surroundings, make decisions, and navigate around safely. Hardware takes a combination of cameras, LiDAR, and radar sensors to detect objects, people, and signs and perform lane-keeping and obstacle avoidance. For instance,

Table 1
Comparison between traditional methods and deep learning–based method

Aspect	Traditional methods	Deep learning–based methods
Feature Extraction	Hand-tuned attributes such as SIFT (Scale-Invariant Feature Transform) and HOG (Histogram of Oriented Gradients), where hand selection and feature tuning are required using knowledge from the domain	Features are automatically learned from data through neural network architectures, such as CNNs (Convolutional Neural Networks), which extract relevant features directly from images without human intervention
Performance	It performs well for easier tasks with smaller dataset sizes, where problems could be conveniently specified using rules or heuristics. It doesn't perform for complex tasks and requires a lot of handcrafted feature engineering	Deep learning methods such as CNNs are less sensitive to viewpoint, scale, and noise, as they generalize from large and new data. Optimal architectures and data augmentation improve robustness
Robustness	The traditional approaches are sensitive to perspective, scale, lighting, and noise. Traditional approaches make use of fixed features, which may not generalize to other states	Deep learning techniques require large amounts of labeled data to train models effectively. Numerous data points are required to obtain sophisticated representations and relations, especially for applications such as image classification and object detection
Data Requirements	The traditional approaches must utilize less data as they employ handcrafted features, which can utilize smaller sets of data. The traditional approaches perform well in low-data tasks	Deep learning–based methods require large amounts of labeled data to effectively train models. A significant amount of data is needed to learn complex representations and relationships, especially for tasks such as image classification and object detection
Examples	Examples include edge detection (Canny, Sobel), feature matching (SIFT, ORB), and histogram-based methods (e.g., color histograms for object recognition). These are applied to simpler tasks such as basic image processing and feature-based matching	Examples include image classification (CNNs), object detection (YOLO, Faster R-CNN), and semantic segmentation (U-Net). They are used for more sophisticated tasks such as autonomous driving, facial recognition, and medical imaging

object detection algorithms such as YOLO (You Only Look Once) recognize other automobiles, pedestrians, and cyclists in real time. Similarly, in robotics, computer vision enables robots to interact with their environment, either for industrial purposes (e.g., factory production lines) or for use in applications such as robotic surgery or autonomous drones.

6.3. Sports and other areas

Computer vision is transforming the way sports analysis and management are done. Computer vision technologies such as player tracking, motion detection, and event recognition are used for player tracking, team strategy analysis, and performance evaluation. For instance, in football, computer vision is used for player tracking and ball tracking in real time, generating heat maps, fatigue analysis, and team strategy. In tennis, AI is used for line calls, ball tracking, and stroke motion analysis. Computer vision improves fan experience and enhances performance.

7. Comparative Findings

Canny remains a computationally efficient and interpretable baseline, but its accuracy is limited on modern benchmarks. For example, on art conservation imagery, Canny achieved an F1 score of 0.4000 with a high Structural Similarity Index Measure (SSIM) of 0.8833, indicating good structural fidelity but poor edge precision due to false positives in textured regions and missed boundaries in low-contrast areas [41]. Similarly, in BSDS500 evaluations (Table 2), Canny is consistently outperformed by learning-based and deep learning methods. Building on this, Structured Forests [42] improved upon traditional detectors by using patch-based structured learning, enabling real-time performance with strong accuracy on the BSDS500 and NYU Depth datasets. This method achieved near state-of-the-art results at a fraction of the computational cost, establishing itself as one of the most practical learning-based alternatives. Deep learning methods further pushed performance boundaries. Holistically-nested edge detection (HED), for instance, achieved an ODS F-score of 0.782 by leveraging deep supervision and multi-scale contextual features [43]. More recent innovations, such as Neural Homography Propagation (NHP) networks, even surpassed human-level accuracy (human ≈ 0.803) with an ODS of 0.818 [44]. The current state of the art, NBED, advances this further, reaching an ODS of 0.838 through an advanced encoder–decoder framework [45]. Despite these advances, significant challenges remain in detecting edges in low-contrast and heavily occluded regions. Moreover, computational efficiency continues to be a bottleneck. For instance, while Canny executes in 0.808 s per image, deep detectors such as DexiNed require nearly 40 s on the same dataset [41]. This highlights an important trade-off: lightweight classical detectors remain attractive for resource-constrained environments, whereas deep learning models dominate in accuracy for research and industrial applications.

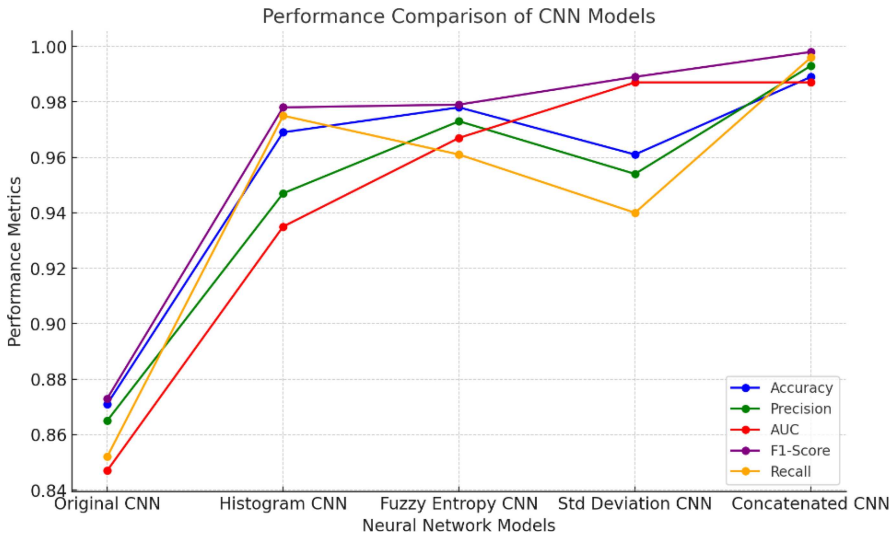
In parallel to edge detection, feature descriptors such as HOG, SIFT, and SURF have played a central role in traditional vision pipelines. HOG, in particular, is effective for capturing local edge information [46], although it is sensitive to illumination changes and background clutter. Research has consistently demonstrated that HOG can be improved through both descriptor enhancements and algorithmic optimizations. For instance, a 2015 study using night-vision data found that a Modified HOG descriptor boosted detection rates by approximately 5.35% at 10^{-4} False Positives Per Window (FPPW) on nocturnal pedestrian imagery, and 2.22% improvement on daytime (INRIA) data [47]. Similarly, enhancements such as "Circle HOG" combined with Enhanced Histogram of Oriented Gradients (EHOG) delivered around 4.5% higher detection accuracy over the original HOG on INRIA benchmarks [48]. Beyond descriptor tuning, integral channel feature methods (ChnFtrs) outperformed HOG substantially, achieving 86% detection versus 77% for HOG at 1 false positive per image on INRIA full images, and 60% versus 50% on Caltech Pedestrian benchmarks [49]. Finally, the use of a lookup table with integral images speeds up HOG computation by 5–10×, enabling efficient real-time detection [50]. Collectively, these findings highlight both the robustness of HOG and practical avenues for improvement across different operating contexts. A similar pattern of limitations is seen in holistic face-recognition methods. Traditional approaches such as Principal Component Analysis (PCA)-based Eigenfaces are particularly vulnerable to partial occlusion, since occlusion corrupts the global feature representation. Empirical results confirm that recognition accuracy for PCA can drop sharply as occlusion increases. For example, on the FRGC v2.0 dataset, PCA recognition accuracy fell from approximately 86% at 30% occlusion to about 61% at 40% occlusion [51]. This underscores the need for more robust methods, such as local feature–based or occlusion-aware approaches. Scalability to large datasets and robustness against occlusion thus remain persistent challenges in handcrafted methods.

Deep learning greatly changed this situation. CNNs revolutionized feature extraction as they learned hierarchical representations directly from the data. Unlike HOG or SIFT, CNNs are capable of learning variations specific to datasets and more

Table 2
Comparison between traditional methods and deep learning–based method

Method	Type	ODS F-score	Key highlights
Canny (1986)	Traditional	– (baseline)	Classic, efficient baseline, but low accuracy on BSDS500
HED (2015)	Deep Learning	0.782	Introduced deep supervision and multi-scale features; large improvement over handcrafted detectors
Structured Forests	Learning-Based	Near SOTA (no exact value)	Patch-based structured learning; real-time, efficient, and competitive accuracy
NHP-based Deep Network	Deep Learning	0.818	Surpassed human-level performance (human ~ 0.803)
NBED (2024)	Deep Learning	0.838	State-of-the-art encoder–decoder model on BSDS500

Figure 6
Comparison of different CNN models' performance



complex patterns since they can surmount handcrafted features' robustness and scalability limitations. For instance, AlexNet reduced the ImageNet classification error from 26.2% relative to conventional methods to 15.3% [52]. High-performance detectors such as Faster R-CNN, YOLO, and SSD employ CNN backbones to enable real-time detection with high precision. They convincingly surpass traditional sliding-window pipelines on HOG and SVM by achieving notable speed and accuracy improvements. For example, YOLOv4 achieves 43.5 mAP [53] on the COCO [54] benchmark while traditional methods fall behind. Figure 6 shows a fuzzy logic–based image enhancement method [55] that enhances a concatenated CNN model for pneumonia detection. The study demonstrates that classification performance is significantly improved using fuzzy entropy–based enhancement, achieving 98.9% accuracy, 99.3% precision, 99.8% F1-score, and 99.6% recall, thereby outperforming traditional image enhancement methods in medical image analysis.

Besides detection, deep segmentation models such as U-Net and DeepLab utilize context-aware learning to capture high-level semantic features. As opposed to the common techniques such as GMMs, the models learn complex spatial dependencies through complex object structures. For instance, DeepLabv3+ achieves 89% mean Intersection over Union on PASCAL VOC 2012 [56], as compared to 74.7% achieved by Conditional Random Field (CRF)-based algorithms [57]. These capabilities are further generalized in models such as GANs and VAEs, enabling data augmentation and realistic image generation. These models address the issues of data scarcity in training datasets by diversifying and scaling them. For example, GAN-based data augmentation has been shown to improve CNN accuracy on medical imaging-related tasks by 5–10% [58].

Meanwhile, GMMs remain widely used for background subtraction and density estimation due to their computational efficiency. However, they assume linear separability in the feature space, which limits their ability to model complex textures and object boundaries. Classical GMM–based background subtraction methods achieve moderate F-measure values (often in the 0.70–0.80 range depending on scenario) on standard benchmarks such as CDnet2014 [59]; by contrast, modern deep learning approaches such as the FgSegNet family reach F-measure values of ≈0.98 on the same dataset [60]. ViTs replace convolutional layers with multi-head self-attention across patches of the image, which natively allows them to model global context. When scaled up and pretrained on extremely large datasets (e.g., JFT-300M), ViTs have been demonstrated to surpass equivalent CNNs on ImageNet (e.g., large ViT sizes achieve ≈88.5% top-1 after enormous pretraining), though small ViT sizes trained on only ImageNet-1k tend to perform worse than powerful ResNet baselines (ResNet-152 top-1 ≈78%), so the benefit relies heavily on model scale and pretraining dataset [61]. Figure 7 [62] proposes TransResNet, a parallel Transformer-CNN architecture with a Cross Grafting Module that fuses multi-resolution features for improved medical image segmentation, demonstrating state-of-the-art or competitive performance across 10 datasets, including skin lesion, retinal vessel, and polyp segmentation tasks.

Taken together, these observations reveal a clear divide:

a. Traditional methods (HOG, GMMs) are interpretable but fail under occlusion, noise, and large-scale variability.

b. Deep learning methods (CNNs, ViTs) achieve superior performance (+10–20% in accuracy) but demand high computational power.

c. GANs and VAEs uniquely address data scarcity, a critical bottleneck in earlier pipelines.

d. ViTs offer a promising direction for modeling long-range dependencies, though efficiency and data requirements remain unsolved challenges.

From this review, several key findings emerge. First, deep learning approaches universally outperform traditional methods in terms of accuracy, scalability, and robustness. For example, CNN-based edge detectors achieve ~15–20% higher F-scores than Canny. Second, fusion of features, whether handcrafted (e.g., HOG + color) or deep (multi-scale CNN features), consistently improves performance, suggesting that leveraging complementary cues yields statistically significant benefits. Third, robustness to occlusion and corruption remains a challenge. While CNNs and ViTs outperform classical methods, performance still drops significantly under adverse conditions. Finally, ViTs show particular promise for long-range dependency modeling and occlusion

Figure 7
Example images with ground truth and predicted masks for skin lesion (row 1), polyp (row 2), and retinal vessel (row 3) segmentation tasks

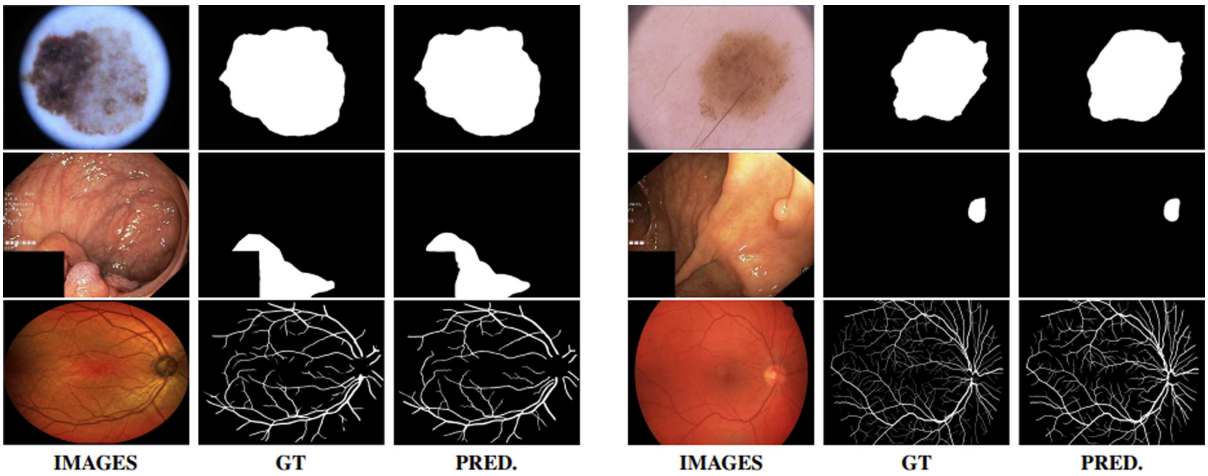


Table 3
Summary of the comparison of traditional versus deep learning approaches in computer vision

Method	Key statistics	Strengths	Weaknesses/findings
HOG + SVM	Miss rate ~10% at 10^{-4} FPR (pedestrian detection)	Efficient, interpretable	Poor under occlusion/clutter
Canny	ODS F-score ~0.811 on BSDS500	Good localization	Sensitive to noise, tuning required
CNN (e.g., ResNet-50)	Top-1 ~80.67% on ImageNet	Strong local feature extraction, efficient for small data	Limited global context
ViT-B/16	Top-1 ~79.9% on ImageNet	Global context, robust to occlusion	Data-hungry, compute-intensive
ViT in medical imaging	95.2% accuracy (kidney stone task)	Superior in complex domains	Not ideal for low-data tasks
GAN (CNN discriminator)	Boost CNN accuracy on medical imaging tasks by 5–10%	Stable generative performance	Transformer-based discriminators less reliable
GMM	Above 80% F1-score on the PETS2001 dataset	Simple, unsupervised segmentation	Too simplistic for rich image content

resilience, but they remain computationally expensive and less interpretable. Table 3 summarizes these key findings.

In diverse domains, computer vision continues to deliver measurable real-world performance improvements. In medical imaging, CNN-based models tripled in accuracy for pneumonia diagnosis (up to ~95%), achieved ~98% accuracy and precision in tuberculosis detection, and even exceeded 99% accuracy in Chronic Obstructive Pulmonary Disease (COPD) exacerbation prediction on chest X-rays and CT scans [63]. Meanwhile, ViTs applied to medical imaging achieved an F1 score of 0.9532 and a recall of 0.9533 on chest X-rays, signaling competitive performance with CNNs [64]. In autonomous driving, certain LiDAR-based models on the KITTI benchmark now reach car detection Average Precisions (APs) in the high 80s (e.g., ~90% on the "moderate" difficulty), often while running at real-time frame rates [65]. Sports applications likewise reflect impressive accuracy—computer vision systems such as Hawk-Eye help increase key decision accuracy in soccer from 82% to 96%, while deep learning models using convolutional Long Short-Term Memory networks (LSTMs) achieve 97% classification accuracy in beach-volleyball activity recognition [66, 67]. Lastly, in military-esque or high-speed tracking scenarios, specialized CNN processors supporting ultra-high frame rates (e.g., tens of thousands of frames per second) are used for tasks such as missile tracking and spark-plug detection, providing precise, real-time analysis that far outpaces traditional compute methods [68, 69].

Thus, in summary, it has been made clear through evidence that, although deep learning may take the top spot in accuracy and robustness, classical approaches, as well as learning-based approaches, have their own advantages in efficiency, interpretability, and applicability in resource-constrained environments. The future of research in this field, therefore, does not lie in discarding traditional approaches, but rather in using a mix of traditional approaches, learning-based approaches, and deep learning approaches, thereby achieving a balance between efficiency, interpretability, and accuracy. Traditional approaches, such as edges and HOG, help in achieving efficiency through lightweight processing, thereby reducing complexity. Moreover,

it also improves interpretability through traditional visual cues, while deep learning improves accuracy through semantics.

The comparison between different domains also indicates the areas of strength, weakness, and research direction, thereby helping us understand what areas need more research, thereby indicating future trends.

8. Emerging Trends and Future Prospects

Computer vision trends and opportunities are marked by innovation in AI techniques, real-time processing, improved spatial interpretation, and greater focus on explainability and fairness. More potential lies ahead with advancements in multimodal learning, synthetic data, healthcare, and privacy-preserving techniques, transforming industries and enabling human–computer interaction.

8.1. Trends

AI-powered and Deep Learning Advances: Deep learning, particularly with CNNs, remains a top approach in computer vision, yet emerging advances such as ViTs are gaining more traction. These models, which are founded on the self-attention mechanism, illustrate state-of-the-art performance on image classification and object detection tasks, rivaling the traditional CNN-based models. As AI models become more powerful and capable of handling massive datasets [70], the accuracy and robustness of computer vision systems will continue to improve.

Edge Computing and Real-Time Processing: However, with the emergence of edge computing [71], computer vision is also used for real-time applications where the processing is done on the device itself and not on the cloud. This is another area that is important to the development of autonomous cars, robots, and augmented reality (AR). Other devices, such as smartphones, drones, and wearable devices, are employing computer vision for edge computing.

Explainability and Fairness in AI: As AI models, especially computer vision, are being used for high-stakes applications such as healthcare, security, and law enforcement, the need for explainable AI [72] has been rising. The aim of the researchers is to make the decision-making process of computer vision more transparent so that humans can better understand the process. Another important aspect of computer vision is fairness and bias, and the need to reduce discrimination using diverse techniques and datasets.

3D Vision and Spatial Awareness: The future of computer vision is heading toward better 3D vision [73], where objects and their relations are understood in 3D space. This includes improvements in depth sensors, stereo vision, and LiDAR, which can be applied to autonomous vehicles, AR, and robotics. This is going to allow a more holistic understanding of the environment, which can be used for complex interactions and predictions.

8.2. Future prospects

Multimodal learning and cross-modal systems are an emerging trend, combining visual, auditory, and textual data to support computer vision models [74]. Integrating vision with NLP [75] enables tasks such as image captioning and visual question answering. This movement leads to more holistic AI systems that understand and respond to the world more like humans.

Synthetic Data and Augmentation: As obtaining large labeled datasets is costly and tedious, synthetic data generation [76] is becoming popular. Synthetic data is created via simulation, rendering, or generative models such as GANs. These methods produce diverse labeled data for object detection and scene understanding, enhancing training without real-world data, especially in applications such as autonomous driving, where data collection is expensive.

AI in Healthcare and Diagnostics: The use of computer vision in healthcare is increasing with advances in medical image analysis. AI systems can identify early disease signals [77], from cancer detection in radiology to retinal disease detection. Improved AI diagnostics will be increasingly used in clinics to assist physicians, reduce errors, and provide faster diagnoses, improving patient outcomes.

Augmented Reality (AR) and Virtual Reality (VR): Computer vision is central to the development of AR and VR technologies [78]. As AR and VR devices are increasingly used in industries such as entertainment, education, and retail, computer vision will make the experience more interactive and immersive. For example, real-time object detection and tracking in AR will enable users to interact with the virtual world placed over their real surroundings. While that happens, VR platforms will utilize computer vision to render simulations more realistic and improve users' experiences.

Privacy-Preserving Computer Vision: With increasing privacy concerns, privacy-preserving [79] computer vision techniques are becoming increasingly popular. For instance, federated learning offers a framework to train models on decentralized devices without raw data ever leaving the device, ensuring privacy. It will be crucial in applications such as surveillance, healthcare, and personal assistants that involve sensitive data.

9. Discussions

The above article provided an overview of the history and current status of computer vision, focusing on AI and how it aids in understanding images. Computer vision is a wide and complex field, and the above article has discussed the basics of computer vision to ensure that the reader understands computer vision in a better manner. Computer vision is a rapidly evolving field, and the latest developments in computer vision are a clear indication of its importance in AI. Images have been used within this essay to make the topic interesting and easy to understand. Computer vision is a technology used within AI to interpret images.

10. Conclusion

In conclusion, computer vision is the foremost technology of AI wherein machines can decode and understand visual data. Despite vast strides in developing computer vision algorithms and techniques, there are limitations and limitations. Despite the advancements, technical problems related to the proper interpretation of complex settings and machines being capable of truly "understanding" real-world scenarios are still extensive. The future of computer vision is highly promising, with ongoing research continuously pushing the frontiers of innovation. With additional advancement of such technologies, they will have a greater effect on other sectors, offering groundbreaking solutions to medicine and autonomous systems. The continued evolution of computer vision will undoubtedly play a vital role in shaping the destiny of AI.

Ethical Statement

The authors declare that this study did not require formal ethical approval, as it is based on publicly available datasets and

constitutes a review article. No human participants or personal data were directly involved. Furthermore, any images reproduced from previously published work are properly cited in accordance with academic standards. According to the research guidelines of The Kyoto College of Graduate Studies for Informatics, Institutional Review Board (IRB) or ethics committee approval is not required for such studies.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

Author Contribution Statement

Mohammad Mehedi Hassan: Conceptualization, Methodology, Software, Formal analysis, Resources, Data curation, Writing – original draft, Visualization, Project administration. **Stephen Karungaru:** Validation, Investigation, Resources, Writing – review & editing, Supervision, Project administration. **Rezaul Bashar:** Validation, Writing – review & editing.

References

[1] Sheikh, H., Prins, C., & Schrijvers, E. (2023). Artificial intelligence: Definition and background. In H. Sheikh, C. Prins, & E. Schrijvers (Eds.), *Mission AI: The new system technology* (pp. 15–41). Springer Cham. https://doi.org/10.1007/978-3-031-21448-6_2

[2] Goel, A., Goel, A. K., & Kumar, A. (2023). The role of artificial neural network and machine learning in utilizing spatial information. *Spatial Information Research*, *31*(3), 275–285. https://doi.org/10.1007/s41324-022-00494-x

[3] Turing, A. M. (1950). Computing machinery and intelligence. *Mind*, *59*(236), 433–460. https://doi.org/10.1093/mind/LIX.236.433

[4] Atchison, D. (2023). *Optics of the human eye*. USA: CRC Press.

[5] Szeliski, R. (2022). *Computer vision: Algorithms and applications*. Germany: Springer Nature.

[6] Salmon, J. F. (2024). *Kanski's clinical ophthalmology e-book: A systematic approach*. Netherlands: Elsevier Health Sciences.

[7] Perera Molligoda Arachchige, A. S., & Svet, A. (2021). Integrating artificial intelligence into radiology practice: Undergraduate students' perspective. *European Journal of Nuclear Medicine and Molecular Imaging*, *48*(13), 4133–4135. https://doi.org/10.1007/s00259-021-05558-y

[8] Chen, R., Karungaru, S., Terada, K., & Xu, C. (2025). 3D visualization system of breast magnetic resonance images based on deep learning and volume rendering. *IEEE Access*, *13*, 1–20. https://doi.org/10.1109/ACCESS.2025.3584874

[9] Nsinga, R., Karungaru, S., & Terada, K. (2022). Auto-differentiated fixed point notation on low-powered hardware acceleration. *Journal of Signal Processing*, *26*(5), 131–140. https://doi.org/10.2299/jsp.26.131

[10] Ahmad, H., Farhan, M., & Farooq, U. (2023). Computer vision techniques for military surveillance drones. *Wasit Journal of Computer and Mathematics Science*, *2*(2), 53–59. https://doi.org/10.31185/wjcms.148

[11] Karungaru, S., Tsuji, R., & Terada, K. (2022). Driving assistance: Pedestrians and bicycles accident risk estimation using onboard front camera. *International Journal of Intelligent Transportation Systems Research*, *20*(3), 768–777. https://doi.org/10.1007/s13177-022-00324-2

[12] Wiseman, Y. (2021). Autonomous vehicles. In M. Khosrow-Pour. D.B.A (Ed.), *Encyclopedia of Information Science and Technology, Fifth Edition* (pp. 878–889). Igi Global. https://doi.org/10.4018/978-1-7998-3479-3.ch001

[13] Umbaugh, S. E. (2023). *Digital image processing and analysis: Computer vision and image analysis*. USA: CRC Press.

[14] SMenconero, S. (2021). Image processing for knowledge and comparison of Piranesi's Carceri editions. In D. Villa & F. Zuccoli (Eds.), *International and Interdisciplinary Conference on Image and Imagination*, 1–10. Springer. https://doi.org/10.1007/978-3-031-25906-7_1

[15] Pu, M., Huang, Y., Liu, Y., Guan, Q., & Ling, H. (2022). Edter: Edge detection with transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1402–1412. https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.00146

[16] Pradeep, N. R., & Ravi, J. (2021). An accurate fingerprint recognition algorithm based on histogram oriented gradient (HOG) feature extractor. *International Journal of Electrical Engineering & Technology*, *12*(2), 19–32. https://doi.org/10.34218/IJEET.12.2.2021.003

[17] Shi, L., & Zhao, Y. (2023). Edge detection of high-resolution remote sensing image based on multi-directional improved Sobel operator. *IEEE Access*, *11*, 135979–135993. https://doi.org/10.1109/ACCESS.2023.3338355

[18] Chaabane, S. B., Hijji, M., Harrabi, R., & Seddik, H. (2022). Face recognition based on statistical features and SVM classifier. *Multimedia Tools and Applications*, *81*(6), 8767–8784. https://doi.org/10.1007/s11042-021-11816-w

[19] Lu, S., Ding, Y., Liu, M., Yin, Z., Yin, L., & Zheng, W. (2023). Multiscale feature extraction and fusion of image and text in VQA. *International Journal of Computational Intelligence Systems*, *16*(1), 54. https://doi.org/10.1007/s44196-023-00233-6

[20] Luo, C., Sun, X., Sun, X., & Song, J. (2021). Improved Harris corner detection algorithm based on Canny edge detection and Gray difference preprocessing. *Journal of Physics: Conference Series*, *1971*(1), 012088. https://doi.org/10.1088/1742-6596/1971/1/012088

[21] Cuhadar, C., & Tsao, H. N. (2022). A computer vision sensor for AI—Accelerated detection and tracking of occluded objects. *Advanced Intelligent Systems*, *4*(11), 2100285. https://doi.org/10.1002/aisy.202100285

[22] Greff, K., Belletti, F., Beyer, L., Doersch, C., Du, Y., Duckworth, D., . . . , & Tagliasacchi, A. (2022). Kubric: A scalable dataset generator. In *IEEE/CVF conference on computer vision and pattern recognition*, 3749–3761.

[23] Dhillon, D., & Chouhan, R. (2022). Enhanced edge detection using SR-guided threshold maneuvering and window mapping: Handling broken edges and noisy structures in Canny edges. *IEEE Access*, *10*, 11191–11205. https://doi.org/10.1109/ACCESS.2022.3145428

[24] Wu, Y. C., & Feng, J. W. (2018). Development and application of artificial neural network. *Wireless Personal*

Communications, *102*(2), 1645–1656. https://doi.org/10.1007/s11277-017-5224-x

[25] Chong, L. (2024). Decoding the alignment problem: Revisiting the 1958 NYT report on Rosenblatt's perceptron through the lens of information theory. In C. Stephanidis, M. Antona, S. Ntoa, & G. Salvendy (Eds.), *International Conference on Human-Computer Interaction*, 23–35. Springer. https://doi.org/10.1007/978-3-031-62110-9_3

[26] Almansi, K. Y., Shariff, A. R. M., Kalantar, B., Abdullah, A. F., Ismail, S. N. S., & Ueda, N. (2022). Performance evaluation of hospital site suitability using multilayer perceptron (MLP) and analytical hierarchy process (AHP) models in Malacca, Malaysia. *Sustainability*, *14*(7), 3731. https://doi.org/10.3390/su14073731

[27] Zhang, X., Zhang, X., & Wang, W. (2023). Convolutional neural network. In X. Zhang, X. Zhang, & W. Wang (Eds.), *Intelligent Information Processing with Matlab* (pp. 39–71). Springer. https://doi.org/10.1007/978-981-99-6449-9_2

[28] Ullah, A., Elahi, H., Sun, Z., Khatoon, A., & Ahmad, I. (2022). Comparative analysis of AlexNet, ResNet18 and SqueezeNet with diverse modification and arduous implementation. *Arabian Journal for Science and Engineering*, *47*(2), 2397–2417. https://doi.org/10.1007/s13369-021-06182-6

[29] Li, D., Ling, H., Kim, S. W., Kreis, K., Fidler, S., & Torralba, A. (2022). Bigdatasetgan: Synthesizing imagenet with pixel-wise annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21330–21340.

[30] Zhang, X. (2021). The AlexNet, LeNet-5 and VGG NET applied to CIFAR-10. In *International Conference on Big Data & Artificial Intelligence & Software Engineering*, 414–419. https://doi.org/10.1109/ICBASE53849.2021.00083

[31] Xu, W., Fu, Y. L., & Zhu, D. (2023). ResNet and its application to medical image processing: Research progress and challenges. *Computer Methods and Programs in Biomedicine*, *240*, 107660. https://doi.org/10.1016/j.cmpb.2023.107660

[32] Neshat, M., Ahmed, M., Askari, H., Thilakaratne, M., & Mirjalili, S. (2024). Hybrid inception architecture with residual connection: Fine-tuned inception-ResNet deep learning model for lung inflammation diagnosis from chest radiographs. *Procedia Computer Science*, *235*, 1841–1850. https://doi.org/10.1016/j.procs.2024.04.175

[33] Koonce, B. (2021). EfficientNet. In B. Koonce (Ed.), *Convolutional neural networks with swift for tensorflow* (pp. 109–123). Apress. https://doi.org/10.1007/978-1-4842-6168-2_10

[34] Takahashi, S., Sakaguchi, Y., Kouno, N., Takasawa, K., Ishizu, K., Akagi, Y., . . . , & Hamamoto, R. (2024). Comparison of vision transformers and convolutional neural networks in medical image analysis: A systematic review. *Journal of Medical Systems*, *48*(1), 84. https://doi.org/10.1007/s10916-024-02105-8

[35] Brynjolfsson, E., Li, D., & Raymond, L. (2025). Generative AI at work. *The Quarterly Journal of Economics*, *140*(2), 889–942. https://doi.org/10.1093/qje/qjae044

[36] Zhang, X., Jiang, W., Shen, C., Li, Q., Wang, Q., Lin, C., & Guan, X. (2025). Deep learning library testing: Definition, methods and challenges. *ACM Computing Surveys*, *57*(7), 1–37. https://doi.org/10.1145/3716497

[37] Hmidani, O., & Alaoui, E. I. (2022). A comprehensive survey of the R-CNN family for object detection. In *International Conference on Advanced Communication Technologies and Networking*, *1–6*. https://doi.org/10.1109/CommNet56067.2022.9993862

[38] Nazir, A., & Wani, M. A. (2023). You only look once-object detection models: A review. In *International Conference on Computing for Sustainable Global Development*, 1088–1095.

[39] Mo, Y., Wu, Y., Yang, X., Liu, F., & Liao, Y. (2022). Review the state-of-the-art technologies of semantic segmentation based on deep learning. *Neurocomputing*, *493*, 626–646. https://doi.org/10.1016/j.neucom.2022.01.005

[40] Zhang, S., & Zhang, C. (2023). Modified U-Net for plant diseased leaf image segmentation. *Computers and Electronics in Agriculture*, *204*, 107511. https://doi.org/10.1016/j.compag.2022.107511

[41] Targa, L., Cano, C., Solbes-García, Á., Casas, S., Alba, E., & Portalés, C. (2025). Automated edge detection for cultural heritage conservation: Comparative evaluation of classical and deep learning methods on artworks affected by natural disaster damage. *Applied Sciences*, *15*(15), 8260. https://doi.org/10.3390/app15158260

[42] Wang, X., Wang, X., Li, J., Liang, W., & Bi, C. (2024). Research on pavement crack detection based on random structure forest and density clustering. *Automation*, *5*(4), 467–483. https://doi.org/10.3390/automation5040027

[43] Xie, S., & Tu, Z. (2015). Holistically-nested edge detection. In *IEEE international conference on computer vision*, 1395–1403. https://doi.org/10.1109/ICCV.2015.164

[44] Hu, G., & Saeli, C. (2024). Enhancing deep edge detection through normalized hadamard-product fusion. *Journal of Imaging*, *10*(3), 62. https://doi.org/10.3390/jimaging10030062

[45] Li, Y., Poma, X. S., Xi, Y., Li, G., Yang, C., Xiao, Q., . . . , & Li, Z. (2026). A new baseline for edge detection: Make encoder—decoder great again. *Signal Processing: Image Communication*, *142*, 117485. https://doi.org/10.1016/j.image.2026.117485

[46] Ayalew, A. M., Salau, A. O., Abeje, B. T., & Enyew, B. (2022). Detection and classification of COVID-19 disease from X-ray images using convolutional neural networks and histogram of oriented gradients. *Biomedical Signal Processing and Control*, *74*, 103530. https://doi.org/10.1016/j.bspc.2022.103530

[47] Yongjun, Z., Yong, Z., Guoliang, L., Daimeng, W., & Ruzhong, C. (2015). Efficient and real-time pedestrian detection at night-time environments. *International Journal of Innovative Computing, Information and Control*, *11*(02), 599.

[48] Zhao, Y., Zhang, Y., Cheng, R., Wei, D., & Li, G. (2015). An enhanced histogram of oriented gradients for pedestrian detection. *IEEE Intelligent Transportation Systems Magazine*, *7*(3), 29–38. https://doi.org/10.1109/MITS.2015.2427366

[49] Dollár, P., Tu, Z., Perona, P., & Belongie, S. J. (2009). Integral channel features. In *British Machine Vision Conference*, *2*(3), 1–11. https://doi.org/10.5244/C.23.91

[50] Huang, C., & Huang, J. (2017). A fast HOG descriptor using lookup table and integral image. *arXiv Preprint:1703.06256*.

[51] Yang, W. J., Lo, C. Y., & Chung, P. C. (2022). Weighted module linear regression classifications for partially-occluded face. In P. E. Ambrósio (Ed.), *Digital Image Processing Applications* (pp. 1–21). IntechOpen.

[52] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

[53] Terven, J., Córdova-Esparza, D. M., & Romero-González, J. A. (2023). A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine Learning and Knowledge Extraction*, *5*(4), 1680–1716. https://doi.org/10.3390/make5040083

[54] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., . . . , & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *European conference on computer vision*, 740–755. Springer. https://doi.org/10.1007/978-3-319-10602-1_48

[55] Buriboev, A. S., Muhamediyeva, D., Primova, H., Sultanov, D., Tashev, K., & Jeon, H. S. (2024). Concatenated CNN-based pneumonia detection using a fuzzy-enhanced dataset. *Sensors*, *24*(20), 6750. https://doi.org/10.3390/s24206750

[56] Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018). Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision*, 801–818.

[57] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., . . . , & Torr, P. H. (2015). Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision*, 1529–1537.

[58] Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., & Greenspan, H. (2018). GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification. *Neurocomputing*, *321*, 321–331. https://doi.org/10.1016/j.neucom.2018.09.013

[59] Wang, Y., Jodoin, P. M., Porikli, F., Konrad, J., Benezeth, Y., & Ishwar, P. (2014). CDnet 2014: An expanded change detection benchmark dataset. In *Conference on Computer Vision and Pattern Recognition Workshops*, 387–394.

[60] Lim, L. A., & Keles, H. Y. (2018). Foreground segmentation using convolutional neural networks for multiscale feature encoding. *Pattern Recognition Letters*, *112*, 256–262. https://doi.org/10.1016/j.patrec.2018.08.002

[61] Dosovitskiy, A. (2020). An image is worth 16 × 16 words: Transformers for image recognition at scale. *arXiv Preprint:2010.11929*.

[62] Sharif, M. H., Demidov, D., Hanif, A., Yaqub, M., & Xu, M. (2022). TransResNet: Integrating the strengths of ViTs and CNNs for high resolution medical image segmentation via feature grafting. In *British Machine Vision Conference Proceedings*.

[63] Ahmed, M. S., Rahman, A., AlGhamdi, F., AlDakheel, S., Hakami, H., AlJumah, A., & Basheer Ahmed, M. I. (2023). Joint diagnosis of pneumonia, COVID-19, and tuberculosis from chest X-ray images: A deep learning approach. *Diagnostics*, *13*(15), 2562. https://doi.org/10.3390/diagnostics13152562

[64] Astley, S. M., & Wismüller, A. (2025). Medical imaging 2025: Computer-Aided diagnosis. In *Society of Photo-Optical Instrumentation Engineers Conference Series*, 13407.

[65] Liang, L., Ma, H., Zhao, L., Xie, X., Hua, C., Zhang, M., & Zhang, Y. (2024). Vehicle detection algorithms for autonomous driving: A review. *Sensors*, *24*(10), 3088. https://doi.org/10.3390/s24103088

[66] Chen, C., Xue, J., Gou, W., Xie, M., & Yao, X. (2025). Quantitative analysis and evaluation of research on the application of computer vision in sports since the 21st century. *Frontiers in Sports and Active Living*, *7*, 1604232. https://doi.org/10.3389/fspor.2025.1604232

[67] Mathew, A., & Akanksha. (2024). Volleyball action recognition based on skeleton data using LSTM. In *International Conference on Computer Vision and Robotics*, 397–407. https://doi.org/10.1007/978-981-97-8868-2_31

[68] Singh, U. K., Padmanabhan, V., & Agarwal, A. (2014). Dynamic classification of ballistic missiles using neural networks and hidden Markov models. *Applied Soft Computing*, *19*, 280–289. https://doi.org/10.1016/j.asoc.2014.02.015

[69] Farooq, J., & Bazaz, M. A. (2023). Hybrid deep neural network for data driven missile guidance with maneuvering target. *Defence Science Journal*, *73*(5), 602–611. https://doi.org/10.14429/dsj.73.18481

[70] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., . . . , & Girshick, R. (2023). Segment anything. In *IEEE/CVF international conference on computer vision*, 4015–4026. https://doi.org/10.1109/ICCV51070.2023.00371

[71] Hua, H., Li, Y., Wang, T., Dong, N., Li, W., & Cao, J. (2023). Edge computing with artificial intelligence: A machine learning perspective. *ACM Computing Surveys*, *55*(9), 1–35. https://doi.org/10.1145/3555802

[72] Dwivedi, R., Dave, D., Naik, H., Singhal, S., Omer, R., Patel, P., . . . , & Ranjan, R. (2023). Explainable AI (XAI): Core ideas, techniques, and solutions. *ACM Computing Surveys*, *55*(9), 1–33. https://doi.org/10.1145/3561048

[73] Poggi, M., & Moeslund, T. B. (2021). Computer vision for 3D perception and applications. *Sensors, 21*(12), 3944. https://doi.org/10.3390/s21123944

[74] Bouchey, B., Castek, J., & Thygeson, J. (2021). Multimodal learning. In J. Ryoo & K. Winkelmann (Eds.), *Innovative Learning Environments in STEM Higher Education: Opportunities, Challenges, and Looking Forward* (pp. 35–54). Springer International Publishing. https://doi.org/10.1007/978-3-030-58948-6_3

[75] Lane, H., & Dyshel, M. (2025). *Natural language processing in action*. USA: Simon and Schuster.

[76] Figueira, A., & Vaz, B. (2022). Survey on synthetic data generation, evaluation methods and GANs. *Mathematics*, *10*(15), 2733. https://doi.org/10.3390/math10152733

[77] Mehmood, S., Ghazal, T. M., Khan, M. A., Zubair, M., Naseem, M. T., Faiz, T., & Ahmad, M. (2022). Malignancy detection in lung and colon histopathology images using transfer learning with class selective image processing. *IEEE Access*, *10*, 25657–25668. https://doi.org/10.1109/ACCESS.2022.3150924

[78] Tan, Y., Xu, W., Li, S., & Chen, K. (2022). Augmented and virtual reality (AR/VR) for education and training in the AEC industry: A systematic review of research and applications. *Buildings*, *12*(10), 1529. https://doi.org/10.3390/buildings12101529

[79] Hinojosa, C., Niebles, J. C., & Arguello, H. (2021). Learning privacy-preserving optics for human pose estimation. In *IEEE/CVF International Conference on Computer Vision*, 2573–2582. https://doi.org/10.1109/ICCV48922.2021.00257

How to Cite: Hassan, M. M., Karungaru, S., & Bashar, R. (2026). A Short Review on Computer Vision: Visualizing the World Through Machine. *Artificial Intelligence and Applications*. https://doi.org/10.47852/bonviewAIA62027265