linkcolorblue
citecolorblue
urlcolorblue

Faculty of Mathematics
and Computer Science

Machine learning course (ML)

A Comparative Survey of Recent Deep Learning Architectures for Object Recognition

Gherasim Delia-Catalina

Department of Computer Science, Babeş-Bolyai University
1, M. Kogălniceanu Street, 400084, Cluj-Napoca, Romania
E-mail: delia.gherasim@stud.ubbcluj.ro

Abstract

Object recognition is a fundamental problem in computer vision, enabling systems to automatically identify and classify visual content across diverse real-world scenarios. With the rapid evolution of deep learning, numerous architectures have been developed to improve accuracy, computational efficiency, and generalization. This paper presents a comparative survey of four deep learning architectures used for object recognition: ResNet-50, EfficientNet-B0, MobileNetV3-Large, and Vision Transformer (ViT-Base), tracing the progression from traditional Convolutional Neural Networks (CNNs) to modern transformer-based models. The survey focuses on their architectural innovations, including the use of residual connections, compound scaling, depthwise separable convolutions, and global self-attention mechanisms. The paper reviews the design principles and analyzes the performance of these models based on benchmark metrics reported on large-scale datasets, primarily ImageNet. Key metrics compared include Top-1 accuracy, total learnable parameters, and Floating-Point Operations (FLOPs), which together assess how well a model balances accuracy and computational efficiency under different hardware limitations, from high-performance GPUs to mobile devices. The findings underscore the relevance of efficiency oriented CNNs, such as MobileNetV3 and EfficientNet-B0, for resource constrained edge computing, contrasted with the rising dominance of transformer-based models, such as ViT, in achieving accuracy. This comparative analysis aims to provide an overview for selecting suitable deep learning architectures for object recognition tasks across varying computational and application domains, including autonomous driving and medical imaging.

Keywords:
Deep Learning; Object Recognition; CNN; Vision Transformer; EfficientNet

© 2025 .

1. Introduction

Computer vision originated in the 1960s at MIT, when Larry Roberts proposed using geometric principles and the properties of polyhedra to reconstruct the three-dimensional shape of an object from its two-dimensional image [11]. Since then, the field has progressed rapidly, making it possible to develop systems for tasks like facial recognition, medical image interpretation, and autonomous vehicle navigation.

Object recognition involves detecting and classifying the objects present in an image. This process is inherently challenging, as it requires integrating multiple sources of information such as geometric constraints, the semantic context of the scene, and the visual features available within the image [19].

The goal of Object Recognition (OR) models is to identify what objects appear in an image and where they are located. OR combines the principles of image classification and object detection. While all OR systems aim to determine the present objects, their implementation varies widely. Traditional approaches relied on features such as edges, corners, and textures, failing to generalize across varying lighting conditions, viewing angles, and object occlusions. As a result, their performance was often limited by the designer's ability to manually engineer suitable feature representations.

The rise of deep learning has enabled models to extract layered feature representations directly from the data without manual design. Convolutional Neural Networks (CNNs) extract spatial features at multiple levels of abstraction, leading to significant improvements in accuracy and robustness. This shift from manually designed features to data-driven learning has changed how object recognition systems are developed and deployed.

Despite these advances, selecting an appropriate deep learning architecture remains a challenge. Numerous models have been proposed—each offering trade-offs between accuracy, computational efficiency, and suitability for deployment on various platforms, from high-performance servers to edge devices. Understanding these trade-offs is essential for designing systems that balance recognition performance with real-time constraints and resource availability.

The objective of this work is to present a comparative survey of recent deep learning architectures used for object recognition. The study aims to analyze their design principles, performance characteristics, and application contexts, highlighting the strengths and limitations.

The paper will first offer a theoretical background in the first chapter, explaining the concept of CNNs and their advantages. In the third chapter we will discuss Recent Deep Learning Architectures for Object Recognition: ResNet-50, EfficientNet-B0, MobileNetV3 and the Vision Transformer. The forth chapter presents a discussion comparing accuracy and performance, while the fifth presents the conclusions.

2. Theoretical Overview

2.1. Convolutional Neural Networks

A digital image is a tensor with height, width, and three color channels (R, G, B). In deep networks, early layers detect simple patterns such as edges, while deeper layers identify more complex shapes. Fully connected networks are not ideal for images because they require many parameters and ignore local spatial structure. Convolution addresses this by applying small filters to local regions and reusing the same weights across the image. A convolutional layer performs this operation by sliding kernels across the input to produce feature maps [16].

CNNs are therefore appropriate for image tasks. Like other neural networks, they consist of neurons that compute outputs and update weights, but CNNs rely on convolutional and pooling operations to learn spatial patterns efficiently [15]. A typical CNN includes an input layer and several hidden layers. As depth increases, features become more abstract and the receptive field grows, allowing deeper layers to capture larger and more complex structures [16].

The convolutional layer applies feature detectors to the input in order to generate activation maps. These maps highlight learned patterns such as edges [2]. A small kernel moves across the image, computing element-wise products and sums. Padding is used to handle image borders, and the stride controls how far the kernel moves each step. After convolution, a non-linear activation function such as ReLU is usually applied [4].

Goodfellow et al. define convolution as an operation between: the input x and the kernel w, producing an output s(t)

$$s(t) = (x * w)(t)$$

In machine learning, the input is typically multidimensional, and the kernel is a set of learnable parameters. In image processing, convolution is applied to multiple dimensions at once. For a 2d image and a kernel, the convolution can be written as:

$$S(i, j) = (I * K)(i, j) = \sum_{m} \sum_{n} I(m, n)K(i - m, j - m, j - n)$$

[7]

The result, $s(i, j)$, is a feature map representing the response of the kernel applied to the input image. This map is a condensed version of the original image that emphasizes the learned patterns. Following convolution, pooling layers are often used to further reduce the spatial size of the feature map while preserving its important information

2.1.1. Pooling Layers

A pooling layer downsamples feature maps to reduce computation and improve robustness to small shifts. The two common types are max pooling, which selects the largest value in each region, and average pooling, which computes the mean. Pooling produces a smaller representation while preserving key information.

2.1.2. Fully Connected Layers

After features are extracted, the network typically uses fully connected layers for classification or prediction. The feature maps are flattened and passed through one or more FC layers. Activation functions like softmax convert the final outputs into probabilities suitable for tasks such as image classification.

2.1.3. Advantages of CNNs

CNNs have several advantages over traditional neural networks when working with images. They reduce the amount of computation, improve performance, and use weight sharing, where the same filter is applied across different parts of an image. This greatly reduces the number of parameters. CNNs also rely on local receptive fields, meaning they first detect small local patterns and later combine them to recognize more complex shapes. [6]

Unlike fully connected networks, CNNs exploit the two-dimensional nature of images through local connectivity and weight sharing. This design requires far fewer parameters, making training faster and simpler. This idea is inspired by how cells in the visual cortex process visual information. Another major advantage of CNNs is that they do not require manual feature engineering; the network learns useful features automatically.

2.1.4. Disadvantages of CNNS

The most commonly encountered problem in CNN image classification and object recognition is overfitting. The amount of data required for training is very large, but the complexity of analyzing images makes it hard to abstract the shapes if the camera or lighting change.

2.2. Vision Transformers

Originating in Natural Language Processing, the Transformer treats data as a sequence of tokens. For vision tasks, an image is split into patches, which are flattened, transformed into a vector through a linear projection, and then fed into the Self-Attention mechanism. Unlike convolution, which is local, self-attention is global. It computes the relationship between every patch and every other patch in the image simultaneously. The core operation, Scaled Dot-Product Attention, uses three matrices derived from the input: Queries (Q), Keys (K), and Values (V).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Here, the dot product QK^T measures the similarity between patches. The softmax function converts these scores into normalized attention weights, which are then applied to combine the Values (V). [3]

This lacks the inductive biases of CNNs. ViTs do not inherently know that adjacent pixels are related or that objects are translation-invariant. Consequently, they must learn these spatial rules from data, requiring significantly larger datasets to converge to the same performance levels as CNNs on smaller tasks. [22]

2.3. *Neural Architecture Search*

Models like MobileNetV3 and EfficientNet are based on automated search algorithms. NAS typically involves a search space (possible layer types, kernel sizes, expansion ratios) and a search strategy (Reinforcement Learning, Evolutionary Algorithms) to find an architecture that maximizes a reward function. This reward function is often multi-objective, balancing accuracy against computational costs like FLOPs or direct latency measurements on target hardware. EfficientNet utilizes a compound scaling method derived from NAS observations, while MobileNetV3 employs a platform-aware NAS (NetAdapt) to tailor the network specifically for mobile CPU constraints. [1]

2.4. *Performance Metrics in Object Recognition*

Evaluating object recognition models requires the following metrics to measure how well they identify and classify objects.

- **Accuracy:** Top-1 accuracy shows how often the model's most confident prediction matches the true class. Top-5 accuracy considers whether the correct class appears within the model's five most likely predictions. The Top-5 metric is particularly useful for datasets with many closely related categories, such as ImageNet.
- **Parameters:** This refers to the number of millions of learnable weights in the network. More parameters generally mean a higher capacity to learn complex patterns but also require more memory to store the model.
- **FLOPs (Floating Point Operations):** This measures the computational effort needed to perform a single forward pass through the network. It indicates how much processing power is needed to classify one image. Lower FLOPs mean the model is more efficient computationally.
- **Inference Time:** The duration for a model to process an image and generate a prediction, typically reported in milliseconds (ms). This is important for real-time applications like autonomous driving.

2.5. *Datasets Commonly Used*

- **ImageNet:** The most famous dataset for object recognition. The standard challenge (ILSVRC) uses over 1.2 million training images across 1,000 different object classes.
- **CIFAR-10/CIFAR-100:** Smaller datasets consisting of 32x32 pixel images in either 10 or 100 classes. They are useful for quickly testing new ideas and architectures without requiring massive computational resources.
- **COCO (Common Objects in Context):** While ImageNet focuses on classifying a single main object, COCO contains images with multiple objects in complex everyday scenes. It is widely used for more advanced tasks like object detection (finding bounding boxes) and segmentation.

3. Recent Deep Learning Architectures for Object Recognition

3.1. ResNet-50

ResNet, introduced by [9], addresses the degradation problem in deep convolutional neural networks, where adding more layers can lead to worse accuracy. As network depth increased, accuracy saturates and then degrades rapidly, not due to overfitting, but due to the difficulty in optimizing deep models where gradients vanish during backpropagation. The key idea behind ResNet is *residual learning*. Instead of learning a direct mapping, the network learns a *residual function*

$$F(x) = H(x) - x$$

, where H(x) is the desired output and x is the input. The final output becomes F(x)+x.[12] This approach makes learning easier, especially when the optimal mapping is close to the identity.

To enable residual learning, ResNet uses skip connections that add the input of a block directly to its output. This helps gradients flow through the network more effectively and helps maintain stable training in very deep networks. A standard residual block usually consists of two or more convolutional layers, each paired with batch normalization and a ReLU activation function. When input and output sizes differ, a projection layer is used; otherwise, the shortcut connection is an identity mapping. This design allows ResNet architectures (ResNet-34, ResNet-50) to become significantly deeper without suffering performance loss. With transfer learning, ResNet models can also be adapted efficiently to a wide range of classification tasks.

Specifically, ResNet-50 utilizes a ”bottleneck” block design to enhance computational efficiency. Each block is composed of three consecutive convolutional layers:

1. A 1×1 convolution that lowers the feature map’s dimensionality.
2. A 3×3 convolution responsible for the main feature extraction.
3. A 1×1 convolution that brings the dimensionality back to its original size.

This structure reduces the number of parameters and matrix multiplications, enabling the training of deeper networks without prohibitive computational costs.

Context in Object Recognition: ResNet-50 is widely used not just as a standalone classifier, but as the feature extractor for more complex object detection systems like Faster R-CNN and RetinaNet [18]. Its hierarchical feature representations capture both low-level textures and high-level semantic shapes, which makes it a standard baseline for comparing new models. [18].

3.2. EfficientNet-B0

[20] introduced EfficientNet to improve both accuracy and efficiency when scaling CNNs. Traditional approaches increased network depth, width, or input resolution independently, often leading to decreased performance. EfficientNet employs compound scaling, which uniformly increases network depth, width, and input resolution using a single scaling factor ϕ.:

- Depth: $d = \alpha^{\phi}$
- Width: $w = \beta^{\phi}$
- Resolution: $r = \gamma^{\phi}$

Here, α, β, γ are constants found through grid search and must satisfy

$$\alpha * \beta^2 * \gamma^2 = 2$$

This ensures that as the network grows, the receptive field and channel capacity increase in proportion to the input resolution. The authors show that balanced scaling allows the model to capture more meaningful detail than scaling a single dimension alone.

EfficientNet-B0 is constructed using Mobile Inverted Bottleneck Convolution (MBConv) blocks, first introduced in MobileNetV2. Instead of the traditional ResNet bottleneck structure (wide → narrow → wide), MBConv adopts an inverted layout (narrow → wide → narrow) and uses depthwise separable convolutions to greatly reduce the number of parameters. EfficientNet also incorporates Squeeze-and-Excitation (SE) modules within these blocks. SE components capture relationships between channels by first compressing global spatial information through average pooling and then reweighting the channels with a simple gating function (sigmoid). This mechanism helps the network highlight important features while suppressing less relevant ones, adding strong representational benefits at very low computational cost [20].

The EfficientNet family consistently outperforms earlier CNN architectures and transfers well to new datasets, achieving high accuracy with fewer parameters. EfficientNet-B0 serves as the base model produced through neural architecture search, designed to balance accuracy and computational efficiency. It relies on MBConv blocks and SE modules to effectively capture channel-wise interactions.

Context in Object Recognition: EfficientNet-B0 is particularly valuable in resource-constrained environments that demand high accuracy, such as cloud-based inference APIs. It achieves superior ImageNet accuracy compared to ResNet-50 while reducing parameter count by an order of magnitude, demonstrating that rational scaling is more effective than simply adding layers.

3.3. MobileNetV3

MobileNet is designed for efficient mobile and embedded hardware vision tasks. It uses depthwise separable convolutions and inverted residual blocks with linear bottlenecks,

MobileNetV3 is explicitly engineered for mobile and embedded vision applications, prioritizing low latency and energy efficiency. It synthesizes automated search techniques with novel architectural components to optimize the trade-off between speed and accuracy [10].

The architecture is based on Depthwise Separable Convolutions, which split a conventional convolution in two:

1. A single spatial filter is applied independently to each input channel.
2. A 1×1 convolution combines the outputs across channels linearly.

This factorization significantly reduces the computational cost. MobileNetV3 further optimizes this structure using Platform-Aware Neural Architecture Search (NAS) via algorithms like NetAdapt, which tunes the network layer-by-layer for specific hardware latencies. Furthermore, it introduces the *Hard-Swish* activation function, defined as

$$x \cdot \frac{\mathrm{ReLU6}(x+3)}{6}$$

. This function approximates the Swish non-linearity

$$x \cdot \mathrm{sigmoid}(x)$$

but uses piecewise linear operations that are computationally cheaper.

Context in Object Recognition: MobileNetV3 is the standard for on-device object recognition tasks, such as real-time face detection and augmented reality. Its design ensures that complex inference can be performed locally on CPUs or DSPs without draining the device's battery

3.4. Vision Transformer(ViT)

The Vision Transformer represents a change from convolution-based models to architectures centered on self-attention. Instead of processing images through convolutional filters, ViT divides an image into fixed-size patches and treats them as a sequence, similar to words in natural language processing. A standard Transformer encoder then models relationships between patches, enabling the network to capture dependencies. When trained on large datasets and subsequently fine-tuned for particular tasks, ViT outperforms CNNs. Recent research has explored hybrid models and techniques like knowledge distillation to combine the strengths of Transformers and CNNs. [8]

The Vision Transformer (ViT), introduced by [5], avoids the inductive biases of CNNs in favor of a pure Transformer architecture.

ViT treats an image as a series of patches instead of a traditional pixel grid. The input image $x \in \mathbb{R}^{H \times W \times C}$ is divided into N fixed-size patches, which are flattened and linearly projected into embedding vectors.

1. **Patch Position Embedding:** Since the Transformer implies no inherent notion of spatial order, learnable position embeddings are added to the patch embeddings to retain positional information.
2. **Self-Attention:** The core mechanism is Multi-Head Self-Attention (MSA). This allows the model to capture global dependencies across the entire image in a single layer [21].

Context in Object Recognition: Vision Transformers (ViT) perform particularly well in large-scale learning scenarios. They may lag behind CNNs on smaller datasets because they lack built-in inductive biases, but they scale efficiently. When trained on extensive datasets, ViTs capture global context better than CNNs, making them ideal for high-accuracy recognition tasks when ample data and computational resources are available.

4. Discussion

Table 1 summarizes key differences in accuracy, parameter efficiency, and computational cost (FLOPs) across representative object recognition architectures evaluated on ImageNet. These results illustrate the central trade-off between **efficiency** (parameter count) and **performance** (Top-1 accuracy).

Table 1. Comparison of Object Recognition Architectures on ImageNet

Model	**Params (M)**	**Top-1 Acc. (%)**	**FLOPs**
Inference Latency GPU ms	**Inference Latency Mobile ms**		
ResNet-50	76.1%	25.6	4.1
4	25		
EfficientNet-B0	77.1%	5.3	0.39
1.5	5		
MobileNetV3-Large	5.4	75.2%	0.22
2.3	0.7		
ViT-Base/16	77.9%	86.0	17.6
10	N/A		

[13]

4.1. Efficiency and Performance

Modern CNN architectures demonstrate better efficiency compared to earlier models. **EfficientNet-B0** delivers higher accuracy than **ResNet-50** while using nearly five times fewer parameters. **MobileNetV3-Large** offers similar efficiency and is tailored for low-latency inference on mobile and embedded hardware.

MobileNetV3-Large, while slightly less accurate than EfficientNet-B0, has a better FLOPs efficiency (0.22B vs 0.39B). Its architecture avoids operations that are practically slow on mobile hardware.

ResNet-50 is inefficient in terms of parameters (25.6M), but its dense convolution structure is highly optimized on GPUs (via cuDNN), often resulting in faster training than EfficientNet.

ViT-Base is computationally heavy, but shows peak performance. When data availability is unconstrained, ViT scales better than CNNs. CNN performance tends to plateau as model size increases, whereas Transformers continue to improve with data and scale, exhibiting a ”scaling law” behavior.

The **Vision Transformer** achieves the highest accuracy, but requires significantly more parameters. This is because models that scale well with data typically incur larger computational costs.

4.2. Architectural Differences: CNNs vs. Vision Transformers

The difference between CNNs and Vision Transformers is due to their **inductive biases**.

- **CNNs** embed biases such as locality and translation invariance, enabling strong generalization even with moderate training data. This makes models like ResNet, EfficientNet, and MobileNet effective for both standard datasets and transfer learning.
- **ViTs** treat images as sequences of patches and do not encode spatial relationships by design. As a result, they require **large-scale pre-training** to learn these patterns, but once trained, they can generalize extremely well [5].

4.3. Deployment and Scaling Considerations

Table 2. Deployment and Scaling Characteristics of CNNs and ViTs

Factor	**Efficiency-Oriented CNNs**	**Scalability-Oriented ViTs**
Primary Strength	Low parameter count and low latency.	High performance with large data and models.
Data Requirements	Perform well on standard datasets	Require massive pre-training datasets.
Performance Scaling	Accuracy saturates as model size grows.	Accuracy improves steadily with scale [22].
Typical Deployment	Edge and mobile devices.	Data centers and high-resource environments.

CNNs are preferred when efficiency and accessibility are considered. ViTs, while computationally demanding, are well suited for large-scale vision tasks because of their scalability. [24].

4.4. Suitability for different domains

4.4.1. Medical Imaging

In this domain CNNs are prefered. Their strong inductive biases (locality, translation invariance) allow them to learn effective features from small datasets without overfitting. For tasks like skin cancer classification or pneumonia detection from X-rays, EfficientNet-B0 and ResNet-50 often outperform ViTs. A comparative study on DermaMNIST showed ResNet-18 achieving 81.25% accuracy compared to ViT-Base’s poor performance due to the lack of sufficient training data. [3]

4.4.2. Autonomous driving

For on-vehicle perception (detecting lanes, signs, pedestrians), MobileNetV3 and EfficientNet are the industry standards. Their low latency allows them to run on embedded automotive grade chips at high frame rates. [23] However, for the primary perception stack, the industry is moving toward Transformer-based models. While purely vision-based ViTs are too slow for some real-time loops, hybrid models like DETR (Detection Transformer) are becoming standard for 3D object detection. [14]

5. Conclusions and future work

This survey compared four distinct architectures for object recognition. ResNet-50 established the foundation for deep networks, while EfficientNet and MobileNetV3 demonstrated that optimizing depth, width, and resolution can have impressive results with a fraction of the computational cost. The Vision Transformer marked a paradigm shift, proving that self-attention mechanisms can replace convolutions for large-scale tasks.

Currently, CNNs (like EfficientNet) remain highly competitive for general-purpose use due to their efficiency and ease of training. However, Transformers dominate in high-compute, massive-data regimes.

Future Work: Research is moving toward architectures that integrate CNNs' capturing local features while incorporating the global context learned by Transformers. Additionally, self-supervised learning (methods that learn without human labels) and multimodal models (learning from both text and images, such as CLIP) represent the next frontier in creating robust, general-purpose recognition systems [17].

6. Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author used Gemini in order to correct the grammar and the logical flow of sentences. AFter using this tool, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

References

[1] , . EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling — research.google. https://research.google/blog/efficientnet-improving-accuracy-and-efficiency-through-automl-and-model-scaling/. [Accessed 07-12-2025].

[2] Ajit, A., Acharya, K., Samanta, A., 2020. A review of convolutional neural networks, in: 2020 international conference on emerging trends in information technology and engineering (ic-ETITE), IEEE. pp. 1–5.

[3] Amangeldi, A., Taigonyrov, A., Jawad, M.H., Mbonu, C.E., 2025. Cnn and vit efficiency study on tiny imagenet and dermamnist datasets. arXiv preprint arXiv:2505.08259 .

[4] Ankile, L.L., Heggland, M.F., Krange, K., 2020. Deep convolutional neural networks: A survey of the foundations, selected improvements, and some current applications. arXiv preprint arXiv:2011.12960 .

[5] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al., 2021. An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations.

[6] Elngar, A.A., Arafa, M., Fathy, A., Moustafa, B., Mahmoud, O., Shaban, M., Fawzy, N., 2021. Image classification based on cnn: a survey. Journal of Cybersecurity and Information Management 6, 18–50.

[7] Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press. http://www.deeplearningbook.org.

[8] Han, K., Wang, Y., Chen, H., Chen, X., Guo, J., Liu, Z., Tang, Y., Xiao, A., Xu, C., Xu, Y., et al., 2022. A survey on vision transformer. IEEE transactions on pattern analysis and machine intelligence 45, 87–110.

[9] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770–778.

[10] Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al., 2019. Searching for mobilenetv3, in: Proceedings of the IEEE/CVF international conference on computer vision, pp. 1314–1324.

[11] Khan, A.A., Laghari, A.A., Awan, S.A., 2021. Machine learning in computer vision: A review. EAI Endorsed Transactions on Scalable Information Systems 8.

[12] Liang, J., 2020. Image classification based on resnet, in: Journal of Physics: Conference Series, IOP Publishing. p. 012110.

[13] Naminas, K., . Image Classification Models: Top Picks for Your ML Pipeline — labelyourdata.com. https://labelyourdata.com/articles/image-classification-models. [Accessed 07-12-2025].

[14] Nazeri, A., Zhao, C., Pisu, P., 2024. Evaluating the adversarial robustness of detection transformers. arXiv preprint arXiv:2412.18718 .

[15] O'shea, K., Nash, R., 2015. An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458 .

[16] Pinaya, W.H.L., Vieira, S., Garcia-Dias, R., Mechelli, A., 2020. Convolutional neural networks, in: Machine learning. Elsevier, pp. 173–191.

[17] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al., 2021. Learning transferable visual models from natural language supervision, in: International conference on machine learning, PMLR. pp. 8748–8763.

[18] Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, pp. 91–99.

[19] Suetens, P., Fua, P., Hanson, A.J., 1992. Computational strategies for object recognition. ACM Computing Surveys (CSUR) 24, 5–62.

[20] Tan, M., Le, Q., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks, in: International conference on machine learning, PMLR. pp. 6105–6114.

[21] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need, in: Advances in neural information processing systems, pp. 5998–6008.

[22] Zhai, X., Kolesnikov, A., Houlsby, N., Beyer, L., 2022. Scaling vision transformers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12104–12113.

[23] Zhang, J., Cao, J., Chang, J., Li, X., Liu, H., Li, Z., 2023. Research on the application of computer vision based on deep learning in autonomous driving technology, in: INTERNATIONAL CONFERENCE ON WIRELESS COMMUNICATIONS, NETWORKING AND APPLICATIONS, Springer. pp. 82–91.

[24] Zhang, Y., Tiňo, P., Leonardis, A., Tang, K., 2021. A survey on interpretable and explainable deep learning for computer vision. arXiv preprint arXiv:2108.06842 .