FEDERAL UNIVERSITY OF TECHNOLOGY, AKURE
JOURNAL OF ENGINEERING AND ENGINEERING TECHNOLOGY

REVIEW OF CONVOLUTIONAL NEURAL NETWORK MODEL FOR OBJECT DETECTION, LOCALIZATION AND CLASSIFICATION

[1] Adejumobi, I. O., [2] Adejumobi, P. O.*, [3] Yesufu, T. K.

[1] Department of Mechanical and Mechatronics Engineering, College of Engineering, Afe Babalola University, Ado-Ekiti PMB 5454, Nigeria.

[2] Department of Technology Development, Raw Materials Research and Development Council, Abuja, Nigeria.

[3] Department of Electronics and Electrical Engineering, Faculty of Technology, Obafemi Awolowo University, Ile-Ife, PMB 5538, Nigeria

*Corresponding author: Adejumobiio@abuad.edu.ng

Adejumobi, I. O., Adejumobi, P. O., Yesufu, T. K. (2026): Review of Convolutional Neural Network model for Object Detection, Localization and Classification. FUTA Journal of Engineering and Engineering Technology 20 (Special), 25-34

Received Date: 15.01.2026 **Accepted Date: 10.02.2026**

Abstract

Machine learning Algorithms have been used as central components in contemporary technological design and image processing to identify, locate, detect and classify items. Deep learning algorithms and data structures can be a powerful tool for this. This paper reviews several deep learning algorithms using convolutional neural networks (CNNs) blocks to Detect, Classify and Localize objects. Various convolutional architectures are elaborated, detailing how certain configurations of convolutional blocks affect performance in detection, as well as strengths and weaknesses. This study aims to provide insight into the use of various CNN models which guide developers in selecting and developing effective models or algorithms for automatic detection, localization and classification systems.

Keywords: *YOLO, Convolutional blocks, Machine learning, Localization, Detection, Classification*

Introduction

Machine learning is considered to be the solution to allow safe automated systems in the complex and dynamic machines we have today (Adejumobi and Ojo, 2021). Machine learning essentially is the process of teaching machines how to do things that would otherwise be done by humans so that less work is required of human operators (Adebisi *et al.,* 2024). This is done via supervised, semi-supervised or unsupervised learning, in which machines learn from previously experienced data.

Machine learning has wide applications in other technological innovations such as spam detection, multimedia idea retrieval, image classification, video recommendation, text mining and many others. It uses different machine learning techniques that learn features of the data at different levels of abstractions, thus improving the artificial intelligence field.

Supervised and unsupervised learning use a training data set (Adejumobi *et al.,* 2023). The supervised algorithms train until they minimize probability error and have a sufficient level of confidence. They can be further grouped within anomaly detection, dimension reduction, and regression. To form subgroups within the data depending on the similarity of the data, a learning algorithm that is not supervised uses the inputted data to establish relationships between them and it is usually classified into two main categories: clustering and association rule learning. These machine learning algorithms are powerful and provide useful tools for image processing, feature extraction, object classification, and object detection. Images comprise pixels, or picture elements, which are square and arranged in rows and columns in a two-dimensional array, or matrix, which must be handled by machine learning algorithms. Image processing is application of technique to an image for the purpose of improving or capturing important information from the image. The procedure of transmuting raw data into a simplified version for decision-making (such as detection of object, classification of object and object localization or recognition (Elnemr et al., 2016)). is known as feature extraction. Dissimilar kinds of machine learning models have been engaged to improve images and to allow for improved object detection and image localization in computer vision tasks (Diwan *et al*., 2022). Object classification is the process in computer vision by which an object is classified according to its visual appearance. In image processing, object classification entails categorizing pixels into classes, groups, or categories according to spectral properties, which are stored as digital numbers. For instance, an image classification algorithm can see whether or not an image encompasses a human figure (Ponnusamy *et al.,* 2017) by classifying pixels into classes. After extracting feature descriptors at the interest points, the object categories to which the objects belong are identified from a known set of features by matching the descriptors; this is called object localization. Then, the identified object is attempted to be matched perfectly through all frames of the video or blended bigger data set others noted frames and time durations (Patro and Boddupally, 2016). Other examples of object detection may include face detection, skeleton detection or pedestrian detection among other subtasks. The aim is to detect known categories of objects in an image, such as faces, cars, or humans. The aim is to identify known class object instances in an image, such as people, cars or faces (Amit *et al.,* 2020). Object recognition forms part of the basic elements of computer vision, and is applied in the semantic understanding of videos and images. It is used in face recognition, image classification, human behavior analysis and autonomous driving (Zhao *et al.,* 2019).

Detection of object can be divided into two groups: single-shot object detector and two-stage object detector. The single-shot approach predicts the presence and placement of objects by processing the entire image in a single pass. It processes images efficiently using fully CNNs. You Only Look Once (YOLO) and the Single Shot MultiBox Detector (SSD) are two examples. Figure 1 exhibits a single-shot object detector, while Figure 2 shows a two-stage detector. In the two-stage approach, a Region Proposal Network (RPN) creates regions of interest in the first stage, then bounding-box regression and classification of object are applied to the region suggestions in the second stage. These models tend to be slower even though they reach excellent accuracy. Region-based Convolutional Neural Networks (R-CNN), Mask R-CNN and Faster R-CNN are a few examples (Girshick, 2015).

These are the key ideas and progressions in machine learning and computer vision, especially as they relate to image processing. In designing an automated system or a machine, choosing the right machine learning model is crucial to improve performance, efficiency, and precision (Adejumobi *et al*., 2025). Despite this, the hundreds of existing CNNs create challenges in determining which models are well suited for tasks. This paper aims at addressing this challenge by investigating various CNN architectures for machine learning in automated systems. It also provides important information and practical considerations for researchers and practitioners when selecting and designing optimized CNN models for various automated system and machine applications.

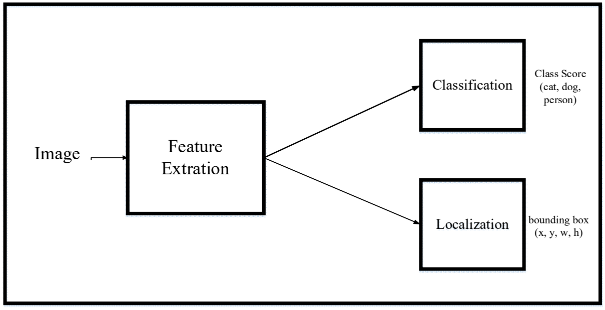

Figure1: Single stage object detection

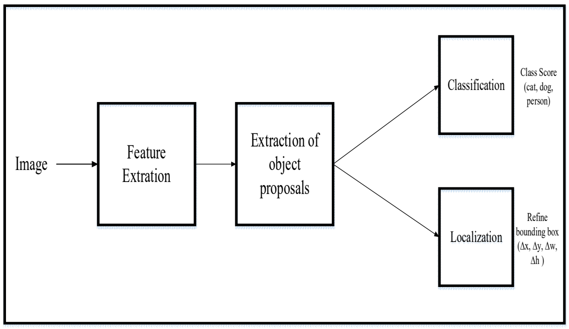
Image
Feature Extration
Extraction of object proposals
Classification
Class Score (cat, dog, person)
Localization
Refine bounding box (Δx, Δy, Δw, Δh)

Figure 2: Two stage object detection

Literature review

Machine learning algorithms are very important in various artificial intelligence applications, providing reliable and efficient solutions that replicate human functions. These algorithms are designed to be fast and effective. They encompass a range of methods, including regression, decision tree algorithm, support vector machine algorithms and logistic regression (Jayanthi, 2022), among others.

Deep learning (DL) has significantly advanced computer vision with remarkable achievements in natural language processing, speech recognition, object detection, picture classification, image segmentation, video processing, and many other domains (Bhatt, 2021). However, deep reinforcement learning (DL) algorithms with automated feature extraction, like Convolutional Neural Networks (CNN), Region-Convolutional Neural Networks, Fast R-CNN, and Deep Reinforcement Learning, eliminate the need for significant human intervention and domain knowledge.

(Alzubaidi et al., 2021), describe these algorithms as having an architecture of a multi-layer data illustration where the ending layers excerpt high-level information while the first-level layers of the deep learning algorithm capture the low-level features. The convolution layer's features are generalized by the CNN model, enabling networks to identify these features independently.

In Convolutional Neural Networks, the primary task involves comparing pixel values to identify objects, shapes, and edges within the input data. This approach reduces computation in convolutional networks. Processing in a Convolutional Neural Network involves three fundamental steps: receiving input, processing information, and generating output (Singh, 2020). The input, hidden, and output layers, as well as their component stages known as neurons, reflect the convolutional layers. There are two primary processes in the training phase: backward propagation and forward propagation. During forward propagation, input images or data, represented as numerical values (e.g., pixel intensities in an image), are processed by hidden layer neurons through mathematical operations. The results are sent to the output layer, which generates the ultimate predictions. In backward propagation, the network assesses the closeness of the output to the actual values. Based on the degree of closeness, it estimates the error amongst the final output and the real values and updates parameter values accordingly. This process is iteratively repeated with updated parameter values to generate new outputs. A Convolutional Neural Network consists of two primary components: the convolution layers, responsible for extracting feature from the input, and the fully connected (dense) layers, that generate the final output based on the features extracted by the convolution layer, as depicted in Figure 3.

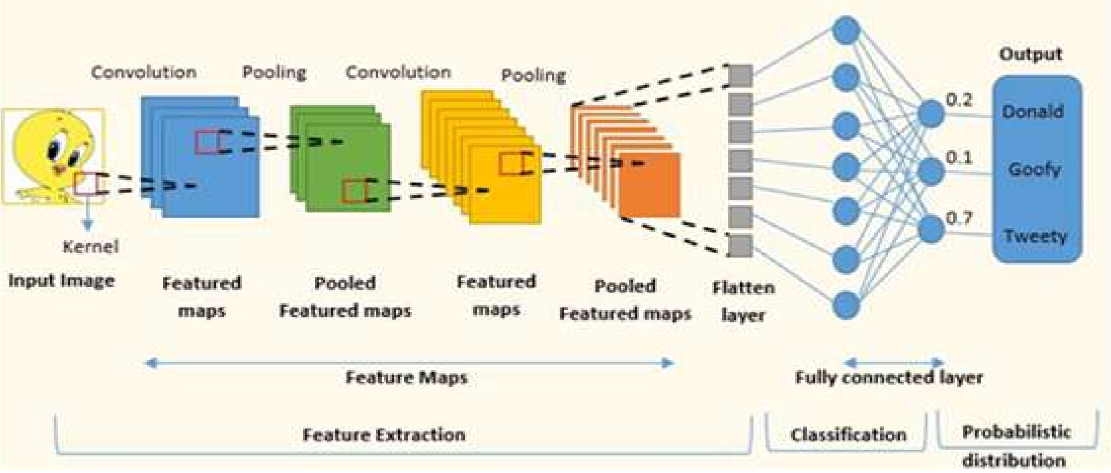

Figure 3: Overview of convolutional neural network

Single Stage Convolutional Neural Network Algorithms

Convolutional neural network models' architecture, according to Eshaq *et al.*, (2021), is centred on how layers are arranged to maximize the network's functional capabilities and eventually increase performance.

LeNet is a multi-layer network that is well-suited for image recognition tasks due to its capability to learn from high-dimensional and highly complex data. The architecture of LeNet-5 has eight layers, of which five are convolutional and three are fully connected as indicated by Indolia *et al.*, (2018). LeNet can be considered the first architecture of a CNN. Application: character recognition on handwritten documents.

AlexNet was the first successful large scale convolutional neural network, which received a lot of attention by winning the ImageNet classification challenge (Swapna *et al.,* 2020). Its architecture is composed of three max pooling layers, five convolutional layers, two fully connected layers, one softmax layer and two normalization layers. Individual convolutional layer contains a nonlinear activation function ReLU and a convolutional filter, as indicated by Thalagala and Walgampaya (Thalagala and Walgampaya, 2021). Application: fabric defect detection.

An architecture that comprised of 13 convolutional layers, 3 fully connected layers with ReLU activation is referred to as Visual Geometry Group Network (VGG) . VGG-16, in particular, boasts a greater number of layers in comparison to AlexNet and employs smaller filters. An even more extensive variant, VGG-19, has also been developed. VGG-16 stands as one of the widely employed architectures in object detection, having achieved notable performance levels. It is utilized in various algorithms, such as Fast R-CNN, as mentioned by Benali and Amrouch (Benali and Amrouch, 2020). Application include Object recognition.

MobileNet, as described by Boonsirisumpun and Surinta (2022), is a CNN model designed with the intention of reducing the size of a standard CNN model, thus making it suitable for use on mobile devices. The key concept behind MobileNet is the replacement of typical convolutional filters with two distinct steps: depthwise and pointwise separable convolutions, resulting in a significantly smaller convolution filter. MobileNet consists of 27 convolution layers, with 13 depthwise convolution layers, one average pooling layer, one fully connected layer, and one softmax layer. Another version, MobileNet-v2, has been introduced for object detection purposes, incorporating residual blocks to allow shortcut connections directly between the bottleneck layers (Elharrouss *et al.,* 2022).

GoogLeNet: Boonsirisumpun and Surinta (2022) also described GoogLeNet as a novel architecture known as the Inception network. In its initial version, GoogLeNet consisted of 22 convolution layers and introduced an inception module. Each inception module is a composite structure comprising various sizes for each convolution node, as well as a max-pooling node. GoogLeNet (Inception V1) emerged as the winner at ILSVRC14, surpassing the performance of VGGNets in object detection. Application include mobile vision tasks (e.g., Google optimization).

ResNet-152: Residual networks (ResNet) were introduced as a family of deep neural networks, each sharing a similar structure but varying in depth. The key innovation brought by ResNet is the incorporation of residual learning units, designed to mitigate the challenges associated with deep networks. ResNet is a feedforward network featuring shortcut connections that introduce new inputs into the network and yield fresh outputs. One of the unique advantages of ResNet is its ability to augment classification accuracy without a substantial increase in model complexity (Nguyen *et al.,* 2018). Application include Image detection and classification.

Darknet serves as the foundational structure for the object detection method known as YOLO. The Darknet-19 architecture resembles VGG-19, featuring 19 convolution layers, but incorporates a unique inception-like module. Darknet-53, used as the backbone for YOLOv3, houses 53 convolution layers and includes residual connections, similar to Inception-ResNet (Boonsirisumpun and Surinta, 2022).

Cross Stage Partial Network (CSPNet): Wang et al. CSPNet (Wang et al., 2020) is an architecture that divides the feature map of the base layer into two sections. This helps propagate the gradient flow by splitting and later merging the flows through a cross-stage hierarchy. CSPNet is capable of improving both speed and accuracy of object detection task while decreasing the computational complexity of the network.

YOLO (You Only Look Once) is a well-known real-time object detection technology and deep learning model that is found on Fully Convolutional One-stage Object Detection (FCOS). Instead of splitting the images into multiple regions, YOLO was fast and simple, performing bounding box predictions simultaneously. This allows YOLO models to achieve real time detection of multiple objects within an image (Terven, 2023). For YOLO, images are split into a grid, for apiece grid cell, class likelihood and bounding box predictions are produced. There are five values for each bounding box prediction: Pc, bx, by, bh, and bw. Where, Pc is the confidence score, bx and by are center coordinates, and bh and bw are box dimensions. It is commonly outputted as a tensor which can further be filtered for duplicate detections using non-maximum suppression (NMS) (Terven, 2023). The mathematic derivation of YOLO is illustrated in equation (1).

$$\mathrm{S}\times\mathrm{S}\times(B\times 5+C) \qquad (1)$$

Where is the grid division, C is the confidence score of prediction and B is the bounding box. The ratio of the intersection area to the union area of the ground truth bounding box and the anticipated bounding box is known as the Intersection over Union, or IoU. The overlay between the projected bounding boxes and the ground truth is measured.

Two-Stage Convolutional Neural Network Algorithm

Region-Based Convolutional Neural Network (R-CNN)*:* R-CNN detects objects present in an input image by using a deep ConvNet that proposes thousands of bounding boxes. Region proposals are generated using selective search which proposes multiple regions of a single image. In a more sophisticated type of object detection called selective search, a sliding window and image pyramid is used to sample each region of interest (ROI) in an image. Uijlings et al. (2013) refined a superpixel based selective search method by using the data collected on image structure to guide the sampling strategy such that it would sample some possible location of an object. Thetechnique over-segments an image based on colour, texture, size, moreshape and a final meta-similarity which is a linear combination ofsimilarity metrics. An object position could then be reseourced using a sliding window, sliding left to right and top down at each level of the image pyramid.

Faster R-CNN: Espinosa et al. (2017) presented the architecture of the Faster R-CNN model, a deep convolutional neural network which contains two main integral parts: a Region Proposal Network (RPN) and an Object Detection Network. It starts with the pre-training of the complete image using numerous max-pooling and convolutional (conv) layers that would eventually output a convolution feature map. In this phase, the max-pooling layer is substituted with a Region of Interest (RoI) layer. The network is then modified, as the connected layers and softmax classifier are replaced with detection specific layers. All RoIs suggested by the RPN undergo RoI pooling, where a fixed-length feature vector is obtained for each from the feature map. Both these feature vectors are fed into fully connected (fc) layers that split in two; one for classifying the background and the other one for predicting the bounding box, as originally proposed by Girshick, (2015).

Convolution Model and Architectures

Convolution Layer in CNN: The convolution layer's features are generalized by the CNN model, enabling networks to identify these features independently. In Convolutional Neural Networks (CNNs), the primary task involves comparing pixel values to identify objects, shapes, and edges within the input data. This approach reduces computation in convolutional networks. Processing in a CNN involves three fundamental steps: receiving input, processing information, and generating output (Singh, 2020). The input, hidden, and output layers as well as their component stages, known as neurons, reflect the convolutional layers. The training procedure comprises of two core phases: backward propagation and forward propagation. During forward propagation, input images or data, represented as numerical values (e.g., pixel intensities in an image), are processed by hidden layer neurons through mathematical operations. The results are passed to the output layer, which generates predictions. In backward propagation, the network evaluates the closeness of the output to the actual values, estimates the error, and updates parameter values accordingly. This process is iteratively repeated with updated parameters until optimal outputs are achieved. A CNN includes two types of layers, the convolution layers that are used to excerpt features and the fully connected (dense) layers which will output the final result based on the features extracted. The convolution layer is basically the first layer where features of the input image are extracted. It is a mathematical convolution of the input image with a pre-defined filter of certain size. The filter is convolved over the input image, computing a dot-product amongst the filter and the image section where it is placed. This produces a feature map that conveys the presence of certain characteristics of the image such as corners and edges. This feature map is then used by the next layers to continue learning features. Convolution layers have the advantage of maintaining spatial relationship between the pixels. (Indolia et al., 2018; Bezdan and Džakula, 2019). The parameters of a convolution layer revolve around the use of trainable kernels, which convolve across the spatial dimensions of the input to produce 2D activation maps (O'shea and Nash, 2015). These kernels generally have small spatial dimensions but span the entire depth of the input map. After training, the activation maps can be visualized to show distinctive features such as patterns in numerical digits. Each kernel generates an activation map, and stacking these maps along the depth dimension forms the full output volume of the convolution layer. Convolution kernels divide an image into small segments, known as receptive fields, which help extract feature patterns. Kernels interact with receptive fields through weighted multiplication, producing outputs that capture essential image characteristics (Bhatt, 2021). The convolution operation can be stated mathematically as shown in Equation (2).

$$f_l^k(p,q) = \sum_c \sum_{(x,y)} i_c(x,y).e_l^k(u,v) \qquad (2)$$

Where, $i_c(x,y)$ is an element of the input image tensor I_c, this is element wise multiplied by $e_l^k(u,v)$ index of the k^{th} convolutional kernel k_l of the l^{th} layer. Whereas output feature-map of the k^{th} convolutional operation can be expressed as in Equation (iii)

$$F_l^k = [f_l^k(1,1),......,f_l^k(p,q),.....,f_l^k(P,Q)] \qquad (3)$$

Where, F_l^k is input feature matrix for l^{th} layer and k^{th} neuron, (p,q) element of feature matrix, (x,y) element of c^{th} channel of an image, (u,v) element of k^{th} kernel of l^{th} layer.

Pooling Layer in CNN: Generally, a pooling layer trails a convolutional layer. Its main objective is to decrease the spatial dimensions of the convolved feature maps, thereby lowering computational costs while retaining vital information. This reduction is achieved by analyzing each feature map independently and minimizing inter-layer connections. Essentially, pooling combines and condenses the features extracted by convolution layers [31].

There are several types of pooling operations, depending on the aggregation technique used:

Max Pooling: Chooses the maximum value from each sub-region of the feature map.

Average Pooling: Calculates the average of values within a sub-region.

Sum Pooling: Computes the total sum of values in the specified region (Zhang *et al.,* 2021).

Pooling layers typically serve as a transition between convolutional layers and fully connected (FC) layers, ensuring that critical features are preserved while reducing data size. The pooling operation can be mathematically expressed as shown in Equation (4).

$$z_l^k = g_p(F_l^k) \qquad (4)$$

Where, z_l^k signifies the pooled feature-map of l^{th} layer for k^{th} input features-map F_l^k g_p (.) defines the kind of pooling operation.

Filter: A filter is a single template or pattern that finds similarities between different places or regions of the input image and the stored template when it is convolved across the input. The filter focuses on a small part of the image at a time. Equations (5) and (6) shows the convolution of an input channel and a $3x3$ filter.

Let

$$B = \begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} \qquad (5)$$

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \qquad (6)$$

Then, we sum up the pairwise product (dot product) to give equation (7) which is the output channel called the feature map.

$$C = a_{11}b_{11} + a_{12}b_{12} + \dots\dots\dots\dots\dots\dots + a_{33}b_{33} \qquad (7)$$

Where A is the input channel, B is the filter and C is the feature map.

Activation Function: During forward propagation convolution is performed over the height and width of the input volume by each filter, estimating the dot products of the input with the entries of the filter at different locations. The outcome is then subjected to a nonlinear activation function, for instance the hyperbolic tangent (tanh), sigmoid, or rectified linear unit (ReLU). The result of these operations is called feature maps or activation maps, which reflects the response of the filter at each spatial location (Kuo, 2016).

A CNN is primarily interested in understanding how the input is related to the output and thus encodes this learned experience in the filter weights. One of the difficulties in interpreting CNNs is to understand the nonlinear activation unit applied after the convolutional operation (Cepni, 2020). This layer is primarily used to introduce non–linearity to the linear information mined from the convolutional layers in order to allow the network to learn to map complex functions and to speed up the training of the network as a whole (Redmon *et al.,* 2017).

Batch Normalization: Batch normalization is a method to increase speed and control overfitting used within neural networks of data science. It involves preprocessing data into batches of numerical data, ensuring a common scale without altering its shape, as opposed to treating it as a single output. Batch normalization can be applied to data before model training.

Concatenation: Concatenation combines multiple columns into a single vector-valued column, which can significantly expedite data processing, predominantly when handling a large number of columns, ranging from hundreds to thousands. Deep learning models often require extensive datasets to achieve high performance. When the dataset comprises millions of rows (observations/instances), fitting the entire dataset into a computer's memory becomes nearly impossible. In such cases, each training step would be time-consuming and computationally intensive if the entire dataset were used for every gradient update during training. To address these challenges, data is processed in batches, which represent portions of the complete dataset.

Fully connected layer: CNN classifier is situated at the end of each individual CNN architecture in the Fully Connected Layer. The Fully Connected (FC) layer corresponds to connections between neurons in the layer and operates as a feed-forward artificial neural network (ANN) performing functions similar to a multi-layer perceptron neural network. The input to the FC layer originates from the latter pooling or convolutional layer; it is represented as a vector that results from a flattening of the feature maps. The output of the FC layer represents the final output of the CNN. Among other things, this fully connected layer performs two important transformations on the data that it receives: a linear transformation and a non-linear transformation.

Experimental Findings: This study presents a comprehensive review of 97 peer-reviewed journal articles published in reputable outlets such as Springer, Elsevier, and Scopus-indexed journals as shown in figure 4. The review period extended from 1998 to 2025.

These articles primarily focus on automatic object detection, localization, and classification using convolutional neural network (CNN)-based algorithms. An initial screening was conducted to categorize the papers based on their specific areas of

Table 1: Brief description of CNN models

Model	Year	Key Features / Contributions	Applications	Reference
LeNet-5	1998	An early CNN applied to digit recognition; it pioneered deep learning for vision.	Handwritten digit recognition, document analysis	(LeCun et al., 1998)
AlexNet	2012	Introduced deep CNNs using ReLU, dropout, trained on multiple GPUs; winner of ILSVRC 2012.	Large-scale image classification, transfer learning.	(Krizhevsky *et al.,* 2012)
VGG-16/19	2014	Very deep CNNs using small 3×3 [3] kernels.	Image classification, medical imaging, object recognition.	(Simonyan *et al.,* 2014)
GoogLeNet (Inception)	2014	Deeper but more efficient inception modules introduced.	ImageNet classification, object recognition, embedded systems.	(Szegedy *et al.,* 2015)
ResNet-152	2015	Residual learning, solved the vanishing gradient problem.	Fine-grained recognition, video classification, detection	(He *et al.,* 2016)
Darknet-53	2023	Allowing deeper learning	User's surf and encrypted forms of data	(Obaidat *et al.,* 2025; Kant *et al.,* 2024)
MobileNet / v2	2017–2018	Light CNNs for mobile/edge devices.	Mobile apps, embedded vision, robotics, AR/VR	(Howard *et al*., 2017)
YOLOv1	2016	First YOLO, unified detection framework.	Real-time object detection (traffic monitoring, robotics)	(Redmon *et al.,* 2016)
YOLOv2 / YOLO9000	2017	9000 objects: joint detection + classification.	Multi-class detection, autonomous driving	(Redmon and Farhadi, 2018)
YOLOv3	2018	Multi-scale predictions, residual connections.	Surveillance, drone vision, medical imaging	(Bochkovskiy *et al.,* 2020)
YOLOv4	2020	CSPNet backbone, training tricks optimized.	Real-time security systems, industrial inspection	(Wang *et al.,*202; Bochkvoskiy *et al.,* 2020)
YOLOv5	2020	Modular, implemented in PyTorch and widely used	Research, autonomous vehicles, smart cities	(Jocher, 2020; Kant, 2024)
YOLOv6	2022	Efficient, sector-oriented deployment.	E-commerce, logistics, manufacturing	(Li *et al.,* 2023; Howard *et al.,* 2017)
YOLOv7	2022	Accuracy + speed at state-of-the-art levels.	Smart surveillance, UAV navigation, robotics	(Wang *et al.,* 2022)
YOLOv8	2023	New backbone, anchor-free detection.	Real-time analytics, smart traffic, agriculture	(Jocher *et al.,* 2023a; Jocher *et al.,* 2023b)
YOLOv9	2023	Re-parameterization, advanced backbone.	Edge AI, autonomous navigation, industrial inspection	(Li *et al.,* 2023; Xu et al., 2023)
YOLOv10	2024	Reduced redundancy, efficiency gains.	Real-time embedded AI, autonomous driving, smart IoT	(Xu *et al.,*2023)
YOLOv11	2024	Faster training, improved generalization.	Robotics, real-time monitoring, healthcare	(Ultralytics, 2024)
YOLOv12	2025	Latest YOLO, improved scalability.	AIoT, advanced surveillance, autonomous systems	(Tian *et al.,* 2024)

application. Nine papers were excluded due to their limited relevance to the objectives of this study. Furthermore, 49 papers were omitted from the main analysis as they concentrated solely on single-stage or two-stage detection frameworks, which are beyond the defined scope of this review. Nevertheless, these papers were examined to evaluate the flexibility, scalability, complexity, and frequency of use associated with such methods. Although not included in the core analysis, insights drawn from these studies informed recommendations on suitable model selections based on varying problem contexts. The remaining selected papers were thoroughly analyzed to identify the detection and classification models employed. This review also outlines the strengths and limitations of both single-stage and two-stage object detection approaches grounded in CNN architectures. Additionally, it examines the development of the YOLO (You Only Look Once) series from YOLOv1 to YOLOv12 [35, 36, 37, 39] highlighting the specific advancements introduced in each successive version.

Results and Discussion

Figure 5 illustrates the distribution of the reviewed papers regarding CNN algorithms. This work uncovers the scarcity of research specifically utilizing two-stage CNNs, only 19 amongst the 97 papers employ these two-stage methods for object detection, classification and localization. Two stage CNN techniques, which outperformed single stage methods as well as YOLO techniques, were rarely proposed. 36 of the reviewed papers addressed single stage methods and the most common were studies on YOLO approaches, totaling 42 papers.

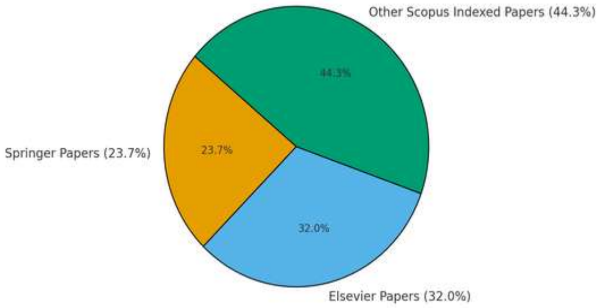

Figure 4:Distribution of Published Papers Reviewed

From the review conducted, it was clear that although two stage CNN methods perform better in dynamic environments and complex scenes for detecting, localizing, classifying objects, they require both a region finding step, and a tracking step while it also gives room for the combinations of two or more CNN architecture. The brief description of CNN models with their use is presented in the Table 1. This makes them very long and complex to run, often requiring very fast hardware to be performant. In contrast, YOLO methods can achieve similar results more quickly and in a single step, without splitting the process into region proposals, localization, and selection. This advantage led to the development of multiple improved versions of YOLO, which are also discussed in this work. Single-stage methods are suitable for systems that do not need specialized hardware and are used in simpler applications, such as phones, calculators, and wristwatches, among others. Table (1) shows the year, contributions and applications of CNN models. The contributions within this paper lay the groundwork for future bench marking and empirical research to continue to deploy CNN-based detectors, localizers and classifiers in very difficult, low-resource settings.

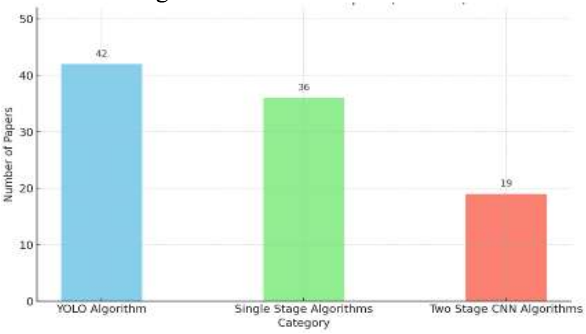

Figure 5: Distribution of Reviewed Papers for different CNN Categories

Conclusion

This paper has given a thorough rundown of various models and methods used for object recognition, localization, and classification. We investigated the strengths and weaknesses of several convolutional network topologies and highlighted the impact of design decisions on accuracy and speed. We discussed the accuracy, real time and real time capabilities of CNN models, which will help researchers and developers to choose or create models that suit the requirements of particular tasks, settings, and resources.

References

Adebisi, O., Afolayan, A., Ayoade, I., Adejumobi, P. and Adejumobi, I. (2024). Integration of deep learning techniques in mechatronic devices and systems: Advancement, challenges, and opportunities. In: *Proceedings of the International Conference on Science, Engineering and Business Driving Sustainable Development Goals (SEB4SDG)*, 1–6.

Adejumobi, I.O. and Ojo, J.A. (2021). Surveillance system for suspicious vehicular movement. *International Journal of Computer Applications Technology and Research*, 8(8), 336–340.

Adejumobi, I.O., Adejumobi, P.O., Ikumapayi, O.M., Banjoko, A.O., Olipede, D.E., Adebisi, O.A. and Yesufu, T. (2025). Exploring fault causality and predictive maintenance in PV cells: Approaches toward sustainable energy systems. *NIPES Journal of Science and Technology Research (JSTR)*, 7(2), 3223–3228.

Adejumobi, P.O., Adejumobi, I.O., Adebisi, O.A., Ayanlade, S.O. and Adeaga, I.I. (2023). Automatic classification of breeds of dog using convolutional neural network. *Nigerian Journal of Technological Development*, 20(3), 199–209.

Adejumobi, P.O., Ojo, J.A., Adejumobi, I.O., Adebisi, O.A. and Ayanlade, S.O. (2025). Development of a sorting system for mango fruit varieties using convolutional neural network. *International Journal of Computer Science and Engineering*, 28(1), 87–99.

Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O. *et al.* (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, 8(53), 1–74.

Amit, A., Kumar, V. and Sharma, P. (2020). Recent trends in object detection. *Multimedia Tools and Applications*, 79(45–46), 33423–33441.

Benali, A. and Amrouch, H. (2020). A comparative study of CNN architectures in object detection. *International Journal of Computer Vision Research*, 8(2), 15–27.

Bezdan, T. and Džakula, A. (2019). Convolutional neural networks: An overview. *Annals of Computer Science and Information Systems*, 18, 15–22.

Bhatt, D., Patel, P., Talsania, P., Patel, M. and Vaghela, S. (2021). Convolutional neural networks in computer vision: A review. *Machine Learning with Applications*, 2, 100006.

Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint*, arXiv:2004.10934.

Boonsirisumpun, S. and Surinta, O. (2022). A review of CNN models for mobile and edge devices. *Journal of Mobile Computing*, 11(2), 45–57.

Cepni, A. (2020). Activation functions in deep learning. *Neural Processing Letters*, 51, 2237–2250.

Diwan, R., Sharma, M. and Singh, S. (2022). Advances in object detection models for computer vision. *Pattern Recognition Letters*, 159, 23–31.

Elharrouss, O., Almaadeed, N., Al-Maadeed, S. and Akbari, Y. (2022). MobileNet-V2 for object detection. *Journal of Imaging*, 8(3), 72.

Elnemr, M., Ramzy, A.R. and Aly, K.N. (2016). Feature extraction techniques in image processing. *International Journal of Computer Applications*, 149(3), 24–30.

Eshaq, A., Hassan, H. and Ali, R. (2021). Convolutional neural networks: Architectures and applications. *International Journal of Computer Science Trends and Technology*, 9(2), 8–15.

Espinosa, A., Giraldo, J. and Lozano, R. (2017). Faster R-CNN: Advances in region proposal networks. *International Journal of Computer Applications*, 179(25), 10–15.

Girshick, R. (2015). Fast R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1440–1448.

Gu, W., Wang, Y. and Lu, Y. (2018). Pooling methods in convolutional neural networks. *Neural Computing and Applications*, 30(7), 1989–2001.

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

Howard, A.G. *et al.* (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint*, arXiv:1704.04861.

Indolia, N., Kumar, P., Singh, S.P. and Singh, K. (2018). Conceptual understanding of convolutional neural network—A deep learning approach. *Procedia Computer Science*, 132, 679–688.

Jayanthi, J. (2022). Machine learning algorithms and their applications. *International Journal of Artificial Intelligence Research*, 6(2), 33–42.

Jocher, G. (2020). YOLOv5: Implementation in PyTorch. *Ultralytics*. [Online]. Available: https://github.com/ultralytics/yolov5

Jocher, G., Chaurasia, A. and Qiu, J. (2023). YOLOv8: Next-generation real-time object detection and segmentation. *Ultralytics Technical Report*.

Jocher, G., Chaurasia, A. and Stoken, A. (2023). YOLOv8: Cutting-edge object detection. *Ultralytics*. [Online]. Available: https://github.com/ultralytics/ultralytics

Kant, R., Pal, R., Dixit, A.K. and Kaur, G. (2024). Dark net and deep web: Legal issues and regulations. In: *Proceedings of the International Conference on Innovations in Data Analytics*, Springer, 557–582.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012). ImageNet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems (NIPS)*, 1097–1105.

Kuo, H. (2016). Understanding the role of nonlinear activation in CNNs. *IEEE Transactions on Neural Networks and Learning Systems*, 27(7), 1896–1907.

Li, C., Li, L., Geng, Y., Jiang, H., Cheng, M. *et al.* (2023). YOLOv6 v3.0: A full-scale reloading. *arXiv preprint*, arXiv:2301.05586.

Li, C., Li, L., Jiang, H., Weng, K., Geng, Y. *et al.* (2022). YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint*, arXiv:2209.02976.

Li, Y., Zhang, Y. and Wu, Y. (2021). Sum pooling for efficient CNNs. *Journal of Computational Vision*, 17(3), 201–212.

Nguyen, H., Pham, D. and Tran, L. (2018). ResNet-based image detection models. *Procedia Computer Science*, 144, 332–339.

Obaidat, M.J., Al-Syouf, I.A., Awawdeh, Y.F., Masa'deh, A.E. and Al-Haija, Q.A. (2025). Darknet threats and detection strategies: A concise overview. In: *Proceedings of the International Conference on Information and Communication Systems (ICICS)*, 1–6.

O'Shea, K. and Nash, R. (2015). An introduction to convolutional neural networks. *arXiv preprint*, arXiv:1511.08458.

Patro, S. and Boddupally, V. (2016). Object localization and detection techniques in computer vision. *Procedia Computer Science*, 89, 709–716.

Ponnusamy, P., Balamurugan, R. and Ilango, P. (2017). Image classification using machine learning approaches. *International Journal of Computer Vision and Image Processing*, 7(2), 1–13.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.

Redmon, J. and Farhadi, A. (2017). YOLO9000: Better, faster, stronger. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7263–7271.

Redmon, J. and Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv preprint*, arXiv:1804.02767.

Sehgal, R., Kumar, V. and Sharma, S. (2022). Image processing applications in medical imaging. *Journal of Imaging Science*, 66(3), 112–118.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint*, arXiv:1409.1556.

Swapna, P., Ramesh, R. and Thomas, S. (2020). AlexNet and its applications in computer vision. *International Journal of Image, Graphics and Signal Processing*, 12(4), 1–9.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015). Going deeper with convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1–9.

Terven, A. (2023). YOLO for object detection: Concepts and applications. *arXiv preprint*, arXiv:2301.07520.

Tian, Y. *et al.* (2025). YOLOv12: Attention-centric real-time object detectors. *arXiv preprint*.

Uijlings, J., van de Sande, K., Gevers, T. and Smeulders, A. (2013). Selective search for object recognition. *International Journal of Computer Vision*, 104(2), 154–171.

Ultralytics (2024). YOLOv11: High-performance object detection model. *GitHub Repository*. Available: https://github.com/ultralytics/ultralytics

Wang, C., Bochkovskiy, A. and Liao, H.Y.M. (2020). CSPNet: A new backbone for object detection. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 390–391.

Wang, C.Y., Bochkovskiy, A. and Liao, H.Y.M. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint*, arXiv:2207.02696.

Xu, Z., Wu, Y. and Wu, H. (2023). YOLOv9: A hybrid transformer-based object detection model with improved accuracy and speed. *arXiv preprint*, arXiv:2308.12914.

Xu, Z., Wu, Y. and Wu, H. (2024). YOLOv10: Real-time object detection with efficient design. *arXiv preprint*, arXiv:2405.14458.

Zhao, Z., Zheng, P., Xu, S. and Wu, X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11), 3212–3232.