

Article

Not peer-reviewed version

Deep Convolutional Neural Networks: A Comprehensive Review

[Ibomojye Domor Mienye](#)*, [Theo G. Swart](#), [George Obaido](#), Matt Jordan, Philip Ilono

Posted Date: 19 August 2024

doi: 10.20944/preprints202408.1288.v1

Keywords: CNN; deep learning; facial recognition; image classification; medical imaging; neural networks; remote sensing; segmentation



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Deep Convolutional Neural Networks: A Comprehensive Review

Ibomoiye Domor Mienye ^{1,†*}, Theo G. Swart ^{1,†}, George Obaido ², Matt Jordan ^{3,†}
and Philip Ilono ^{4,†}

¹ Institute of Intelligent Systems, University of Johannesburg, Johannesburg, 2006, Gauteng, South Africa; tgswart@uj.ac.za

² Center for Human-Compatible Artificial Intelligence (CHAI), Berkeley Institute for Data Science (BIDS), University of California, Berkeley, California, 94720, USA; gobaido@berkeley.edu

³ Department of Physics, University of Nottingham, Nottingham, United Kingdom; ezymj4@gmail.com

⁴ Wolfson School of Mechanical, Electrical and Manufacturing Engineering, Loughborough University, Loughborough, LE11 3TU, United Kingdom; philip_ilono@hotmail.co.uk

* Correspondence: bomoiyem@uj.ac.za (I.D.M.)

† These authors contributed equally to this work.

Abstract: Deep convolutional neural networks (CNNs) have revolutionised computer vision technology by automatically learning hierarchical representations of image data, leading to state-of-the-art performance in visual recognition tasks. This article presents a comprehensive review of deep CNNs, from their evolution and architectures to the current state-of-the-art research. The review also provides the core concepts and building blocks of CNNs and their concise mathematical representations. Furthermore, it explores applications of deep CNNs in three popular domains, including medical diagnosis, remote sensing, and facial recognition. This review will benefit researchers and practitioners in computer vision and artificial intelligence, as well as industry professionals seeking to leverage the latest advancements in deep learning technologies.

Keywords: CNN; deep learning; facial recognition; image classification; medical imaging; neural networks; remote sensing; segmentation

1. Introduction

Deep convolutional neural networks have significantly transformed the field of computer vision and are the state-of-the-art algorithms in image classification and segmentation tasks [1–3]. They are at the core of modern computer vision technologies and have achieved exceptional performances in interpreting visual information [4–6]. These networks typically consist of multiple layers, including convolutional layers that detect spatial hierarchies in images by applying filters, pooling layers that reduce dimensionality and highlight important features, and fully connected layers that integrate these features into final predictions [7]. This advanced architecture allows them to solve complex image recognition tasks that are challenging for traditional machine learning (ML) approaches [8].

Furthermore, deep CNNs are robust at extracting and interpreting intricate patterns within vast image datasets, leading to state-of-the-art performance in a wide range of visual recognition tasks [9–11]. Meanwhile, the impact of deep CNNs extends beyond the traditional boundaries of computer vision, influencing numerous fields and interdisciplinary studies. From enhancing natural language processing capabilities by integrating visual data to revolutionizing how autonomous systems perceive their environments, CNNs are crucial in bridging technological gaps between varied domains. For instance, their application in medical imaging has augmented diagnostic procedures and introduced substantial improvements in predictive healthcare, enabling early detection of diseases and personalized treatment strategies [12,13].

Similarly, in environmental monitoring, CNNs have ensured the analysis of complex datasets, improving models for climate prediction and disaster response [14]. This cross-disciplinary applicability demonstrates the versatility and the potential of CNNs to advance different domains.

Furthermore, in recent years, researchers have proposed new architectures and explored their applications in new domains, achieving varying performances, which has created the need for a comprehensive review of these architectures, innovations, and applications. Therefore, this research bridges that gap by providing a comprehensive review of deep CNN architectures and their evolution, highlighting key milestones and breakthroughs in the field. The study explores the various techniques and strategies that have been proposed to improve the performance and generalisation capabilities of deep CNNs and examines the recent advancements in network architectures, training methodologies, and applications.

This proposed research is important as it aims to provide a comprehensive overview of the current state of deep CNNs, providing up-to-date literature for beginners and experts alike. Meanwhile, by understanding these architectures and the techniques used to improve their performance, researchers and practitioners in the field of computer vision can gain valuable insights into how to further advance the capabilities of deep CNNs. Also, reviewing deep CNN applications in the literature and their performances can provide a granular understanding of how specific deep CNN configurations can be optimally applied to different tasks, thus paving the way for more tailored and efficient CNN-based solutions.

The rest of this paper is structured as follows: Section II discusses some relevant related works. Section III presents the background and overview of machine learning and deep learning. Section IV discusses neural networks and their components, which are the main building blocks of CNNs. Section V presents an overview of CNN. Section VI presents the evolution of Deep CNNs and their architectures, Section VII discusses notable applications of deep CNNs, and Section VIII reviews their applications in recent literature. Section IX presents challenges and future research directions, while Section X concludes the study.

2. Related Works

CNNs have achieved remarkable success in tasks such as image classification, object detection, semantic segmentation, and many others [15–18]. Several researchers have reviewed CNNs, together with their design principles, architectures, capabilities, and performance across various applications. A recent study by Dhillon and Verma [17] presented a detailed review of various CNN architectures, starting from classical models like LeNet to more recent ones such as AlexNet and ENAS. Alzubaidi et al. [7] extended the review by investigating deep learning and CNNs, providing a comprehensive analysis of deep learning, which is now the gold standard in the ML community in recent years. The review discusses the evolution of CNN architectures and highlights several applications of DL across diverse domains, including cybersecurity, natural language processing, bioinformatics, robotics, and medical information processing. Similarly, Cong and Zhou [19] reviewed CNN architectures by exploring the research advancements and optimization techniques associated with typical CNN architectures. The study then proposed a novel approach for classifying CNN architectures based on modules, aiming to accommodate the growing diversity of network architectures with unique characteristics.

Meanwhile, the application of CNNs spans a diverse range of fields, each with its own set of challenges and opportunities. In the agriculture domain, Kamilaris and Prenafeta-Boldú [20] presented an overview of several studies that have applied CNNs to various agricultural and food production challenges. The study further examined the methodologies employed in the reviewed research, including data preprocessing techniques, model training procedures, and validation strategies. Similarly, Lu et al. [21] further reviewed CNN applications in plant disease classification. The study outlined the DL principles involved in plant disease classification and summarized the main challenges encountered when using CNNs for this purpose, along with corresponding solutions.

CNNs have also been applied in aerial image processing. For example, Kattenborn et al. [22] studied the use of CNN in vegetation remote sensing, highlighting its potential to transform the field. Liu et al. [23] reviewed CNN architectures and their applications in aerial view image processing using

drone technology. The research studied the principles and classic architectures of CNNs. Furthermore, the study discussed the latest research trends in CNNs, including advancements in model architectures and techniques for improving accuracy in aerial image analysis. Furthermore, CNNs have been employed in smart homes for various applications such as activity recognition, anomaly detection, and home automation. A review by Yu et al. [24] summarized the use of CNNs in smart home environments and highlighted their role in improving the efficiency and convenience of residential living.

Object detection has been a fundamental task in computer vision, and CNNs have significantly advanced this area. Reviews, such as the study by Zhiqiang and Jun [25] and Sultana et al. [1] provided an overview of the evolution of CNN-based object detection techniques and their applications in diverse fields ranging from autonomous driving to surveillance systems. In a similar research, Ji et al. [26] presented a review of CNN in object detection, focusing specifically on the application of CNN-based encoder-decoder models in salient object detection (SOD). The study concludes with a comprehensive summary that highlights the findings and provides suggestions for future research in the field of CNN-based encoder-decoder models for SOD.

Density map generation has also been explored using CNNs, particularly in crowd analysis and surveillance. Reviews such as the work by Sindagi and Patel [27] provide an overview of CNN-based approaches for generating density maps from images and their applications in crowd management and urban planning. Machine fault diagnosis is another area where CNNs have shown promising results. Reviews by Jiao et al. [28], Wen et al. [29] and Tang et al. [30] provided comprehensive insights into the application of CNNs for fault diagnosis in machinery and industrial systems. They highlighted the importance of CNNs in enhancing maintenance efficiency and minimizing downtime in industrial operations. These reviews extensively discussed the effectiveness of CNN-based approaches in detecting and diagnosing faults, thereby facilitating proactive maintenance strategies and optimizing equipment performance.

CNNs have been instrumental in medical image analysis, disease diagnosis, and drug discovery. Reviews, such as Anwar et al. [31], Sarvamangala et al. [32] and Lu et al. [33] have extensively surveyed CNN-based approaches in medical imaging. These reviews demonstrate the potential of CNNs in improving healthcare diagnostics and treatment by enabling accurate and efficient analysis of medical images.

The surveyed studies have shown significant progress in employing CNN-based approaches across diverse applications, indicating the robustness of these models in solving complex visual recognition tasks. However, despite these advances, several studies reveal recurring limitations, such as the intensive computational demand of training deep CNNs, lack of transparency in model decisions, and overfitting. These challenges constrain the practical deployment of Deep CNNs and highlight the need for continued innovation in network architecture and optimization techniques. Meanwhile, it is evident that some CNN architectures are particularly well-suited for specific applications but may not perform optimally in others. Hence, exploring recent CNN applications in the literature and their performances can therefore guide future research to better match architectural choices with application needs. Therefore, this review addresses these gaps by providing a comprehensive evaluation of recent developments in CNN designs and their applications. It provides new perspectives and methodologies that aim to enhance model efficiency, contributing to a deeper understanding of CNN's capabilities and opening avenues for future advancements in the field.

3. Background

Artificial Intelligence (AI), machine learning, and deep learning (DL) are interconnected fields that form the backbone of modern computational technologies, which have led to innovation and efficiency in several industries. AI is a broad discipline in computer science focused on creating systems capable of performing tasks that typically require human intelligence [34]. Meanwhile, ML, a subset of AI, is the study of computer algorithms that improve automatically through experience and by the use of data. Unlike traditional programming, where humans explicitly code the behaviour, ML enables

computers to learn and make predictions or decisions based on data [35]. DL, a further subset of ML, involves neural networks with many layers. These deep neural networks have significantly advanced the field of AI, achieving remarkable results in complex tasks. The relationship between these three concepts can be described as follows:

$$AI \supseteq ML \supseteq DL \tag{1}$$

This hierarchy illustrates that while all ML is AI, not all AI is ML; similarly, while all DL is ML, not all ML is DL. Furthermore, this relationship can be visualized in Figure 1, showing a set of concentric circles, with AI being the broadest category that encompasses ML, which in turn encompasses DL.

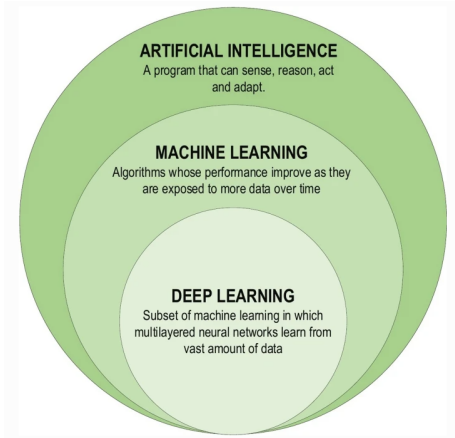


Figure 1. Relationship between AI, ML, and DL [36]

4. Neural Network and Its Components

Artificial neural networks (ANNs) are a class of machine learning algorithms inspired by the structure and function of the human brain. They are composed of nodes, or neurons, connected by edges, which can transmit signals processed by the neurons [37]. Neural networks have been widely applied in various fields, including pattern recognition, speech recognition, and computer vision, due to their ability to learn and model complex non-linear relationships [38]. A typical neural network consists of an input layer, one or more hidden layers, and an output layer [39]. Each layer is made up of a number of interconnected neurons. The input layer receives the raw data, the hidden layers perform computations and transformations, and the output layer produces the final prediction or classification. The basic structure of a neural network is shown in Figure 2.

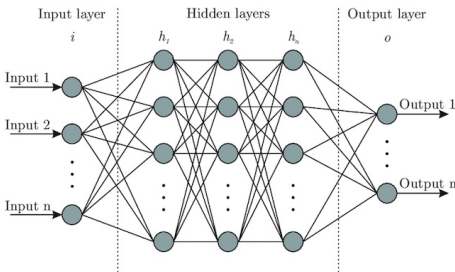


Figure 2. Basic structure of a Neural Network

The learning process in a neural network involves adjusting the weights and biases to minimize the difference between the actual output and the predicted output, typically using a method such as gradient descent. The most common loss function used in this optimization process is the mean squared error for regression problems or cross-entropy loss for classification problems. The various neural network components are discussed as follows:

4.1. Neurons

A neuron is the basic processing unit of a neural network. The operation of a neuron in a neural network can be mathematically represented as:

$$y = f \left(\sum_{i=1}^n w_i x_i + b \right) \quad (2)$$

where x_i are the input values, w_i are the weights associated with each input, b is the bias, and f is the activation function that introduces non-linearity into the model [40]. The neuron computes a weighted sum of its inputs, adds a bias, and then applies an activation function to the result.

4.2. Weights and Biases

Weights (w) and biases (b) are the parameters the neural network learns during training. They are adjusted to minimize the loss function, which measures the difference between the network's predictions and the actual target values [41,41].

4.3. Layers

Neural networks are structured into layers, each serving a unique function in the data processing pipeline. These layers are categorized into three main types: the input layer, hidden layers, and the output layer. The input layer is the gateway for data to enter the neural network. It is responsible for receiving raw input data, with one neuron for each input feature [42]. The input layer primarily serves to pass on the raw input data to subsequent layers without applying any transformations. It can be represented mathematically as:

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^T \quad (3)$$

where \mathbf{x} represents the input vector consisting of n features that are fed into the network. Hidden layers are the core of a neural network. They lie between the input and output layers and are responsible for performing various computations and transformations on the input data [43]. The architecture, number, and size of these hidden layers can vary greatly depending on the complexity of the task and the design of the network. Each hidden layer consists of a number of neurons, where each neuron is connected to all the neurons in the previous layer [39]. These layers apply weights, biases, and activation functions to the inputs they receive, which allows the network to learn complex patterns and relationships in the data. It is represented mathematically as:

$$\mathbf{h}_i = f(\mathbf{W}_i \mathbf{h}_{i-1} + \mathbf{b}_i) \quad (4)$$

where \mathbf{h}_i is the output of the i -th hidden layer, f is the activation function, \mathbf{W}_i and \mathbf{b}_i are the weight matrix and bias vector for the i -th layer, respectively, and \mathbf{h}_{i-1} represents the output of the previous layer. The output layer is the final layer in a neural network. It takes the high-level features extracted by the hidden layers and transforms them into a format suitable for making predictions. The design of the output layer, including the choice of activation function, depends on the specific task the network is intended to perform [39,39]. For instance, a softmax activation function is often used for multi-class classification tasks, as it can output a probability distribution over the classes. The output layer essentially summarizes the information that the network has learned about the data fed through the input and hidden layers. The output vector (\mathbf{y}) of the network is represented as:

$$\mathbf{y} = f_{\text{output}}(\mathbf{W}_{\text{output}} \mathbf{h}_L + \mathbf{b}_{\text{output}}) \quad (5)$$

where f_{output} is the activation function used in the output layer and \mathbf{h}_L is the output of the last hidden layer (L).

4.4. Activation Functions

Activation functions are another important component of neural networks that play a crucial role in determining the output of each neuron. These functions introduce non-linearity into the network, allowing it to learn complex patterns and relationships in the data [44]. A commonly used activation function is the rectified linear unit (ReLU), $f(x) = \max(0, x)$, which replaces all negative values with zero. ReLU has been shown to improve the convergence of neural networks by preventing the vanishing gradient problem and speeding up training [45]. The sigmoid function, $\sigma(x) = \frac{1}{1+e^{-x}}$, is another popular activation function, which compresses the output of each neuron to a value between 0 and 1, making it useful for binary classification tasks. Tanh, $\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$, is another activation function that is similar to the sigmoid function, but rather compresses the output to a value between -1 and 1, making it useful for tasks where the output needs to be centered around zero.

Researchers have recently introduced new activation functions, such as the Leaky ReLU, which allows a small, positive gradient for negative input values to prevent neurons from becoming completely inactive, a problem often referred to as 'dead neurons' [46]. Other activation functions, such as the exponential linear unit (ELU) [47] and the parametric rectified linear unit (PReLU) [48], have also been proposed to address the limitations of traditional activation functions.

4.5. Loss Function

The loss function quantifies the difference between the predicted output of the network and the true target output. It is a crucial component of training a neural network. Loss functions are vital in the optimization process of neural networks, as they provide a measure of how well the model is performing during training [49]. By minimizing the loss function, the neural network is able to adjust its weights and biases to improve its predictions and ultimately improve the model's accuracy. A widely used loss function is the mean squared error (MSE), $L(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$, which calculates the average of the squared differences between the predicted and actual outputs. Another popular loss function is the cross-entropy loss, $L(y, \hat{y}) = -\sum_i y_i \log(\hat{y}_i)$; it is commonly used in classification tasks to measure the difference between the model's predicted probability distribution and the true distribution of class labels.

In addition to these common loss functions, there are also specialized loss functions designed for specific tasks, such as the Huber loss function [50] for robust regression and the focal loss function [51] for imbalanced classification problems. An important consideration when choosing a loss function is its impact on the optimization process. Some loss functions may lead to faster convergence and better generalization, while others may be more sensitive to noisy data [52]. Therefore, understanding the characteristics of different loss functions and how they interact with the neural network architecture is crucial for achieving optimal performance.

4.6. Optimizer

Optimizers in neural networks are algorithms used to adjust the parameters of the network, such as weights and biases, to minimize (or maximize) an objective function [53]. These adjustments are crucial for the learning process, ensuring the network gradually improves its accuracy by iteratively reducing errors in predictions. The choice of optimizer can significantly influence the performance of the model and its convergence rate. The widely used optimizers are discussed as follows:

4.6.1. Gradient Descent

Gradient descent is a fundamental optimizer that updates the model parameters, typically the weights, by moving them in the direction of the negative gradient of the objective function with respect to the parameters [54]. The update equation is:

$$\theta_{t+1} = \theta_t - \eta \nabla J(\theta_t), \quad (6)$$

where θ represents the parameters, η is the learning rate, and $\nabla J(\theta_t)$ is the gradient of the loss function J with respect to the parameters at iteration t .

4.6.2. Stochastic Gradient Descent

Stochastic gradient descent (SGD) modifies the standard gradient descent algorithm by updating the parameters using only a single sample or a subset of the training data [55]. The idea is to approximate the gradient of the entire training set by using the gradient of a single instance, which reduces the computational burden significantly:

$$\theta_{t+1} = \theta_t - \eta \nabla J(\theta_t; x_i, y_i), \quad (7)$$

where (x_i, y_i) is a randomly selected sample from the dataset.

4.6.3. Batch Gradient Descent

Batch gradient descent is a form of gradient updating where the model parameters are updated only once per epoch, using the gradient computed from the entire dataset [56]. This approach is computationally expensive and impractical for very large datasets but can lead to stable convergence:

$$\theta_{t+1} = \theta_t - \eta \nabla J(\theta_t; X, Y), \quad (8)$$

where (X, Y) represents the entire dataset.

4.6.4. Mini-Batch Gradient Descent

Mini-batch gradient descent is a balance between the stochastic and batch versions [57]. It updates the parameters using a small subset of the training dataset called a mini-batch:

$$\theta_{t+1} = \theta_t - \eta \nabla J(\theta_t; X_{\text{mini}}, Y_{\text{mini}}), \quad (9)$$

where $(X_{\text{mini}}, Y_{\text{mini}})$ is a mini-batch of the dataset. This method balances the advantage of faster computation with the benefit of stable convergence.

4.6.5. Adaptive Moment Estimation

Adaptive moment estimation (Adam) is one of the most popular modern optimizers. It combines the advantages of two other stochastic gradient descent extensions: adaptive gradient algorithm (AdaGrad) and root mean square propagation (RMSProp). It computes individual adaptive learning rates for different parameters:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t, \quad (10)$$

where \hat{m}_t and \hat{v}_t are estimates of the first and second moments of the gradients, respectively, and ϵ is a small scalar used to prevent division by zero [58].

4.7. Backpropagation

Backpropagation is the fundamental algorithm used for training neural networks. It involves the systematic application of the chain rule to efficiently compute gradients of a loss function with respect to all weights in the network:

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial w} \quad (11)$$

$\frac{\partial L}{\partial w}$ represents the gradient of the loss function L with respect to a weight w , $\frac{\partial L}{\partial y}$ is the gradient of the loss with respect to the output y , and $\frac{\partial y}{\partial w}$ is the gradient of the output with respect to the weight.

This gradient tells us how much a small change in the weight w will affect the loss. Backpropagation proceeds in a reverse manner across the network's layers, starting from the output back to the first hidden layer.

5. Overview of CNN and Its Building Blocks

Convolutional Neural Networks are a class of deep neural networks, highly effective for processing data with a grid-like topology, such as images. CNNs are particularly known for their ability to capture spatial and temporal dependencies in an image through the application of relevant filters [59]. The architecture of a CNN allows it to automatically and adaptively learn spatial hierarchies of features, from low-level to high-level patterns. Figure 3 shows the schematic of a CNN, which is composed of several layers that transform the input image to produce an output (e.g., a class label). Also, CNNs include several key components that contribute to their ability to effectively process and learn from image data. Two such components are Batch Normalization and Dropout, each playing a crucial role in enhancing the training stability and generalization of CNN models. The key components of CNN, shown in Figure 3, are presented and discussed as follows:

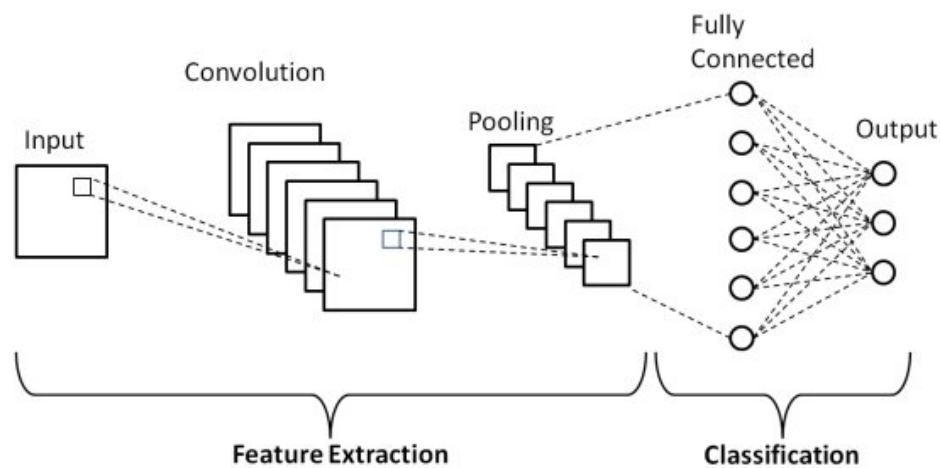


Figure 3. Basic CNN Architecture [60]

5.1. Convolutional Layer

The convolutional layer is the core building block of a CNN that performs most of the computational tasks. It applies a set of learnable filters to the input image. Each filter activates certain features from the input. These features are then passed through an activation function to introduce non-linearity into the model, allowing it to learn more complex patterns[61]. ReLU is the most commonly used activation function in CNNs [62]. The output of the convolutional layer is a set of feature maps that represent different aspects of the input image. Mathematically, the convolution operation is defined as:

$$F(i, j) = (G * H)(i, j) = \sum_m \sum_n G(m, n) H(i - m, j - n) \quad (12)$$

where F is the output feature map, G is the input image, H is the filter, and $*$ denotes the convolution operation.

5.2. Pooling Layer

The pooling layer aims to reduce the spatial dimensions (width and height) of the input volume for the subsequent convolutional layer [63]. It operates independently on each depth slice of the input, downsizing it spatially to reduce the amount of computation needed for the upper layers and to help

make the feature representations more compact and robust to variations in the position of features in the input. The most commonly used form of pooling is max pooling, which selects the maximum value from each patch of the feature map covered by the kernel [64]. Mathematically, for a given input volume, the operation of a pooling layer can be defined as follows for max pooling:

$$P_{ij} = \max_{a,b \in \mathcal{N}_{ij}} X_{ab} \quad (13)$$

where P_{ij} is the output of the pooling operation at position (i, j) , X_{ab} represents the elements of the input volume over which the pooling operation is applied, and \mathcal{N}_{ij} is the neighborhood defined by the pooling window size and stride over the input volume at position (i, j) .

5.3. Fully Connected Layer

In a fully connected (FC) layer, each neuron is connected to every neuron in the previous layer, with each connection possessing its own weight. The output of this layer is computed as a weighted sum of the inputs, to which a bias term is added and then passed through an activation function [65]. This setup allows the layer to learn complex patterns from the entire data set represented by the activations in the previous layer, making FC layers critical for integrating learned features into final predictions or classifications. The mathematical representation is:

$$\mathbf{y} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b}) \quad (14)$$

where \mathbf{x} is the input vector to the fully connected layer, \mathbf{W} represents the weight matrix, \mathbf{b} is the bias vector, σ denotes the activation function, and \mathbf{y} is the output vector of the fully connected layer.

5.4. Batch Normalization

Batch normalization is a technique designed to improve the speed, performance, and stability of artificial neural networks. It is applied as a layer within the network that normalizes the inputs to a layer for each mini-batch [66]. This normalization process involves adjusting and scaling the activations of the previous layer such that they have a mean output activation of zero and a standard deviation of one. Mathematically, it can be represented as:

$$\hat{x}^{(k)} = \frac{x^{(k)} - \mathbb{E}[x^{(k)}]}{\sqrt{\text{Var}[x^{(k)}] + \epsilon}} \quad (15)$$

where $x^{(k)}$ is the input to be normalized, $\mathbb{E}[x^{(k)}]$ is the mean of the input, $\text{Var}[x^{(k)}]$ is the variance of the input, and ϵ is a small constant added for numerical stability. The normalized data $\hat{x}^{(k)}$ is then scaled and shifted by learned parameters γ and β , which allows the model to undo the normalization if that is what the data requires. Batch Normalization helps in reducing the internal covariate shift which occurs when the distribution of each layer's inputs changes during training, as the parameters of the previous layers change [66]. Stabilizing the distribution of the inputs to a layer allows for higher learning rates and reduces the sensitivity to the initial weights.

5.5. Dropout

Dropout is a regularization technique designed to prevent overfitting in neural networks. It functions by randomly deactivating a specified fraction of the neurons (by setting their outputs to zero) at each update during the training phase [67].

$$r_j^{(l)} \sim \text{Bernoulli}(p) \quad (16)$$

where $r_j^{(l)}$ is a masking neuron that is 1 if neuron j in layer l is kept and 0 if it is dropped out, and p is the probability of keeping a neuron active during training. Dropout can be considered a form of

ensemble learning, where each mini-batch trains a different model with a different architecture, but all these models share weights [68]. This randomness helps to increase the robustness of the model, making it less likely to rely on any one feature and, therefore, improving its generalization to unseen data. Dropout is applied by randomly setting a proportion of the activations in a layer to zero during the forward pass. Dropout is not applied at test time; instead, the layer's output activations are scaled by the dropout rate to account for the reduced number of active neurons during training.

6. Evolution of deep CNNs and Architectures

The evolution of deep CNNs can be traced back to the early work on neural networks in the 1980s and 1990s. However, it was not until the early 2010s that deep CNNs really began to show their potential [69]. One of the field's key breakthroughs was the AlexNet by Krizhevsky et al. [70] architecture's development in 2012, which significantly outperformed previous methods in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Since then, researchers have continued to push the boundaries of deep CNNs by exploring new architectures, optimizing training algorithms, and leveraging larger datasets. Another key advancement in recent years has been the development of residual networks (ResNets) in 2015 by He et al. [71], which introduced skip connections to allow for easier training of very deep networks. The different CNN architectures are discussed as follows:

6.1. LeNet

LeNet [72], a groundbreaking CNN architecture developed by Yann LeCun in the 1990s, has played a pivotal role in the advancement of image recognition technology. Originally designed for handwritten digit recognition, LeNet consists of two convolutional layers and three fully connected layers, incorporating average pooling and hyperbolic tangent activation functions. Despite its relative simplicity compared to more modern CNN architectures, LeNet has proven highly effective in various image recognition tasks. The LeNet Architecture is described as follows:

$$\text{Input} \rightarrow \text{C5} \rightarrow \text{S2} \rightarrow \text{C5} \rightarrow \text{S2} \rightarrow \text{F120} \rightarrow \text{F84} \rightarrow \text{Output} \quad (17)$$

where C5 represents a convolutional layer with a 5x5 kernel, S2 represents a subsampling (average pooling) layer with a 2x2 kernel, and F120 represents a fully connected layer with 120 units.

The success of LeNet can be attributed to its innovative design, which was ahead of its time. The use of convolutional layers allows the network to automatically learn features from input images, reducing the need for manual feature extraction [73]. This improves the accuracy of the network and makes it more adaptable to different types of images. The incorporation of average pooling helps to downsample the feature maps, reducing the computational complexity of the network while preserving important information. Additionally, the use of hyperbolic tangent activation functions allows for non-linear transformations of the input data, enabling the network to learn complex patterns and relationships within the images.

One of the key strengths of LeNet is its ability to generalize well to new, unseen data. This is due to the hierarchical structure of the network, which allows it to learn increasingly complex features at each layer [72]. By combining these features in the fully connected layers, LeNet is able to make accurate predictions on a wide range of image recognition tasks. Despite its success, LeNet does have some limitations. For example, using average pooling can lead to losing spatial information in the feature maps, which may affect the network's ability to accurately localize objects within an image. Additionally, the relatively shallow architecture of LeNet may limit its performance on more complex tasks that require the learning of intricate features.

6.2. AlexNet

AlexNet [70] is a groundbreaking CNN architecture that revolutionized the field by significantly outperforming traditional computer vision methods in the ILSVRC challenge. AlexNet comprises

five convolutional layers followed by three fully connected layers, incorporating ReLU activation functions and max pooling layers throughout. The AlexNet architecture is shown in Figure 4, and it is considered the first deep CNN architecture as it improves the CNN's learning ability by increasing the depth of the network and utilising several parameter optimisation techniques [74].

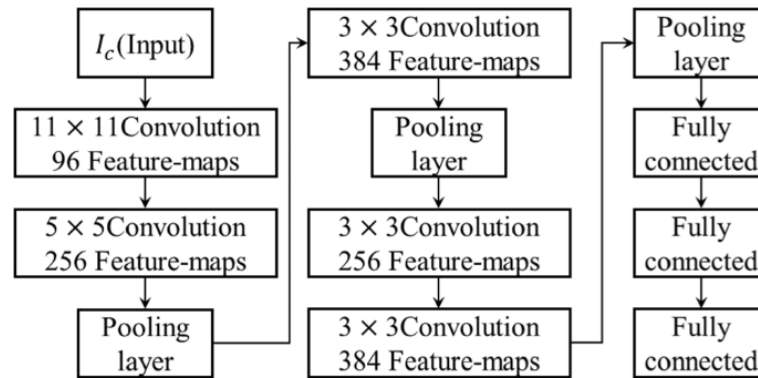


Figure 4. AlexNet Architecture [74]

This architecture introduced the concept of dropout regularization, a technique used to prevent overfitting in deep learning models. Dropout has since become a widely adopted practice in the field of machine learning. The success of AlexNet marked a turning point in the field of computer vision, demonstrating the power of deep learning in image classification tasks. Prior to AlexNet, computer vision systems relied heavily on handcrafted features and shallow learning algorithms. However, the deep architecture of AlexNet allowed for the automatic learning of hierarchical features directly from raw pixel data, leading to a significant improvement in performance. The impact of AlexNet extended beyond just the field of computer vision. Its success sparked a resurgence of interest in deep learning and led to a wave of research and development in the field. Researchers began to explore the potential of deep neural networks in a wide range of applications, from natural language processing to autonomous driving. The success of AlexNet also paved the way for the development of even more powerful CNN architectures.

6.3. ResNet

The ResNet [71] architecture was developed by Microsoft Research in 2015. The hallmark of ResNet lies in its remarkable depth, with some variants boasting up to 152 layers [75]. This depth is made possible by the ingenious use of residual blocks, which feature shortcut connections that bypass one or more layers, as shown in Figure 5. These shortcuts allow for easier training of deep networks by mitigating the vanishing gradient problem, which is common in deep networks. The ResNet is described as follows:

$$\text{Input} \rightarrow \left[\begin{array}{c} \text{Conv Block} \\ \text{Identity Block} \times n \end{array} \right] \times L \rightarrow \text{Avg Pool} \rightarrow F \rightarrow \text{Output} \quad (18)$$

where Conv Block represents a convolutional block that fits the input size to the residual block, Identity Block represents the residual blocks repeated n times, and L represents the total number of layers. Figure 5 shows the ResNet architecture.

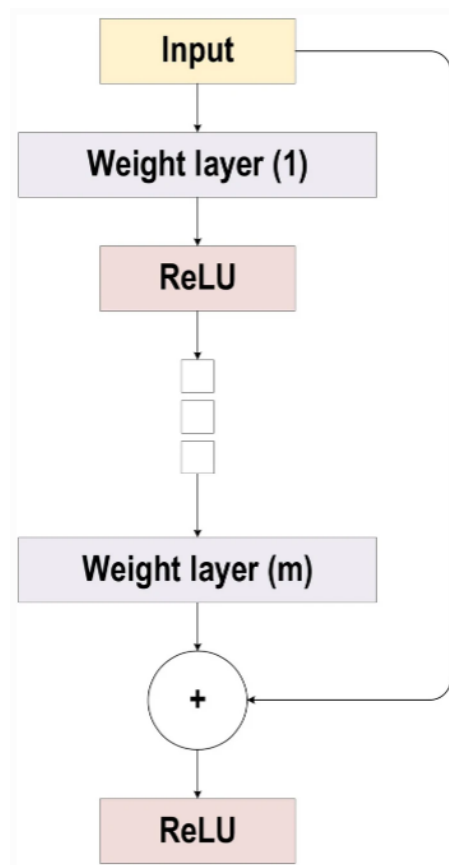


Figure 5. ResNet Architecture

The success of ResNet can be attributed to its ability to effectively learn complex features from raw data, leading to state-of-the-art performance on a wide range of computer vision tasks, such as image classification and object detection. By allowing information to flow through the network more easily, ResNet is able to capture intricate patterns in data that may be missed by shallower networks [76]. An advantage of ResNet is its ability to train very deep networks without suffering from degradation in performance. In traditional deep networks, as the number of layers increases, the accuracy of the network tends to saturate and then rapidly degrade. This phenomenon, known as the degradation problem, is a major obstacle in training deep neural networks. However, ResNet's use of shortcut connections allows for the training of extremely deep networks with minimal degradation in performance [77].

Furthermore, ResNet has been shown to be highly efficient in terms of both computational resources and memory usage [78,79]. This efficiency is crucial for real-world applications where speed and resource constraints are important factors to consider. In addition to its impressive performance on standard computer vision tasks, ResNet has been successfully applied to other domains, such as natural language processing and speech recognition. Its versatility and ability to learn complex patterns make it a valuable tool for a wide range of applications in artificial intelligence.

6.4. VGGNet

VGGNet [80] was developed by the Visual Geometry Group at the University of Oxford in 2014. It is renowned for its simplicity, consisting of either 16 or 19 layers of convolutional and pooling layers. This straightforward architecture has proven to be highly effective in extracting features from images, leading to state-of-the-art performance on various image classification tasks. The success of VGGNet can be attributed to its deep architecture, which allows for the learning of complex hierarchical features. Each convolutional layer in the network acts as a filter, extracting specific features from the input

image. These features are then passed through pooling layers, which downsample the feature maps to reduce computational complexity and increase translation invariance [81]. This process is repeated multiple times throughout the network, with each subsequent layer learning increasingly abstract features. The VGGNet architecture is shown in Figure 6.

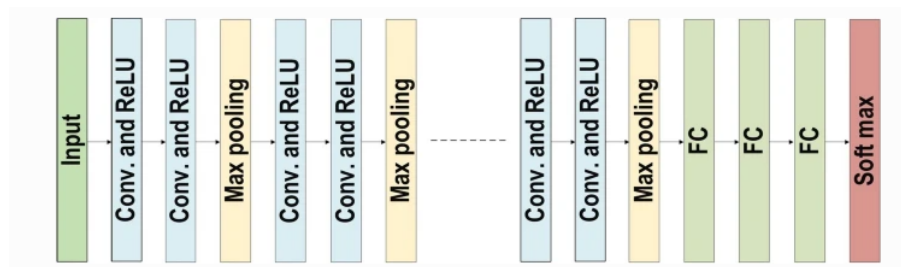


Figure 6. VGGNet Architecture

An important aspect of VGGNet's architecture is the use of small 3x3 convolutional filters. These small filters have been shown to be highly effective in capturing spatial information within the image, allowing the network to learn intricate patterns and textures. Additionally, the use of multiple convolutional layers with small filters helps to increase the depth of the network without significantly increasing the number of parameters, making training more efficient. Additionally, its use of max pooling layers helps reduce the feature maps' spatial dimensions while retaining important information. This downsampling process aids in preserving spatial relationships between features, allowing the network to better generalize to new, unseen images [82]. Also, the use of max pooling helps to introduce translation invariance, ensuring that the network can still accurately classify images even if they are slightly shifted or rotated.

6.5. GoogleNet

GoogleNet [83], or Inception v1, marks a significant advancement in CNN architecture for visual recognition tasks. Introduced in the seminal paper "Going Deeper with Convolutions," GoogleNet was designed to optimize the depth and width of the network while maintaining computational efficiency. The hallmark of GoogleNet is the inception module (shown in Figure 7, a novel architectural unit that allows for increased depth and width without the proportional increase in computational cost. The inception module is based on the idea that instead of deciding which size of convolutional filter to use (e.g., 1x1, 3x3, 5x5), a CNN should utilize all of them in parallel to capture information at various scales. The outputs of these filters are then concatenated along the channel dimension. This approach allows the network to adapt to the most relevant scale of features for each task, improving its ability to capture complex patterns:

$$\text{Inception}(x) = [\text{Conv}_{1 \times 1}(x), \text{Conv}_{3 \times 3}(x), \text{Conv}_{5 \times 5}(x), \text{Pool}(x)] \quad (19)$$

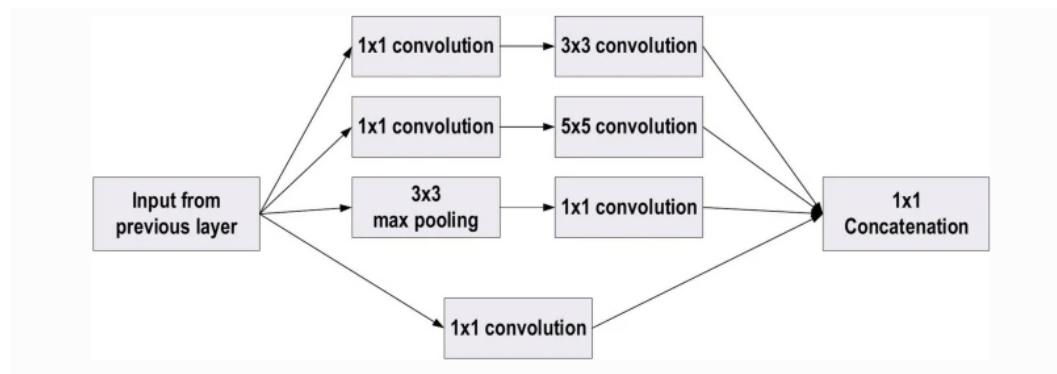


Figure 7. GoogleNet Architecture

GoogleNet significantly increased the depth of the network to 22 layers, a notable increase from previous architectures. Despite its depth, the inception module enables GoogleNet to achieve high computational efficiency. This efficiency is partly achieved by using 1x1 convolutions to reduce the dimensionality of the input feature map before applying larger convolutions, thereby reducing the computational cost [83]. To combat the vanishing gradient problem and encourage gradient flow through the deep network, GoogleNet introduces auxiliary classifiers. These classifiers are added to intermediate layers of the network and contribute to the total loss during training. Although they are not used during inference, these auxiliary structures improve training convergence and, ultimately, the performance of the network. GoogleNet employs global average pooling at the end of the last inception module, replacing the fully connected layers traditionally used in CNNs. This approach significantly reduces the total number of parameters in the network, mitigating overfitting and decreasing the computational burden. GoogleNet's inception architecture has had a significant impact on the field of deep learning, inspiring a series of Inception models that further refine and improve upon the original design.

6.6. DenseNet

Densely Connected Convolutional Networks (DenseNet) [84] is a CNN architecture that was proposed by researchers at Cornell University in 2017. DenseNet introduces a novel architecture that implements the concept of feature reuse, leading to significant improvements in efficiency, compactness, and performance. The defining characteristic of DenseNet is its dense connectivity pattern, shown in Figure 8, where each layer receives input from all preceding layers and passes its own feature maps to all subsequent layers. This connectivity pattern facilitates feature reuse, improves the flow of information and gradients throughout the network, and reduces the number of parameters needed. The fundamental building block of DenseNet is the dense block, where each layer is directly connected to every other layer. Mathematically, the output of the l^{th} layer, x_l , is computed as:

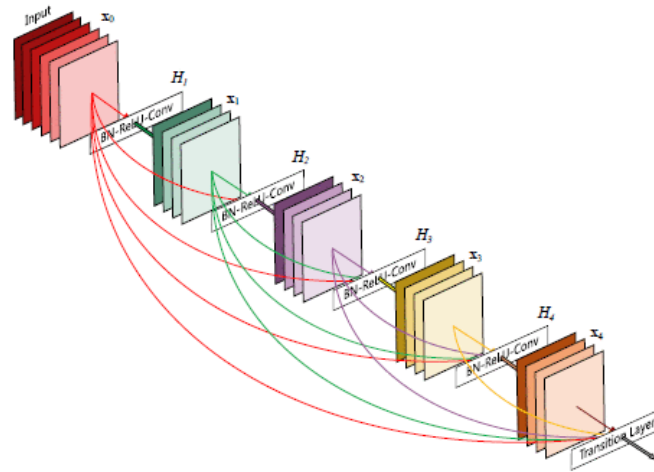


Figure 8. DenseNet [84]

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]) \quad (20)$$

where $[x_0, x_1, \dots, x_{l-1}]$ denotes the concatenation of the feature-maps produced in layers $0, \dots, l-1$, and $H_l(\cdot)$ represents a composite function of operations (BN, ReLU, Conv) [84]. This architecture ensures that each layer can access the gradients from the loss function and the original input signal. A crucial parameter in DenseNet is the growth rate (k), which controls the amount of new information each layer contributes to the global state of the network. The growth rate regulates how much new information is added by each layer, balancing between model capacity and complexity [85,86]. A smaller growth rate keeps the model compact, while a larger growth rate increases the model's capacity to learn complex features. The dense connectivity pattern significantly reduces the need for learning redundant feature maps, thus allowing for a reduction in the number of parameters without sacrificing the depth or capacity of the network. This efficiency makes DenseNet particularly well-suited for applications where model size and computational resources are limited.

To control the size of the feature maps and manage computational complexity, DenseNet introduces transition layers between dense blocks. These layers consist of a batch normalization layer, a 1×1 convolutional layer, and a 2×2 average pooling layer. Transition layers help in compressing the feature-maps and reducing their dimensionality, thus ensuring efficient computation and preventing the network from growing too wide. DenseNet has demonstrated remarkable performance on a wide range of tasks, including image classification [87], object detection, and segmentation. Its ability to efficiently parameterize and learn feature representations makes it a powerful tool for deep learning research and applications. The DenseNet architecture has influenced subsequent developments in CNN design, indicating the benefits of feature reuse and efficient information flow.

6.7. EfficientNet

EfficientNet was developed in 2019 by Tan and Le [88] to systematically balance network depth, width, and resolution in CNNs, which are key factors in determining the model's size, speed, and

accuracy. Prior to EfficientNet, scaling these dimensions was typically performed in an ad-hoc manner, often focusing on just one aspect, such as width or depth, to improve performance. EfficientNet introduces a more structured and effective method for scaling up CNNs, using a compound coefficient to manage the scaling of network dimensions uniformly. The core idea behind EfficientNet is compound scaling [89,90]. Unlike traditional scaling methods that independently scale depth, width, or resolution, EfficientNet scales these dimensions with a fixed set of scaling coefficients determined by a simple yet effective compound coefficient (ϕ). This coefficient is used to simultaneously scale network width, depth, and resolution in a balanced way based on the following formulas:

$$\text{depth} : d = \alpha^{\phi} \quad (21)$$

$$\text{width} : w = \beta^{\phi} \quad (22)$$

$$\text{resolution} : r = \gamma^{\phi} \quad (23)$$

where α , β , and γ are constants that define how each dimension should be scaled, and ϕ is a user-specified coefficient that controls the overall resources available for the model. This method ensures that each additional unit of resource boosts performance in the most efficient way possible. The EfficientNet models are built upon a baseline model, EfficientNet-B0, designed through a neural architecture search (NAS) that optimizes accuracy and efficiency. EfficientNet-B0 is the foundation, and the rest of the EfficientNet models (B1 to B7) are scaled versions of B0 using the compound scaling method [91]. Each subsequent model in the series, from B1 to B7, offers incrementally higher accuracy, with correspondingly higher computational requirements.

EfficientNet models have achieved state-of-the-art accuracy on ImageNet and other benchmark datasets while being significantly more efficient than previous models [92,93]. For example, EfficientNet-B7 achieves much higher accuracy than previous CNN architectures while requiring substantially fewer floating point operations per second (FLOPS), making it powerful and efficient for a wide range of applications, from mobile devices to high-end hardware [94]. Other CNN architectures have been developed in the last few years, including HRNetV2 [94], mobilenetv2 [95], and FractalNet [96] Table 1 presents a summary of the various CNN architectures.

Table 1. Summary of CNN Architectures

Architecture	Author	Year	Innovation	Depth
LeNet	LeCun et al. [72]	1998	Early CNN, handwritten digit recognition	5
AlexNet	Krizhevsky et al. [70]	2012	Uses Dropout and ReLU	8
VGGNet	Simonyan and Zisserman [80]	2014	Increased depth, small filter size	16, 19
GoogLeNet	Szegedy et al. [83]	2015	Inception modules, concatenation	22
Inception-V3	Szegedy et al. [97]	2015	Improved feature representation	48
Highway	Srivastava et al. [98]	2015	Introduced multipath concept	19, 32
Inception-V4	Szegedy et al. [99]	2016	Divided transform and integration concepts	70
ResNet	He et al. [71]	2016	Residual connections to fight vanishing gradient	152
Inception-ResNet-v2	Szegedy et al. [100]	2016	Combination of Inception and ResNet	164
FractalNet	Larsson et al. [96]	2016	Developed Drop-Path regularization	40, 80
Xception	Chollet [101]	2017	Depthwise and pointwise convolution	71
Residual attention	Wang et al. [102]	2017	Introduced attention method	452
DenseNet	Huang et al. [84]	2017	Blocks of layers connected to each other	201
MobileNet-v2	Sandler et al. [95]	2018	Inverted residual structure	53
EfficientNet	Tan and Le [88]	2019	Scaling up CNNs efficiently	380
HRNetV2	Wang et al. [94]	2020	High-resolution representations	–

7. Notable Applications of Deep CNN

In this section, we explore some notable applications of Deep CNNs across different fields, illustrating their versatility and crucial role in advancing modern technology.

7.1. Computer Vision

In computer vision applications, CNNs have established themselves as the main approach for a wide range of image analysis tasks. These include image classification, where CNNs categorize entire images into labels based on the visual content; object detection, which involves identifying and localizing objects within an image complete with bounding boxes; and the more complex image segmentation, which partitions an image into multiple segments to simplify and change its representation.

Image Classification: CNNs are particularly robust at image classification tasks, consistently achieving high accuracy. This robustness enables diverse applications ranging from organizing large photo libraries and enhancing multimedia platforms to providing critical diagnostics in medical imaging, thereby aiding in the early detection and treatment of diseases [103,104].

Segmentation: CNNs are able to analyze and understand the context and the detailed pixel-level composition of images [105]. This capability is crucial in autonomous vehicle systems, which rely on precise interpretations of road conditions, obstacles, and pedestrian areas to navigate safely and efficiently.

7.2. Speech Recognition

In speech recognition, CNNs transform audio waveforms or spectrograms into rich, representative features that capture phonetic and temporal variations, making them ideal for recognizing spoken words or phrases [106]. This technology indicates the significant advancements in voice-activated

systems, such as personal virtual assistants like Siri, Alexa, and Google Assistant, enhancing their responsiveness and accuracy. It also plays a critical role in developing more robust and efficient transcription services to convert speech from various languages and dialects into text with high performance.

7.3. Natural Language Processing

Though recurrent neural networks (RNNs) and transformers predominantly dominate NLP, CNNs have carved a niche in processing sequential data for NLP tasks [107]. They excel in extracting hierarchical features from text, making them suitable for applications such as sentiment analysis, where they determine the emotional tone behind a series of words, and text classification, categorising text into predefined categories [108–110]. CNNs are also employed in aspects of machine translation, particularly in feature learning from large text corpora, enhancing the quality and contextuality of translated content [111,112].

7.4. Object Detection

CNNs have been widely applied in object detection, where they classify images and precisely locate and outline various objects within those images [113]. This dual capability is crucial for applications like security surveillance systems, where continuously detecting and tracking individuals or objects is vital. Additionally, in the digital media industry, CNNs facilitate advanced image editing tools that automatically modify or enhance image elements [114]. Furthermore, they significantly contribute to the reliability and effectiveness of advanced driver-assistance systems (ADAS) by improving the accuracy of detecting and responding to traffic elements, thus enhancing road safety [115–117].

8. Review Deep CNNs in Recent Literature

This section presents a review of recent studies that employed Deep CNNs to solve real-world problems. Specifically, the review focuses on three popular application domains: medical imaging, remote sensing, and face recognition.

8.1. Medical Imaging

Convolutional neural networks have become the core technology in AI-driven medical imaging, achieving excellent performance in automating diagnosis, enhancing image quality, and improving patient outcomes. Their ability to extract and learn features from medical images has led to significant advancements in detecting, classifying, and segmenting various diseases. This section explores some of the key applications of CNN architectures in medical imaging.

CNNs have been widely applied in the detection and classification of diseases from medical images. CNN architectures such as VGG, ResNet, and Inception have been fine-tuned to accurately identify conditions like pneumonia, breast cancer, and diabetic retinopathy from X-rays, mammograms, and retinal images, respectively. For example, Yadav and Jadhav [118] employed a CNN-based architecture to detect pneumonia using chest X-ray images. The study applied transfer learning on two CNN architectures, including the Inceptionv3 and VGG16. Other techniques used for performance comparison include a capsule network (CapsNet) and a support vector machine (SVM). Furthermore, the study applied data augmentation, which significantly improved the performance of the various algorithms. Additionally, the transfer learning on the CNN models proved crucial compared to the SVM and capsule network. The VGG16 achieved the best performance with an accuracy of 93.8%, while the Inception3, CapsNet, and SVM achieved 86.9%, 82.4%, and 77.6%, respectively.

Khatamino et al. [119] employed CNN for detecting Parkinson's disease (PD). PD is a condition affecting the human brain that can cause stiffness and shaking, which can worsen with time. A non-invasive approach for detecting PD has been around for many years. This study proposed a CNN model to learn the features of suspected PD patient's handwritten texts. The study aimed to

identify the handwriting drawing spirals that are usually drawn by PD patients. The study achieved an accuracy of 88%. Similarly, Alissa et al. [120] proposed an approach to detect PD using CNN. Meanwhile, the CNN consists of two convolutional layers with 32 filters, another two convolution layers with 64 filters, and another two convolution layers with 128 filters. The CNN architecture also consists of three max-pooling layers, dense layers, and a flattened layer. The proposed model reached a classification accuracy of 93.5%. Also, Jahan et al. [121], Kurmi et al. [122], Shaban [123] used Inception-V3, ensemble of VGG16, Inception-V3, ResNet50, and Xception, respectively, for detecting PD, achieving an accuracy of 96.6%, 98.45%, 88%, respectively.

Furthermore, Islam et al. [124] proposed a method for detecting COVID-19 disease from X-ray images by combining CNN and LSTM. The CNN was used for feature extraction, while the LSTM classified the extracted features. The study was conducted using 4575 X-ray images, which had 1525 images labelled as COVID-19 positive. The combination of CNN and LSTM resulted in a robust model, achieving 99.4% accuracy. Pan et al. [125] proposed a heart disease detection framework using CNN and MLP. The CNN module was employed for feature extraction, and the MLP classified extracted features. The approach achieved an accuracy of 99.1%, outperforming other baseline models, including a simple RNN and ANN.

Feng et al. [126] proposed an approach for detecting Alzheimer's disease (AD) using a 3D-CNN and stacked bidirectional LSTM (BiLSTM). The deep learning model was proposed to learn using both magnetic resonance imaging and positron emission tomography (PET). Using the AD neuroimaging initiative (ADNI) dataset, the proposed deep learning model achieved an accuracy of 94.82%. Similarly, Farooq et al. [127] used CNN for detecting AD. The study also employed the ADNI dataset. Meanwhile, the authors stated that the use of a high-performance graphical processing unit further enhanced the experimental process and time, with the CNN achieving a classification accuracy of 98.8%.

Sharif et al. [128] developed an approach for identifying gastrointestinal tract (GIT) infections using CNN and k-nearest neighbor (KNN). The study used the wireless capsule endoscopy (WCE) dataset, and the disease regions were extracted using a novel technique called contrast-enhanced colour features. The study used the VGG16 and VGG19 for the feature extraction and the KNN for the classification task. The experimental results indicated an accuracy of 99.42%, showing the robustness of the novel approach.

Furthermore, CNN has also played crucial roles in enhancing the quality of medical images through techniques such as denoising, super-resolution, and artefact reduction. These improvements are essential for better visualization and interpretation of the images by medical professionals [129,130]. For example, Kumar et al. [131] proposed a CNN-based approach for identifying brain tumors. The study aimed to achieve enhanced performance and reduced computational complexity by preprocessing the training images in order to remove noise, thereby enhancing the quality of the images. The proposed approach is a fusion of ResNet and transfer learning, which obtained an accuracy of 99.57%.

CNNs have also been used for automated annotation of medical images, which can significantly reduce the manual effort required in labelling anatomical structures and abnormalities. This application is time-saving and helps create large annotated datasets necessary for training deep learning models. For instance, Dolz et al. [132] proposed a method for suggesting annotations within images using a 3D CNN ensemble. The annotation was a step in an overall model aimed at infant brain MRI segmentation. The study demonstrated the capability of CNNs in the automated annotation of medical images. A summary of the reviewed CNN applications in medical imaging is provided in Table 2.

Table 2. Summary of Medical Imaging Papers

Reference	Year	Architecture	Application	Accuracy(%)
Farooq et al. [127]	2017	CNN	Alzheimer's disease	98.8
Khatamino et al. [?]]	2018	CNN	Parkinson's disease	88
Feng et al. [126]	2019	3D-CNN and stacked BiLSTM	Alzheimer's disease	94.82
Sharif et al. [128]	2019	CNN and KNN	Gastrointestinal tract infection	99.42
Yadav and Jadhav [118]	2019	VGG16	Pneumonia	93.8
Islam et al. [124]	2020	CNN and LSTM	COVID-19 disease	99.4
Pan et al. [125]	2020	CNN and MLP	Heart Disease	99.1
Dolz al al. [132]	2020	3D CNN ensemble	Automated annotation of medical images	-
Shaban [123]	2020	VGG-19	Parkinson's disease	88
Alissa et al. [120]	2021	CNN	Parkinson's disease	93.5
Jahan et al. [121]	2021	Inception-V3	Parkinson's disease	96.6
Kumar et al. [131]	2022	CNN	Enhancing image quality for Brain Tumor	99.57
Kurmi et al. [122]	2022	ensemble of VGG16, Inception-V3, ResNet50, and Xception	Parkinson's disease	98.45

8.2. Remote Sensing

Remote sensing refers to the gathering and interpretation of data without direct physical contact. Much research is being done on the use of Deep CNNs for interpreting data gathered by long-range sensing technologies. One application of this is vehicle detection, which has militaristic advantages. Wang et. al [133] implemented a lightweight framework for aircraft detection called CGC-Net. It utilises a CNN consisting of three convolutional layers, two pooling layers, one dropout layer, and one dense layer. This framework achieved an accuracy of 83.59%, outperforming other benchmark models, including an R-CNN. Remote sensing can also be used for environmental protection purposes. For example, work has been done to identify oil spills from satellite imagery. Huang et al. [134] used a dataset built from 1786C-band Sentinel-1 and RADARSAT-2 vertical polarization synthetic aperture radar (SAR) images containing 15,774 labelled examples of oil spills. They trained and tested a Faster R-CNN model to detect the spills and achieved precision and recall of 89.23% and 89.14%, respectively.

Furthermore, CNNs have significantly improved the accuracy of land cover and land use classification, a fundamental task in remote sensing. By extracting hierarchical features from satellite images, CNN models like U-Net [135] and SegNet [136] have demonstrated superior performance in distinguishing between various land cover types, such as forests, water bodies, urban areas, and agricultural land. This enhanced classification capability is vital for tracking environmental changes, urban development, and managing natural resources. For instance, Carranza-García [137] proposed a CNN-based model for land use and land cover (LULC) classification. The proposed method uses remote sensing data, including radar and hyperspectral data. The method comprises 2D CNN, having two convolution layers and max-pooling, reaching an accuracy of 99.36% on the Flevoland dataset.

Memon et al. [138] also used CNN for land cover classification using the RISAT-1 dataset over the Indian region of Mumbai. The CNN obtained an accuracy of 98.37%. Similarly, Dewangkoro and Arymurthy [139] employed CNN and SVM for land use and land cover classification using the Eurosat remote sensing dataset. The study employed different CNN architectures for feature extraction, including InceptionV3, VGG19, and ResNet50. Meanwhile, the SVM and twin-SVM (TSVM) were used as the classifiers. The experimental results showed that the VGG19 architecture and TSVM obtained the best performance with an accuracy of 94.7%.

The application of CNNs in object detection and recognition in remote sensing images has opened new avenues for detailed Earth observation [140,141]. High-resolution satellite images analyzed

with CNNs enable the identification of specific objects like vehicles, buildings, and trees. This technology is crucial for urban planning, monitoring infrastructure development, and conducting detailed environmental studies. For instance, Krishnaraj et al. [142] employed a discrete wavelet transform (DWT) based CNN model for real-time image processing, and Wang et al. [143] used Faster Region-cNN (Faster R-CNN) for real-time vehicle type identification, achieving excellent performances.

CNNs have been applied in agriculture to monitor crop health, predict yields, and detect diseases or pests. Satellite and UAV images processed with CNNs provide valuable insights into crop conditions, enabling precise agriculture practices. Radha and Swathika [144] used CNN for plant monitoring and detecting diseases using an image dataset of plants comprising healthy and diseased leaves. The plants included Grape, Corn, Tomato, and Strawberry. The CNN model achieved an accuracy of 85%. Similarly, Sankar et al. [145] employed CNN for evaluating crop health using a multi-stage approach of image acquisition, preprocessing, image augmentation, and prediction. The CNN obtained a classification accuracy of 85%. Other studies that employed CNN for plant disease classification include [146–149]

While CNNs have significantly advanced remote sensing applications, there are certain challenges that remain and have gotten the attention of researchers in the field, including the need for large labelled datasets, the interpretation of complex CNN models, and ensuring the models’ generalizability across different regions and conditions [150,151]. Meanwhile, the reviewed remote sensing research works are summarized in Table 3.

Table 3. Summary of Remote Sensing Papers

Reference	Year	Architecture	Application	Accuracy(%)
Wang et al. [143]	2017	Faster R-CNN	Real-time vehicle type identification	90.65
Carranza-García [137]	2019	2D CNN	land use and land cover	99.36
Krishnaraj et al. [142]	2019	DWT-based CNN	Real-time image processing	-
Agarwal et al. [147]	2020	VGG16	Plant disease prediction	93.5
Li et al. [146]	2020	CNN and Kernel SVM	Plant disease prediction	-
Qu et. al [152]	2021	SCONE Ensemble with 4 convolutional blocks)	Supernovae classification	98.18 ± 0.3%
Memon et al. [138]	2021	CNN	land cover	98.37
Dewangkoro and Arymurthy [139]	2021	VGG19 and TSVM	Land use and land cover	95.7
Radha and Swathika [144]	2021	CNN	Plant monitoring and disease prediction	85
Bansal et al. [149]	2021	Ensemble of DenseNet and EfficientNet	Plant disease prediction	96.25
Sankar et al. [145]	2021	CNN	Crop health classification	85
Wang et. al [133]	2022	Lightweight CNN	Aircraft detection	83.59%
Huang et. al [134]	2022	Faster R-CNN	Oil spill detection	-
Moradi and Sharifi [153]	2023	4-CNN Ensemble	Forest Cover Change	95%
Kamdi and Biradar [148]	2023	Ensemble of ResNet101 and VGG16	Crop health and classification	99%

8.3. Face Recognition

Convolutional Neural Networks are the foundational technology in the advancement of facial recognition systems. CNNs have enabled significant breakthroughs in accuracy, speed, and reliability in facial recognition. One area where CNNs have played a key role is in enhancing recognition accuracy.

Deep CNNs, through layers of feature extraction, have the capability to identify subtle facial features with a high degree of precision, even in challenging conditions such as varying lighting, angles, and facial expressions [154]. Studies have shown that architectures like DeepFace [155] and FaceNet [156] have achieved near-human levels of accuracy in face verification tasks. Sahan et al. [157] developed a CNN-based model to enhance facial recognition accuracy. The approach employed a deep 1D CNN classifier together with a linear discriminant analysis (LDA) algorithm, leading to an enhancement in the facial recognition performance. The experimental results indicated that the proposed model achieved an accuracy of 100%.

One of the primary applications of CNNs in facial recognition is extracting distinctive facial features. Deep CNN models can accurately identify unique facial characteristics, such as the distance between the eyes, nose shape, and jawline contour, which are crucial for matching faces with existing databases. The work by Wang et al. [158] focused on obtaining fine-grained feature extraction to enhance the discrimination of facial features. The study employed an enhanced CNN that incorporates a new operator based on the local feature descriptor with the aim of achieving the extraction of fine-grained features of disordered point clouds, leading to an accuracy of 98.9%. Similarly, Awaji et al. [159] proposed an approach for facial feature extraction by integrating a hybrid random forest algorithm with CNN. The random forest utilises features from the VGG16-MobileNet model, obtaining an accuracy of 98.8%.

Furthermore, CNNs have been instrumental in enabling real-time facial recognition, a critical requirement for applications in security and surveillance. The efficiency of CNN architectures allows for the rapid processing of video feeds, enabling the immediate identification and tracking of individuals in live scenarios. This capability is essential for public safety, border control, and preventing unauthorized access to secure locations. For instance, Ullah et al. [160] developed an approach for real-time anomaly detection within a surveillance network using CNN and BiLSTM. The proposed approach involves extracting spatiotemporal features from a sequence of frames using the ResNet-50. Meanwhile, the extracted features were passed to the BiLSTM model for classification. The experimental results showed the proposed approach is able to classify normal/anomaly events in complex surveillance scenes, achieving 85.53% accuracy using the ResNet-50 and BiLSTM.

Rajeshkumar et. al [161] proposed a security system that is functional in an office environment to ensure only employees can pass through a door. A database containing images of the employees was utilised for training. The study employed a Faster R-CNN with VGG-16 architecture. The Faster R-CNN was used to generate bounding boxes, and the VGG-16, pre-trained on the ImageNet dataset, was then used for facial landmark extraction. The approach obtains an accuracy of 99.03%, outperforming other algorithms. Saravanan et al. [162] proposed a novel face mask identification model for security purposes. The justification for the implementation is due to the difficulty of tracking if members of the public were wearing face masks during the COVID-19 outbreak. A transfer learning approach is utilised, with the model of choice being a pre-trained VGG-16 model in which all the layers were functional, and only the fully connected layer was trained to reduce training time and cost. Specifically, the VGG-16 architecture contained 13 convolutional layers, 5 max-pooling layers, and 3 fully connected layers. Two face mask datasets were used: one with 1484 pictures and another with 7200, and VGG-16 obtained an accuracy of 96.5% and 91%, respectively.

CNNs have also been used in virtual facial recognition. For example, Chirra et al. [163] developed an approach to detect the facial emotions of virtual characters using a DCNN architecture. The study employed a multi-block DCNN comprising four blocks to extract different discriminative facial attributes from the input images. Meanwhile, in order to enhance the stability and improve the classification performance, the study proposed using an ensemble of DCNN and SVM, leading to enhanced prediction performance with an accuracy of 99.57% on the Japanese female facial expressions (JAFPE) dataset. In a similar facial expression recognition research, Febrian et. al [164] utilised a combination of BiLSTM and CNN to obtain an accuracy of 99.43%.

Mehendale et al. [165] utilised a 2-part CNN architecture. The first part was for background removal and the second part was aimed at facial feature vector extraction. A 24-component expressional vector generated after background removal was used to identify the 5 possible target facial expressions. The training dataset used contained 10,000 images from 154 distinct individuals, and the proposed method achieved an accuracy of 96%. Debnath et al. [166] used a ConvNet CNN model to detect seven possible emotions from the image datasets. The study utilised the local binary pattern (LBP) for Feature extraction to produce an image feature map. LBP was utilised due to its resistance to fluctuations in light conditions. The proposed model achieved a 92.05% accuracy on the JAFFE set and 98.13% on the CK+ dataset.

Agrawal et al. [6] investigated the impact of CNN parameters on the prediction performance of models using the FER-2013 dataset. The parameters include the number of filters and kernel size. The study concluded by highlighting the significant impact of both parameters on the performance of the CNN and reported that the CNN architecture with a kernel size of 8 and 32 filters achieved a human-level accuracy of 65%. Other facial expression recognition studies have been developed by Almabdy and Elrefael [167], Bendjillali et al. [168], Alkhand et al. [169], and Li et al. [170].

Despite the significant advancements, the field of facial recognition with CNN architectures continues to evolve. More research is still being done to further enhance the accuracy and speed of recognition, reducing biases in facial recognition algorithms and improving the ability to detect spoofing and impersonation attempts [171,172]. Table 4 summarises the reviewed facial recognition studies.

Table 4. Summary of Face Recognition Papers

Reference	Year	Architecture	Application	Accuracy(%)
Lu et al. [173]	2017	VGG-Face	Enhancing facing recognition	-
Almabdy and Elrefael [167]	2019	CNN AlexNet with SVM	Face recognition	-
Bendjillali et al. [168]	2019	DCNN with DWT	Facial Expression Recognition	98.63
Mehendale et al. [165]	2020	2-part CNN	Facial Emotion Recognition	96.0
Agrawal et al. [6]	2020	CNN with a kernel size of 8 and 32 filters	Facial expression recognition	65
Li et al. [170]	2020	Attention mechanism-based CNN	Facial expression recognition	94.33
Chirra et al. [163]	2021	DCNN-SVM	Virtual facial expression	99.57
Debnath et al. [166]	2021	ConvNet	Facial Emotion Recognition	98.13
Alkhand et al. [169]	2021	Deep CNN	Facial Emotion Recognition	96.51
Ullah et al. [160]	2021	ResNet-50 and BiLSTM	Real-time anomaly detection	85.53
Febrian et al. [164]	2022	BiLSTM and CNN	Facial Expression Recognition	99.43
Saravanan et al. [162]	2022	VGG-16	Masked face detection	96.5
Vu et al. [174]	2022	CNN and LBP	Masked face recognition	-
Sahan et al. [157]	2023	1D CNN and LDA	Enhancing facing recognition	100
Singh et al. [175]	2023	3D-CNN and ConvLSTM	Facial Expression Recognition	95.1
Dang [176]	2023	FaceNet with SSD	Attendance System	95.0
Rajeshkumar et al. [161]	2023	Faster R-CNN	Smart Office Automation	99.03
Wang et al. [158]	2023	Enhanced CNN	Facial Feature Extraction	98.9
Awaji et al. [159]	2023	VGG16-MobileNet	Facial Feature Extraction	98.8
Hangaragi et al. [177]	2023	Face Mesh	Face Recognition	94.23

9. Challenges and Future Research Directions

Despite the success of CNN, several challenges exist, necessitating ongoing research to enhance the capabilities and efficiency of Deep CNNs. This section outlines key challenges faced by current CNN architectures and potential future research directions to address these issues.

One of the significant challenges for Deep CNNs is the requirement for large annotated datasets to achieve high performance [178]. CNNs have shown impressive performance on tasks with large datasets, and they often struggle when faced with limited or imbalanced data. This can lead to biases in the model and reduce its ability to generalize to new, unseen data. This dependence demands extensive data collection and labelling resources and limits the ability of CNNs to generalize from small datasets. Research could focus on developing more efficient learning algorithms that require less data to train. Additionally, exploring transfer learning and few-shot learning could provide pathways to better generalization from limited examples.

The training and deployment of state-of-the-art CNN models often require significant computational resources and energy, which can be prohibitive for real-time applications or deployment on edge devices [179]. CNNs are computationally expensive and require significant resources in terms of both time and hardware. Efforts could be directed toward creating more efficient CNN architectures that maintain high accuracy while reducing computational complexity and energy consumption. Techniques such as network pruning, quantization, and knowledge distillation offer promising approaches to model optimization.

CNNs are often criticized for their "black box" nature, making it challenging to understand how they arrive at specific decisions. While CNNs are able to accurately classify images and detect objects, it is often difficult to understand how these decisions are being made. This lack of transparency is a critical issue in fields where explainability is essential, such as healthcare and autonomous driving. Developing methods to enhance the interpretability of CNNs without compromising their performance could bridge the gap between AI and user trust. Research into explainable AI (XAI) aims to make models more understandable to humans, facilitating broader acceptance and ethical deployment.

The issue of bias in machine learning models, including CNNs, is well-documented [180–182]. Biases in training data can lead to skewed or unfair outcomes, affecting the model's fairness and ethical implications. Addressing bias and ensuring fairness in CNN models requires concerted efforts in developing unbiased data collection practices, bias detection and mitigation techniques, and fairness-aware machine learning algorithms. Addressing these challenges through innovative research will be essential for achieving more versatile, efficient, and ethically responsible AI systems. Future advancements in CNN architectures and learning algorithms promise to unlock new capabilities and applications, further expanding the impact of deep learning across diverse fields.

10. Conclusions

This paper has provided a comprehensive review of Deep CNNs, including their main building blocks, evolution, architectures, and applications across diverse fields such as medical imaging, facial recognition, and remote sensing. The study showed how CNNs drive advancements in accuracy, efficiency, and automation, offering innovative solutions to complex problems. Despite these successes, challenges remain in the form of data dependency, computational demands, interpretability, and ethical concerns. Future research directions have been identified to address these issues, emphasizing the need for efficient learning algorithms, improved model robustness, and fairness in AI applications.

References

1. Sultana, F.; Sufian, A.; Dutta, P. A review of object detection models based on convolutional neural network. *Intelligent computing: image processing based applications* **2020**, pp. 1–16.
2. Bernal, J.; Kushibar, K.; Asfaw, D.S.; Valverde, S.; Oliver, A.; Martí, R.; Lladó, X. Deep convolutional neural networks for brain image analysis on magnetic resonance imaging: a review. *Artificial Intelligence in Medicine* **2019**, *95*, 64–81. doi:https://doi.org/10.1016/j.artmed.2018.08.008.
3. Sun, Y.; Xue, B.; Zhang, M.; Yen, G.G. Evolving Deep Convolutional Neural Networks for Image Classification. *IEEE Transactions on Evolutionary Computation* **2020**, *24*, 394–407. doi:10.1109/TEVC.2019.2916183.
4. Meena, G.; Mohbey, K.K.; Indian, A.; Khan, M.Z.; Kumar, S. Identifying emotions from facial expressions using a deep convolutional neural network-based approach. *Multimedia Tools and Applications* **2023**, *83*, 15711–15732. doi:10.1007/s11042-023-16174-3.
5. Russel, N.S.; Selvaraj, A. MultiScaleCrackNet: A parallel multiscale deep CNN architecture for concrete crack classification. *Expert Systems with Applications* **2024**, *249*, 123658. doi:https://doi.org/10.1016/j.eswa.2024.123658.
6. Agrawal, A.; Mittal, N. Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy. *The Visual Computer* **2019**, *36*, 405–412. doi:10.1007/s00371-019-01630-9.
7. Alzubaidi, L.; Zhang, J.; Humaidi, A.J.; Al-Dujaili, A.; Duan, Y.; Al-Shamma, O.; Santamaría, J.; Fadhel, M.A.; Al-Amidie, M.; Farhan, L. Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data* **2021**, *8*, 1–74.
8. Taye, M.M. Understanding of Machine Learning with Deep Learning: Architectures, Workflow, Applications and Future Directions. *Computers* **2023**, *12*. doi:10.3390/computers12050091.
9. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; Chen, T. Recent advances in convolutional neural networks. *Pattern Recognition* **2018**, *77*, 354–377. doi:https://doi.org/10.1016/j.patcog.2017.10.013.
10. Acharya, U.R.; Oh, S.L.; Hagiwara, Y.; Tan, J.H.; Adam, M.; Gertych, A.; Tan, R.S. A deep convolutional neural network model to classify heartbeats. *Computers in Biology and Medicine* **2017**, *89*, 389–396. doi:https://doi.org/10.1016/j.compbimed.2017.08.022.
11. Krichen, M. Convolutional Neural Networks: A Survey. *Computers* **2023**, *12*. doi:10.3390/computers12080151.
12. Mienye, I.D.; Kenneth Aina, P.; Emmanuel, I.D.; Esenogho, E. Sparse noise minimization in image classification using Genetic Algorithm and DenseNet. 2021 Conference on Information Communications Technology and Society (ICTAS), 2021, pp. 103–108. doi:10.1109/ICTAS50802.2021.9395014.
13. Ahmed, H.; Hamad, S.; Shedeed, H.A.; Hussein, A.S. Enhanced Deep Learning Model for Personalized Cancer Treatment. *IEEE Access* **2022**, *10*, 106050–106058. doi:10.1109/ACCESS.2022.3209285.
14. Guha, S.; Jana, R.K.; Sanyal, M.K. Artificial neural network approaches for disaster management: A literature review. *International Journal of Disaster Risk Reduction* **2022**, *81*, 103276. doi:https://doi.org/10.1016/j.ijdrr.2022.103276.
15. Li, Z.; Liu, F.; Yang, W.; Peng, S.; Zhou, J. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems* **2021**.
16. Krichen, M. Convolutional neural networks: A survey. *Computers* **2023**, *12*, 151.
17. Dhillon, A.; Verma, G.K. Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence* **2020**, *9*, 85–112.
18. Yamashita, R.; Nishio, M.; Do, R.K.G.; Togashi, K. Convolutional neural networks: an overview and application in radiology. *Insights into imaging* **2018**, *9*, 611–629.
19. Cong, S.; Zhou, Y. A review of convolutional neural network architectures and their optimizations. *Artificial Intelligence Review* **2023**, *56*, 1905–1969.
20. Kamilaris, A.; Prenafeta-Boldú, F.X. A review of the use of convolutional neural networks in agriculture. *The Journal of Agricultural Science* **2018**, *156*, 312–322.
21. Lu, J.; Tan, L.; Jiang, H. Review on convolutional neural network (CNN) applied to plant leaf disease classification. *Agriculture* **2021**, *11*, 707.

22. Kattenborn, T.; Leitloff, J.; Schiefer, F.; Hinz, S. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. *ISPRS journal of photogrammetry and remote sensing* **2021**, *173*, 24–49.
23. Liu, X.; Ghazali, K.H.; Han, F.; Mohamed, I.I. Review of CNN in aerial image processing. *The Imaging Science Journal* **2023**, *71*, 1–13.
24. Yu, J.; de Antonio, A.; Villalba-Mora, E. Deep learning (CNN, RNN) applications for smart homes: a systematic review. *Computers* **2022**, *11*, 26.
25. Zhiqiang, W.; Jun, L. A review of object detection based on convolutional neural network. 2017 36th Chinese control conference (CCC). IEEE, 2017, pp. 11104–11109.
26. Ji, Y.; Zhang, H.; Zhang, Z.; Liu, M. CNN-based encoder-decoder networks for salient object detection: A comprehensive review and recent advances. *Information Sciences* **2021**, *546*, 835–857.
27. Sindagi, V.A.; Patel, V.M. A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern Recognition Letters* **2018**, *107*, 3–16.
28. Jiao, J.; Zhao, M.; Lin, J.; Liang, K. A comprehensive review on convolutional neural network in machine fault diagnosis. *Neurocomputing* **2020**, *417*, 36–63.
29. Wen, L.; Li, X.; Gao, L.; Zhang, Y. A new convolutional neural network-based data-driven fault diagnosis method. *IEEE Transactions on Industrial Electronics* **2017**, *65*, 5990–5998.
30. Tang, S.; Yuan, S.; Zhu, Y. Convolutional neural network in intelligent fault diagnosis toward rotatory machinery. *IEEE Access* **2020**, *8*, 86510–86519.
31. Anwar, S.M.; Majid, M.; Qayyum, A.; Awais, M.; Alnowami, M.; Khan, M.K. Medical image analysis using convolutional neural networks: a review. *Journal of medical systems* **2018**, *42*, 1–13.
32. Sarvamangala, D.; Kulkarni, R.V. Convolutional neural networks in medical image understanding: a survey. *Evolutionary intelligence* **2022**, *15*, 1–22.
33. Lu, L.; Zheng, Y.; Carneiro, G.; Yang, L. Deep learning and convolutional neural networks for medical image computing. *Advances in computer vision and pattern recognition* **2017**, *10*, 978–3.
34. Fogel, D.B. Defining Artificial Intelligence, 2022. doi:10.1002/9781119815075.ch7.
35. Baştanlar, Y.; Özuysal, M. Introduction to machine learning. *miRNomics: MicroRNA biology and computational analysis* **2014**, pp. 105–128.
36. Al-Baity, H.H. The Artificial Intelligence Revolution in Digital Finance in Saudi Arabia: A Comprehensive Review and Proposed Framework. *Sustainability* **2023**, *15*. doi:10.3390/su151813725.
37. Kumaraswamy, B. Neural networks for data classification. In *Artificial intelligence in data mining*; Elsevier, 2021; pp. 109–131.
38. Mienye, I.D.; Sun, Y. A Deep Learning Ensemble With Data Resampling for Credit Card Fraud Detection. *IEEE Access* **2023**, *11*, 30628–30638.
39. Ibnu Choldun R., M.; Santoso, J.; Surendro, K. Determining the Number of Hidden Layers in Neural Network by Using Principal Component Analysis. *Intelligent Systems and Applications*; Bi, Y.; Bhatia, R.; Kapoor, S., Eds.; Springer International Publishing: Cham, 2020; pp. 490–500.
40. Caterini, A.L.; Chang, D.E. *Deep neural networks in a mathematical framework*; Springer, 2018.
41. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26.
42. Afsari, R.; Nadizadeh Shorabeh, S.; Bakhshi Lomer, A.R.; Homaei, M.; Arsanjani, J.J. Using Artificial Neural Networks to Assess Earthquake Vulnerability in Urban Blocks of Tehran. *Remote Sensing* **2023**, *15*, 1248.
43. Shwartz-Ziv, R.; Tishby, N. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810* **2017**.
44. Dubey, S.R.; Singh, S.K.; Chaudhuri, B.B. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing* **2022**.
45. Ide, H.; Kurita, T. Improvement of learning for CNN with ReLU activation by sparse regularization. 2017 international joint conference on neural networks (IJCNN). IEEE, 2017, pp. 2684–2691.
46. Dubey, A.K.; Jain, V. Comparative study of convolution neural network's relu and leaky-relu activation functions. *Applications of Computing, Automation and Wireless Systems in Electrical Engineering: Proceedings of MARC 2018*. Springer, 2019, pp. 873–880.
47. Clevert, D.A.; Unterthiner, T.; Hochreiter, S. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289* **2015**.

48. Zhang, Y.D.; Pan, C.; Chen, X.; Wang, F. Abnormal breast identification by nine-layer convolutional neural network with parametric rectified linear unit and rank-based stochastic pooling. *Journal of computational science* **2018**, *27*, 57–68.
49. Mathieu, M.; Couprie, C.; LeCun, Y. Deep multi-scale video prediction beyond mean square error. 4th International Conference on Learning Representations, ICLR 2016, 2016.
50. Gupta, D.; Hazarika, B.B.; Berlin, M. Robust regularized extreme learning machine with asymmetric Huber loss function. *Neural Computing and Applications* **2020**, *32*, 12971–12998.
51. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.
52. Zhang, Z.; Sabuncu, M. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems* **2018**, *31*.
53. Abdolrasol, M.G.; Hussain, S.S.; Ustun, T.S.; Sarker, M.R.; Hannan, M.A.; Mohamed, R.; Ali, J.A.; Mekhilef, S.; Milad, A. Artificial neural networks based optimization techniques: A review. *Electronics* **2021**, *10*, 2689.
54. Andrychowicz, M.; Denil, M.; Gomez, S.; Hoffman, M.W.; Pfau, D.; Schaul, T.; Shillingford, B.; De Freitas, N. Learning to learn by gradient descent by gradient descent. *Advances in neural information processing systems* **2016**, *29*.
55. Yuan, W.; Hu, F.; Lu, L. A new non-adaptive optimization method: Stochastic gradient descent with momentum and difference. *Applied Intelligence* **2022**, pp. 1–15.
56. Khan, M.U.; Jawad, M.; Khan, S.U. Adadb: Adaptive diff-batch optimization technique for gradient descent. *IEEE Access* **2021**, *9*, 99581–99588.
57. Feyzmahdavian, H.R.; Aytakin, A.; Johansson, M. An asynchronous mini-batch algorithm for regularized stochastic optimization. *IEEE Transactions on Automatic Control* **2016**, *61*, 3740–3754.
58. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* **2014**.
59. Feng, P.; Tang, Z. A survey of visual neural networks: current trends, challenges and opportunities. *Multimedia Systems* **2023**, *29*, 693–724.
60. Phung, V.H.; Rhee, E.J. A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets. *Applied Sciences* **2019**, *9*. doi:10.3390/app9214500.
61. Taye, M.M. Theoretical understanding of convolutional neural network: Concepts, architectures, applications, future directions. *Computation* **2023**, *11*, 52.
62. Seyrek, E.C.; Uysal, M. A comparative analysis of various activation functions and optimizers in a convolutional neural network for hyperspectral image classification. *Multimedia Tools and Applications* **2023**, pp. 1–32.
63. Ozdemir, C.; Dogan, Y.; Kaya, Y. A new local pooling approach for convolutional neural network: local binary pattern. *Multimedia Tools and Applications* **2024**, *83*, 34137–34151.
64. Walter, B. Analysis of convolutional neural network image classifiers in a hierarchical max-pooling model with additional local pooling. *Journal of Statistical Planning and Inference* **2023**, *224*, 109–126.
65. Reus-Muns, G.; Alemdar, K.; Sanchez, S.G.; Roy, D.; Chowdhury, K.R. AirFC: Designing Fully Connected Layers for Neural Networks with Wireless Signals. Proceedings of the Twenty-fourth International Symposium on Theory, Algorithmic Foundations, and Protocol Design for Mobile Networks and Mobile Computing, 2023, pp. 71–80.
66. Zhu, S.; Yu, C.; Hu, J. Regularizing deep neural networks for medical image analysis with augmented batch normalization. *Applied Soft Computing* **2024**, p. 111337.
67. Salehin, I.; Kang, D.K. A review on dropout regularization approaches for deep neural networks within the scholarly domain. *Electronics* **2023**, *12*, 3106.
68. Garbin, C.; Zhu, X.; Marques, O. Dropout vs. batch normalization: an empirical study of their impact to deep learning. *Multimedia tools and applications* **2020**, *79*, 12777–12815.
69. Tulbure, A.A.; Tulbure, A.A.; Dulf, E.H. A review on modern defect detection models using DCNNs–Deep convolutional neural networks. *Journal of Advanced Research* **2022**, *35*, 33–48.
70. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems*; Pereira, F.; Burges, C.; Bottou, L.; Weinberger, K., Eds. Curran Associates, Inc., 2012, Vol. 25.
71. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition, 2015, [arXiv:cs.CV/1512.03385].

72. Lecun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **1998**, *86*, 2278–2324. doi:10.1109/5.726791.
73. Balasubramaniam, S.; Velmurugan, Y.; Jaganathan, D.; Dhanasekaran, S. A modified LeNet CNN for breast cancer diagnosis in ultrasound images. *Diagnostics* **2023**, *13*, 2746.
74. Cong, S.; Zhou, Y. A review of convolutional neural network architectures and their optimizations. *Artificial Intelligence Review* **2022**, *56*, 1905–1969. doi:10.1007/s10462-022-10213-5.
75. Pan, T.S.; Huang, H.C.; Lee, J.C.; Chen, C.H. Multi-scale ResNet for real-time underwater object detection. *Signal, Image and Video Processing* **2021**, *15*, 941–949.
76. Wang, H.; Li, K.; Xu, C. A new generation of ResNet model based on artificial intelligence and few data driven and its construction in image recognition model. *Computational Intelligence and Neuroscience* **2022**, 2022.
77. Duta, I.C.; Liu, L.; Zhu, F.; Shao, L. Improved residual networks for image and video recognition. 2020 25th International Conference on Pattern Recognition (ICPR). IEEE, 2021, pp. 9415–9422.
78. Veniat, T.; Denoyer, L. Learning time/memory-efficient deep architectures with budgeted super networks. Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3492–3500.
79. Zhu, Z.; Zhai, W.; Liu, H.; Geng, J.; Zhou, M.; Ji, C.; Jia, G. Juggler-ResNet: A flexible and high-speed ResNet optimization method for intrusion detection system in software-defined industrial networks. *IEEE Transactions on Industrial Informatics* **2021**, *18*, 4224–4233.
80. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* **2014**.
81. Wang, Z.; Zheng, X.; Li, D.; Zhang, H.; Yang, Y.; Pan, H. A VGGNet-like approach for classifying and segmenting coal dust particles with overlapping regions. *Computers in Industry* **2021**, *132*, 103506.
82. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition, 2015, [arXiv:cs.CV/1409.1556].
83. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions, 2014, [arXiv:cs.CV/1409.4842].
84. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261–2269. doi:10.1109/CVPR.2017.243.
85. He, Z.; Yuan, S.; Zhao, J.; Du, B.; Yuan, Z.; Alhudhaif, A.; Alenezi, F.; Althubiti, S.A. A novel myocardial infarction localization method using multi-branch DenseNet and spatial matching-based active semi-supervised learning. *Information Sciences* **2022**, *606*, 649–668.
86. Kang, H.; Yang, Y. An Enhanced Detection Method of PCB Defect Based on D-DenseNet (PCBDD-DDNet). *Electronics* **2023**, *12*, 4737.
87. Huang, Z.; Zhu, X.; Ding, M.; Zhang, X. Medical image classification using a light-weighted hybrid neural network based on PCANet and DenseNet. *Ieee Access* **2020**, *8*, 24697–24712.
88. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. International conference on machine learning. PMLR, 2019, pp. 6105–6114.
89. Lim, J.; Lee, S.; Ha, S. Resolution based Incremental Scaling Methodology for CNNs. *IEEE Access* **2023**.
90. Lin, C.; Yang, P.; Wang, Q.; Qiu, Z.; Lv, W.; Wang, Z. Efficient and accurate compound scaling for convolutional neural networks. *Neural Networks* **2023**, *167*, 787–797.
91. Abedalla, A.; Abdullah, M.; Al-Ayyoub, M.; Benkhelifa, E. Chest X-ray pneumothorax segmentation using U-Net with EfficientNet and ResNet architectures. *PeerJ Computer Science* **2021**, *7*, e607.
92. Shah, H.A.; Saeed, F.; Yun, S.; Park, J.H.; Paul, A.; Kang, J.M. A robust approach for brain tumor detection in magnetic resonance images using finetuned efficientnet. *Ieee Access* **2022**, *10*, 65426–65438.
93. Papoutsis, I.; Bountos, N.I.; Zavras, A.; Michail, D.; Tryfonopoulos, C. Benchmarking and scaling of deep learning models for land cover image classification. *ISPRS Journal of Photogrammetry and Remote Sensing* **2023**, *195*, 250–268.
94. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; others. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence* **2020**, *43*, 3349–3364.

95. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
96. Larsson, G.; Maire, M.; Shakhnarovich, G. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648* **2016**.
97. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision, 2015, [[arXiv:cs.CV/1512.00567](https://arxiv.org/abs/1512.00567)].
98. Srivastava, R.K.; Greff, K.; Schmidhuber, J. Highway Networks, 2015, [[arXiv:cs.LG/1505.00387](https://arxiv.org/abs/1505.00387)].
99. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, 2016, [[arXiv:cs.CV/1602.07261](https://arxiv.org/abs/1602.07261)].
100. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. *Proceedings of the AAAI conference on artificial intelligence*, 2017, Vol. 31.
101. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions, 2017, [[arXiv:cs.CV/1610.02357](https://arxiv.org/abs/1610.02357)].
102. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual attention network for image classification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3156–3164.
103. Helaly, H.A.; Badawy, M.; Haikal, A.Y. Deep learning approach for early detection of Alzheimer's disease. *Cognitive computation* **2022**, *14*, 1711–1727.
104. Arooj, S.; Rehman, S.u.; Imran, A.; Almuhaimeed, A.; Alzahrani, A.K.; Alzahrani, A. A deep convolutional neural network for the early detection of heart disease. *Biomedicines* **2022**, *10*, 2796.
105. Li, E.; Samat, A.; Liu, W.; Lin, C.; Bai, X. High-resolution imagery classification based on different levels of information. *Remote Sensing* **2019**, *11*, 2916.
106. Xia, L.; Chen, G.; Xu, X.; Cui, J.; Gao, Y. Audiovisual speech recognition: A review and forecast. *International Journal of Advanced Robotic Systems* **2020**, *17*, 1729881420976082.
107. Zhao, L.; Qiu, X.; Zhang, Q.; Huang, X. Sequence labeling with deep gated dual path CNN. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **2019**, *27*, 2326–2335.
108. Shimura, K.; Li, J.; Fukumoto, F. HFT-CNN: Learning hierarchical category structure for multi-label short text categorization. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 811–816.
109. Peng, H.; Li, J.; He, Y.; Liu, Y.; Bao, M.; Wang, L.; Song, Y.; Yang, Q. Large-scale hierarchical text classification with recursively regularized deep graph-cnn. *Proceedings of the 2018 world wide web conference*, 2018, pp. 1063–1072.
110. Li, C.; Zhan, G.; Li, Z. News text classification based on improved Bi-LSTM-CNN. 2018 9th International conference on information technology in medicine and education (ITME). IEEE, 2018, pp. 890–893.
111. Huang, J.X.; Lee, K.S.; Kim, Y.K. Hybrid translation with classification: Revisiting rule-based and neural machine translation. *Electronics* **2020**, *9*, 201.
112. Wu, L.; Xia, Y.; Tian, F.; Zhao, L.; Qin, T.; Lai, J.; Liu, T.Y. Adversarial neural machine translation. *Asian Conference on Machine Learning*. PMLR, 2018, pp. 534–549.
113. Jogin, M.; Madhulika, M.; Divya, G.; Meghana, R.; Apoorva, S.; others. Feature extraction using convolution neural networks (CNN) and deep learning. 2018 3rd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT). IEEE, 2018, pp. 2319–2323.
114. Anantrasirichai, N.; Bull, D. Artificial intelligence in the creative industries: a review. *Artificial intelligence review* **2022**, *55*, 589–656.
115. Yasrab, R.; Gu, N.; Zhang, X. An encoder-decoder based convolution neural network (CNN) for future advanced driver assistance system (ADAS). *Applied Sciences* **2017**, *7*, 312.
116. De-Las-Heras, G.; Sanchez-Soriano, J.; Puertas, E. Advanced driver assistance systems (ADAS) based on machine learning techniques for the detection and transcription of variable message signs on roads. *Sensors* **2021**, *21*, 5866.
117. Yang, E.; Yi, O. Enhancing Road Safety: Deep Learning-Based Intelligent Driver Drowsiness Detection for Advanced Driver-Assistance Systems. *Electronics* **2024**, *13*, 708.
118. Yadav, S.S.; Jadhav, S.M. Deep convolutional neural network based medical image classification for disease diagnosis. *Journal of Big Data* **2019**, *6*. doi:10.1186/s40537-019-0276-2.

119. Khatamino, P.; Cantürk, İ.; Özyılmaz, L. A deep learning-CNN based system for medical diagnosis: An application on Parkinson's disease handwriting drawings. 2018 6th International Conference on Control Engineering & Information Technology (CEIT). IEEE, 2018, pp. 1–6.
120. Alissa, M.; Lones, M.A.; Cosgrove, J.; Alty, J.E.; Jamieson, S.; Smith, S.L.; Vallejo, M. Parkinson's disease diagnosis using convolutional neural networks and figure-copying tasks. *Neural Computing and Applications* **2021**, *34*, 1433–1453. doi:10.1007/s00521-021-06469-7.
121. Jahan, N.; Nesa, A.; Layek, M.A. Parkinson's Disease Detection Using CNN Architectures with Transfer Learning. 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), 2021, pp. 1–5. doi:10.1109/ICSES52305.2021.9633872.
122. Kurmi, A.; Biswas, S.; Sen, S.; Sinitca, A.; Kaplun, D.; Sarkar, R. An Ensemble of CNN Models for Parkinson's Disease Detection Using DaTscan Images. *Diagnostics* **2022**, *12*. doi:10.3390/diagnostics12051173.
123. Shaban, M. Deep Convolutional Neural Network for Parkinson's Disease Based Handwriting Screening. 2020 IEEE 17th International Symposium on Biomedical Imaging Workshops (ISBI Workshops), 2020, pp. 1–4. doi:10.1109/ISBIWorkshops50223.2020.9153407.
124. Islam, M.Z.; Islam, M.M.; Asraf, A. A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. *Informatics in Medicine Unlocked* **2020**, *20*, 100412. doi:https://doi.org/10.1016/j.imu.2020.100412.
125. Pan, Y.; Fu, M.; Cheng, B.; Tao, X.; Guo, J. Enhanced Deep Learning Assisted Convolutional Neural Network for Heart Disease Prediction on the Internet of Medical Things Platform. *IEEE Access* **2020**, *8*, 189503–189512. doi:10.1109/ACCESS.2020.3026214.
126. Feng, C.; Elazab, A.; Yang, P.; Wang, T.; Zhou, F.; Hu, H.; Xiao, X.; Lei, B. Deep Learning Framework for Alzheimer's Disease Diagnosis via 3D-CNN and FSBI-LSTM. *IEEE Access* **2019**, *7*, 63605–63618. doi:10.1109/ACCESS.2019.2913847.
127. Farooq, A.; Anwar, S.; Awais, M.; Rehman, S. A deep CNN based multi-class classification of Alzheimer's disease using MRI. 2017 IEEE International Conference on Imaging Systems and Techniques (IST), 2017, pp. 1–6. doi:10.1109/IST.2017.8261460.
128. Sharif, M.; Attique Khan, M.; Rashid, M.; Yasmin, M.; Afza, F.; Tanik, U.J. Deep CNN and geometric features-based gastrointestinal tract diseases detection and classification from wireless capsule endoscopy images. *Journal of Experimental & Theoretical Artificial Intelligence* **2019**, *33*, 577–599. doi:10.1080/0952813x.2019.1572657.
129. Yu, H.; Yang, L.T.; Zhang, Q.; Armstrong, D.; Deen, M.J. Convolutional neural networks for medical image analysis: State-of-the-art, comparisons, improvement and perspectives. *Neurocomputing* **2021**, *444*, 92–110. doi:https://doi.org/10.1016/j.neucom.2020.04.157.
130. Salehi, A.W.; Khan, S.; Gupta, G.; Alabdullah, B.I.; Almjally, A.; Alsolai, H.; Siddiqui, T.; Mellit, A. A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope. *Sustainability* **2023**, *15*. doi:10.3390/su15075930.
131. Ananda Kumar, K.; Prasad, A.; Metan, J. A hybrid deep CNN-Cov-19-Res-Net Transfer learning archtype for an enhanced Brain tumor Detection and Classification scheme in medical image processing. *Biomedical Signal Processing and Control* **2022**, *76*, 103631. doi:https://doi.org/10.1016/j.bspc.2022.103631.
132. Dolz, J.; Desrosiers, C.; Wang, L.; Yuan, J.; Shen, D.; Ben Ayed, I. Deep CNN ensembles and suggestive annotations for infant brain MRI segmentation. *Computerized Medical Imaging and Graphics* **2020**, *79*, 101660. doi:https://doi.org/10.1016/j.compmedimag.2019.101660.
133. Wang, T.; Zeng, X.; Cao, C.; Li, W.; Feng, Z.; Wu, J.; Yan, X.; Wu, Z. CGC-NET: Aircraft Detection in Remote Sensing Images Based on Lightweight Convolutional Neural Network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **2022**, *15*, 2805–2815. doi:10.1109/JSTARS.2022.3159981.
134. Huang, X.; Zhang, B.; Perrie, W.; Lu, Y.; Wang, C. A novel deep learning method for marine oil spill detection from satellite synthetic aperture radar imagery. *Marine Pollution Bulletin* **2022**, *179*, 113666. doi:https://doi.org/10.1016/j.marpolbul.2022.113666.
135. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18. Springer, 2015, pp. 234–241.

136. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2017**, *39*, 2481–2495. doi:10.1109/TPAMI.2016.2644615.
137. Carranza-García, M.; García-Gutiérrez, J.; Riquelme, J.C. A Framework for Evaluating Land Use and Land Cover Classification Using Convolutional Neural Networks. *Remote Sensing* **2019**, *11*. doi:10.3390/rs11030274.
138. Memon, N.; Parikh, H.; Patel, S.B.; Patel, D.; Patel, V.D. Automatic land cover classification of multi-resolution dualpol data using convolutional neural network (CNN). *Remote Sensing Applications: Society and Environment* **2021**, *22*, 100491. doi:10.1016/j.rsase.2021.100491.
139. Dewangkoro, H.I.; Arymurthy, A.M. Land Use and Land Cover Classification Using CNN, SVM, and Channel Squeeze & Spatial Excitation Block. *IOP Conference Series: Earth and Environmental Science* **2021**, *704*, 012048. doi:10.1088/1755-1315/704/1/012048.
140. Gui, S.; Song, S.; Qin, R.; Tang, Y. Remote Sensing Object Detection in the Deep Learning Era—A Review. *Remote Sensing* **2024**, *16*. doi:10.3390/rs16020327.
141. Hoesser, T.; Kuenzer, C. Object detection and image segmentation with deep learning on earth observation data: A review-part i: Evolution and recent trends. *Remote Sensing* **2020**, *12*, 1667.
142. Krishnaraj, N.; Elhoseny, M.; Thenmozhi, M.; Selim, M.M.; Shankar, K. Deep learning model for real-time image compression in Internet of Underwater Things (IoUT). *Journal of Real-Time Image Processing* **2019**, *17*, 2097–2111. doi:10.1007/s11554-019-00879-6.
143. Wang, X.; Zhang, W.; Wu, X.; Xiao, L.; Qian, Y.; Fang, Z. Real-time vehicle type classification with deep convolutional neural networks. *Journal of Real-Time Image Processing* **2017**, *16*, 5–14. doi:10.1007/s11554-017-0712-5.
144. Radha, N.; Swathika, R. A Polyhouse: Plant Monitoring and Diseases Detection using CNN. 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 2021, pp. 966–971. doi:10.1109/ICAIS50930.2021.9395847.
145. Sai Sankar, P.R.; RamaKrishna D.P.S, S.; Venkata Rakesh, M.M.; Raja, P.; Hoang, V.T.; Szczepanski, C. Intelligent Health Assessment System for Paddy Crop Using CNN. 2021 3rd International Conference on Signal Processing and Communication (ICPSC), 2021, pp. 382–387. doi:10.1109/ICSPC51351.2021.9451644.
146. Li, Y.; Nie, J.; Chao, X. Do we really need deep CNN for plant diseases identification? *Computers and Electronics in Agriculture* **2020**, *178*, 105803. doi:10.1016/j.compag.2020.105803.
147. Agarwal, M.; Gupta, S.K.; Biswas, K. Development of Efficient CNN model for Tomato crop disease identification. *Sustainable Computing: Informatics and Systems* **2020**, *28*, 100407. doi:10.1016/j.suscom.2020.100407.
148. Kamdi, S.Y.; Biradar, V. Monitoring and control system for the detection of crop health in agricultural application through an ensemble based deep learning strategy. *Multimedia Tools and Applications* **2023**. doi:10.1007/s11042-023-17695-7.
149. Bansal, P.; Kumar, R.; Kumar, S. Disease Detection in Apple Leaves Using Deep Convolutional Neural Network. *Agriculture* **2021**, *11*. doi:10.3390/agriculture11070617.
150. Cheng, G.; Xie, X.; Han, J.; Guo, L.; Xia, G.S. Remote Sensing Image Scene Classification Meets Deep Learning: Challenges, Methods, Benchmarks, and Opportunities. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **2020**, *13*, 3735–3756. doi:10.1109/JSTARS.2020.3005403.
151. Yuan, Q.; Shen, H.; Li, T.; Li, Z.; Li, S.; Jiang, Y.; Xu, H.; Tan, W.; Yang, Q.; Wang, J.; Gao, J.; Zhang, L. Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sensing of Environment* **2020**, *241*, 111716. doi:10.1016/j.rse.2020.111716.
152. Qu, H.; Sako, M.; Möller, A.; Dörm, C. SCONE: Supernova Classification with a Convolutional Neural Network. *The Astronomical Journal* **2021**, *162*, 67. doi:10.3847/1538-3881/ac0824.
153. Moradi, E.; Sharifi, A. Assessment of forest cover changes using multi-temporal Landsat observation. *Environment, Development and Sustainability* **2023**, *25*, 1351–1360. doi:10.1007/s10668-021-02097-2.
154. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; others. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence* **2020**, *43*, 3349–3364.

155. Taigman, Y.; Yang, M.; Ranzato, M.; Wolf, L. Deepface: Closing the gap to human-level performance in face verification. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1701–1708.
156. Schroff, F.; Kalenichenko, D.; Philbin, J. Facenet: A unified embedding for face recognition and clustering. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.
157. Mohammed Sahan, J.; Abbas, E.I.; Abood, Z.M. A facial recognition using a combination of a novel one dimension deep CNN and LDA. *Materials Today: Proceedings* **2023**, *80*, 3594–3599. doi:10.1016/j.matpr.2021.07.325.
158. Wang, Q.; Lei, H.; Qian, W. Point CNN:3D Face Recognition with Local Feature Descriptor and Feature Enhancement Mechanism. *Sensors* **2023**, *23*. doi:10.3390/s23187715.
159. Awaji, B.; Senan, E.M.; Olayah, F.; Alshari, E.A.; Alsulami, M.; Abosaq, H.A.; Alqahtani, J.; Janrao, P. Hybrid Techniques of Facial Feature Image Analysis for Early Detection of Autism Spectrum Disorder Based on Combined CNN Features. *Diagnostics* **2023**, *13*. doi:10.3390/diagnostics13182948.
160. Ullah, W.; Ullah, A.; Haq, I.U.; Muhammad, K.; Sajjad, M.; Baik, S.W. CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks. *Multimedia Tools and Applications* **2020**, *80*, 16979–16995. doi:10.1007/s11042-020-09406-3.
161. Rajeshkumar, G.; Braveen, M.; Venkatesh, R.; Josephin Shermila, P.; Ganesh Prabu, B.; Veerasamy, B.; Bharathi, B.; Jeyam, A. Smart office automation via faster R-CNN based face recognition and internet of things. *Measurement: Sensors* **2023**, *27*, 100719. doi:https://doi.org/10.1016/j.measen.2023.100719.
162. Saravanan, T.; Karthiha, K.; Kavinkumar, R.; Gokul, S.; Mishra, J.P. A novel machine learning scheme for face mask detection using pretrained convolutional neural network. *Materials Today: Proceedings* **2022**, *58*, 150–156. International Conference on Artificial Intelligence & Energy Systems, doi:https://doi.org/10.1016/j.matpr.2022.01.165.
163. Chirra, V.R.R.; Uyyala, S.R.; Kolli, V.K.K. Virtual facial expression recognition using deep CNN with ensemble learning. *Journal of Ambient Intelligence and Humanized Computing* **2021**, *12*, 10581–10599. doi:10.1007/s12652-020-02866-3.
164. Febrian, R.; Halim, B.M.; Christina, M.; Ramdhan, D.; Chowanda, A. Facial expression recognition using bidirectional LSTM - CNN. *Procedia Computer Science* **2023**, *216*, 39–47. 7th International Conference on Computer Science and Computational Intelligence 2022, doi:https://doi.org/10.1016/j.procs.2022.12.109.
165. Mehendale, N. Facial emotion recognition using convolutional neural networks (FERC). *SN Applied Sciences* **2020**, *2*. doi:10.1007/s42452-020-2234-1.
166. Debnath, T.; Reza, M.M.; Rahman, A.; Band, S.S.; Alinejad-Rokny, H. Four-layer Convnet to Facial Emotion Recognition With Minimal Epochs and the Significance of Data Diversity **2021**. doi:10.21203/rs.3.rs-511221/v1.
167. Almabdy, S.; Elrefaei, L. Deep Convolutional Neural Network-Based Approaches for Face Recognition. *Applied Sciences* **2019**, *9*. doi:10.3390/app9204397.
168. Bendjillali, R.I.; Beladgham, M.; Merit, K.; Taleb-Ahmed, A. Improved Facial Expression Recognition Based on DWT Feature for Deep CNN. *Electronics* **2019**, *8*. doi:10.3390/electronics8030324.
169. Akhand, M.A.H.; Roy, S.; Siddique, N.; Kamal, M.A.S.; Shimamura, T. Facial Emotion Recognition Using Transfer Learning in the Deep CNN. *Electronics* **2021**, *10*. doi:10.3390/electronics10091036.
170. Li, J.; Jin, K.; Zhou, D.; Kubota, N.; Ju, Z. Attention mechanism-based CNN for facial expression recognition. *Neurocomputing* **2020**, *411*, 340–350. doi:10.1016/j.neucom.2020.06.014.
171. Villegas-Ch, W.; García-Ortiz, J. Toward a Comprehensive Framework for Ensuring Security and Privacy in Artificial Intelligence. *Electronics* **2023**, *12*. doi:10.3390/electronics12183786.
172. Ortiz-Clavijo, L.F.; Gallego-Duque, C.J.; David-Diaz, J.C.; Ortiz-Zamora, A.F. Implications of Emotion Recognition Technologies: Balancing Privacy and Public Safety. *IEEE Technology and Society Magazine* **2023**, *42*, 69–75. doi:10.1109/MTS.2023.3306530.
173. Lu, Z.; Jiang, X.; Kot, A. Enhance deep learning performance in face recognition. 2017 2nd International Conference on Image, Vision and Computing (ICIVC), 2017, pp. 244–248. doi:10.1109/ICIVC.2017.7984554.
174. Vu, H.N.; Nguyen, M.H.; Pham, C. Masked face recognition with convolutional neural networks and local binary patterns. *Applied Intelligence* **2022**, *52*, 5497–5512. doi:10.1007/s10489-021-02728-1.

175. Singh, R.; Saurav, S.; Kumar, T.; Saini, R.; Vohra, A.; Singh, S. Facial expression recognition in videos using hybrid CNN & ConvLSTM. *International Journal of Information Technology* **2023**, *15*, 1819–1830. doi:10.1007/s41870-023-01183-0.
176. Dang, T.V. Smart Attendance System based on improved Facial Recognition. *Journal of Robotics and Control (JRC)* **2023**, *4*, 46–53. doi:10.18196/jrc.v4i1.16808.
177. Hangaragi, S.; Singh, T.; N, N. Face Detection and Recognition Using Face Mesh and Deep Neural Network. *Procedia Computer Science* **2023**, *218*, 741–749. International Conference on Machine Learning and Data Engineering, doi:https://doi.org/10.1016/j.procs.2023.01.054.
178. Shin, H.C.; Roth, H.R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D.; Summers, R.M. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging* **2016**, *35*, 1285–1298.
179. Plastiras, G.; Terzi, M.; Kyrkou, C.; Theocharides, T. Edge intelligence: Challenges and opportunities of near-sensor machine learning applications. 2018 IEEE 29th international conference on application-specific systems, architectures and processors (asap). IEEE, 2018, pp. 1–7.
180. Azad, R.; Fayjie, A.R.; Kauffmann, C.; Ben Ayed, I.; Pedersoli, M.; Dolz, J. On the texture bias for few-shot cnn segmentation. Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2021, pp. 2674–2683.
181. Ioannou, S.; Chockler, H.; Hammers, A.; King, A.P.; Initiative, A.D.N. A study of demographic bias in CNN-based brain MR segmentation. International Workshop on Machine Learning in Clinical Neuroimaging. Springer, 2022, pp. 13–22.
182. Tetard, M.; Carlsson, V.; Meunier, M.; Danelian, T. Merging databases for CNN image recognition, increasing bias or improving results? *Marine Micropaleontology* **2023**, *185*, 102296.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.