[1]Pradumn Kumar

[2] Praveen Kumar Shukla

Deep Learning in Computer Vision: A Critical Review

Journal of Electrical Systems

Abstract: - Today, a machine learning tool has gained a lot of consideration as a data analysis and image processing tool, with encouraging results. Massive amounts of data are being generated in every field or any domain at a distressing rate. This review paper provides a brief overview of current technologies and a conceptual explanation of the development of computer vision with image processing and the usage of specific regions in their respective fields. Computer vision with deep learning allows researchers to investigate pictures and videos to attain essential facts, recognize facts on occasions, or provide explanations and exquisite patterns. It utilized the approach of diverse-variety utility areas with large-scale data mining. We employed a review to extract and synthesize the techniques and attributes that have been used in different applications of computer vision, like image classification, image localization, object detection, and many more, with their features. So according to computer vision tasks, there are Deep Belief Networks, Convolutional Neural Networks (CNN), Deep Boltzmann Machines (DBN), Recurrent Neural Networks (RNN), Stacked Denoising Auto Encoders, and Long-Short Term Memory (LSTM) algorithms that are some of the most important advanced machine learning (deep learning) algorithms. Lastly, a quick evaluation of future approaches to constructing advanced machine learning algorithms is provided for challenges and issues related to computer vision with improved performance.

Keywords: Computer vision (CV), Convolutional Neural Network (CNN), Long-Short term Memory(LSTM), Recurrent Neural networks(RNN), Deep Boltzmann Machines(DBMs), Deep Belief Networks(DBNs).

I. INTRODUCTION

Computer vision has emerged as an important area of study, with tasks such as gathering data, sampling images, and interpreting data [1]. This field of study combines principles, approaches, and goals from computer graphics, pattern recognition, digital images, and artificial intelligence [2]. The main objectives of computer vision entail gathering data on actions or descriptions, such as capturing scenes and extracting objects. The selection of approaches for problem-solving in computer vision is contingent upon the specific operating domain and the inherent properties of the information under analysis. Computer vision functions in a manner comparable to human vision, although humans have an advantage in this sector due to their prior knowledge and experience. The human visual system is enhanced by a vast amount of contextual information gathered during one's life, allowing it to differentiate between various objects, calculate their distance, detect movement, and identify any abnormalities in an image. Computer vision allows computers to carry out these tasks by utilizing cameras, statistical analysis, and algorithms, rather than depending on retinas, optic nerves, and a visual cortex. In addition, computer vision achieves these jobs with greater efficiency and speed. An AI system that is trained to inspect items or monitor a product asset possesses the capability to analyze multiple products or perform a specific procedure.

A millisecond, with its exceptional speed, has the ability to identify even the most minute defects or issues that may not be readily apparent. Computer vision is the combination of image processing and pattern recognition [3]. Scientists and engineers have been working for more than sixty years to improve methods for robots to understand and recognize visual information. In 1959, neurophysiologists initiated an experiment to confirm the activation of a cat's brain through the presentation of different visual stimuli. The objective was to establish a connection between the stimuli and the cat's brain response. It was observed that the picture processing initially detects and reacts to coarse edges or lines, suggesting that it starts to evolve by recognizing basic structures like straight edges [4]. Simultaneously, computerized photo scanning technology was created, enabling computer systems to convert photographs into digital format and store them. In 1963, a notable milestone was reached when computer systems

[1] Research Scholar, Department of Computer Science Engineering, Babu Banarasi Das University Lucknow,226028, Uttar Pradesh, India

Email- pradumnyadav18@gmail.com

[2] Professor and Dean, Department of Computer Science Engineering, Babu Banarasi Das University Lucknow,226028, Uttar Pradesh, India

Email- drpraveenumarshukkla@gmail.com;

acquired the capability to convert two-dimensional images into three-dimensional images. In the 1960s, AI emerged as a scholarly area of study, and it also signaled the commencement of the AI effort to address the challenge of human vision. Optical Character Recognition (OCR) technology was initially introduced in 1974 with the purpose of accurately deciphering printed text, irrespective of the font or typeface employed [5]. Moreover, the application of neural networks is essential for the analysis of handwritten text, namely by including perceptual awareness and intelligent character recognition (ICR) [6]. OCR (Optical Character Recognition) and ICR (Intelligent Character Recognition) are now extensively utilized in diverse everyday applications, including bill and report processing, license plate recognition, online payments, and computer translation.

The main goal of computer vision is to create models, extract information, and generate instructions from visual data. Conversely, the main goal of image processing is to utilize deep learning techniques to modify visual data, such as improving clarity and identifying differences, among other tasks [7]. Moreover, it holds a similar level of importance and rarely surpasses human-computer interaction (HCI) [8]. HCI insurance focuses mostly on the holistic design, interface, and various elements of applied sciences related to the interaction between humans and computers. Human-Computer Interaction (HCI) is a multidisciplinary field that examines the interactions between humans and computers. It entails utilizing scientific breakthroughs to comprehend the mechanisms by which humans and computers interact and exchange information. Multi-dimensional data is categorized as spatial data and can be comprehended functionally by both computer vision and human vision. Computer vision cannot replicate the capabilities of the human eye due to certain limitations that affect its effectiveness in various aspects [9][10]. The complexity of the variables, the effectiveness of the algorithms, and the precision of the findings are some of the main problems in their approach. It affects the degree of intricacy in evaluating the effectiveness of computer vision systems. For an algorithm to attain accuracy, robustness, and scalability in managing and monitoring system performance, it is necessary to evaluate certain fundamental qualities of the algorithm during the performance evaluation phase. Several scholars have proposed substantial efforts to broaden and categorize computer vision into various domains and specific applications, such as assisting visually impaired individuals, automating manufacturing processes, remote sensing [11], robotics [10], and computer-mediated human communication [12]. The effectiveness of computer vision systems depends on the architecture of the application system.

The main goal of artificial neural networks (ANNs), a widely used machine learning method, was to imitate the adaptable characteristics of biological nerve systems using software and specialized hardware [13]. It is undeniable that complex artificial neural networks may learn intricate, quasi-functional relationships when given enough processing power and training data. Importantly, their results grow in direct proportion to the quantity of training data, in contrast to conventional approaches. There has been substantial advancement in pattern recognition throughout the intellectual community, resulting in improved capacities for both academic and industrial research.

II. Deep Learning Involvement in the Evolution of Computer Vision Applications

An Artificial neuronal Network (ANN) is a computational model that takes inspiration from the neuronal structure of the human brain. The structure is composed of interconnected nodes, also known as artificial neurons, which are arranged in layers. Data is transmitted across these nodes, and the network adapts the connection intensities (weights) [14] through training in order to acquire knowledge from the data, allowing it to identify patterns, make forecasts, and accomplish diverse tasks in the fields of machine learning and artificial intelligence. The structure can vary from a solitary layer to numerous layers of interconnected nodes (neurons) arranged in a hierarchical manner. In order to train the model, there are several steps that demonstrate how the weights are actually learned. These procedures are outlined as follows [15].

- Assigning initial weights to all nodes.

- During each training job, the forward pass is executed using the current weights to determine the output for each individual node. Horizontally, in the direction from the left side to the right side. The terminal node represents the final output value [16].
- The error is calculated using a loss function after comparing the actual value with the expected value [17].
- Utilize the Backward pass algorithm to transmit the error for each individual node. During the backward pass of backpropagation [18], the weights are adjusted using gradient descent based on the mistake. Then, the

forward pass is used again to obtain the desired results [19]. This process will persist until the desired outcomes are not achieved.

In the second half of the last century, multi-layer neural networks advanced. Why have scholars and corporations focused on deep neural networks in recent years is a valid issue [20]. Many factors contributed to the significant increase in research spending and productivity. Several include briefings:

- Large training data sets with correct labeling are more available [21].
- Parallel computing and multi-core, multi-threaded systems have improved.
- Derivatives and environment configuration are simplified by specialized software platforms. These platforms simplify GPU-based framework integration and deployment. Table 1 lists frameworks.

Table 1 Comparison of Deployment Frameworks for Machine Learning and Deep Learning

Framework	Objectives
Theano [22]	A GPU-based Python library with close NumPy integration, used for performing mathematical operations on multidimensional arrays.
Microsoft Cognitive Toolkit [23]	A Deep Learning Framework (CNTK) that uses directed graphs to describe computations.
TensorFlow [24]	A library used for machine learning and deep learning. TensorFlow creates dataflow models which help to solve numerical computations.
Keras [25]	Used with TensorFlow to reduce cognitive load.
PyTorch [26]	Accelerates the process from prototype to production.
Caffe [27]	A deep learning framework that prioritizes modularity, performance, and expression.
Chainer [28]	Offers various network designs including feed-forward nets, convnets, recurrent nets, and recursive nets.
Scikit-learn [29]	Helps with classification, regression, and clustering tasks.

- Better regularization methods have been developed to prevent overfitting [30] while scaling up. Batch normalization, dropout, data augmentation, and early halting reduce overfitting and improve model performance as we scale.
- By using adaptive learning rates, [31] rigorous optimization yields the best solutions.

Deep learning (DL) allows computational models with several processing layers to learn and represent data at many abstraction levels, replicating how the brain processes multimodal knowledge [32] and implicitly storing complicated huge data structures [33]. Deep learning includes neural networks, multilevel probabilistic models, and other unsupervised and supervised methods\cite [34]. Deep learning approaches have gained popularity due to their superior performance on several tasks compared to state-of-the-art methods [35] and the volume of complicated data from visual, aural, life sciences, social, and sensory sources [36]. Deep Learning uses hierarchical designs to learn somewhat higher ideas from data in Advance Machine Learning. This novel technique has been widely applied in well-established artificial intelligence disciplines including text classification [37], transfer learning [38], natural language processing [39], computer vision [40], and others. Deep learning is developing due to three main factors: increased chip processing (e.g. GPU units), cheaper computer hardware, and better machine learning methodologies [41]. V/SP and NLP advances are examined in this analysis [42]. DL development milestones in various application sectors were highlighted in the survey. Their research showed that DL affects NLP and V/SP. This study examined pattern-matching natural conversation systems' history, development, and use [43].

This study proposes AudVowelConsNet, a V/SP-based speech-based clinical depression evaluation and recognition tool. As DL improves, researchers are investigating commercial applications [44]. A hybrid wind speed forecasting (WSF) model using long short-term memory (LSTM) networks and atomization approaches, including grey wolf optimization, is created for efficient wind power (GWO) use [45]. Computer memory, CPU, and GPU limits make DL difficult in early CV development. Thus, most academics are studying CV ML. Many CV computing methods have been developed, including Expectation-Maximization (EM), K-means clustering, Bayesian Learning, Machine (SVM), Random Forest, K-Nearest Neighbor (KNN), Decision Tree Algorithm, Haar

Classifier, and Support Vector Boosting. Over the last few decades, the DL has evolved rapidly, and its algorithm and design may be split into 10 categories: CNNs, RNNs, MLPs, GANs, Self-Organizing Maps, Restricted Boltzmann Machines, Radial Basis Function Networks, LSTMs, DBNs, and Autoencoders. Comparing literature and findings on CV tasks such semantic segmentation, human posture estimate, object identification in pictures, and image retrieval [46]. CNN was the best CV design after comparing CNN, RBM, Autoencoder, and Sparse Coding. However, model sizes and precisions limited applicability, causing several issues. Lack of information about architectures, comparison for greater performance, sparse dataset training, difficulties implementing real-time applications, and need for stronger models were some.

CNN's 2012–2018 organisation, technology, hardware, and backbones [47]. They prioritized CNN growth. In contrast, our research will advance DL CV and offer a timeline-style perspective. CNN is the most prominent DL technique after its ImageNet performance [48].Image classification is essential for CV applications. CNN has convolutional, pooling, and fully connected layers. A CNN convolves the intermediate feature maps with the full picture using different kernels in the convolution layer to produce distinct feature maps. Pooling layer reduces feature map size. This CNN classifier is usually at the end of each CNN architecture for the completely connected layers. The output from fully linked layers is utilized for image classification or transmitted to another Deep Neural Network. Recent deep learning advances are described. Deep learning has four main applications: detection, semantic segmentation, picture restoration, and visual tracking. The research intended to discover contemporary CV application cases and upcoming DL techniques.

In the recent decade, CNN has dominated computer vision. Following AlexNet's image classification breakthrough, CNN-based network models have emerged. Current deep learning models and their development include:

A. AlexNet: Int this study AlexNet with five convolutional and three connected layers [39]. The rectified linear unit (ReLU) activates the convolutional layers' output after each layer. The original AlexNet model trains on two GPUs. Caffenet and CNN's Alexnet-like design. Caffenet pools first, then normalizes local response in the first two convolutional layers, whereas Alexnet reverses.

B. VGGNet: Developed by Simonyan and Zisserman (2015) [49], VGGNet employs (3X3) convolution filters to deepen the network while keeping stable parameters. Increasing the depth to 16 and 19 weight layers might improve VGG-16 and VGG19. Despite VGG-16 and VGG-19, VGGNet has a large parameter space, making it better than other techniques from the same age, supporting the premise that increasing network depth may affect network performance. VGGNet has over 500 M in its final model, while AlexNet has 200 M. Thus, VGG model training takes longer than AlexNet model training.

C. GoogleNet: Convolutional neural networks like GoogLeNet use the Inception architecture. Inception modules let the network choose convolutional filter sizes in each block. These modules are layered using an Inception network, and maxpooling layers sometimes cut the grid's resolution in half using stride sizes of 2 and 3. A stack of mlpconv layers called Network in-Network (NIN) was introduced by Google researchers [50]. Convolution filters are replaced with nonlinear system function approximators. Another NIN feature is worldwide mean pooling to replace completely linked layers. The softmax layer receives the feature map vector and average. Studies utilizing multiple photo datasets showed that NIN's classification accuracy may be comparable or better with fewer parameters. A new CNN architecture called Inception v1 was proposed by Szegedy et al. (2015).[51] Increasing architectural size for performance is safe. Increasing the number of parameters and computing resource consumption might cause bottlenecks, they said. Inception, or CNN architecture tiers, addressed these challenges. Using the same computing resource, it may increase network depth and width. GoogLeNet, a 22-layer deep model, was created by repeating inception layers. GoogleNet uses NIN principles 1X1 Convolution and global mean pooling. Later study offered ideas to improve Inception v1's basic design [52]. It shows that massive convolution filters often have substantial processing costs. It suggests two stacked 3X3 filters instead of 5X5 (7X7). This style is Inception v2. The authors also explored the batch normalisation (BN) auxiliary, which normalised the output of normal distribution N (0,1) inside each mini-batch of data to reduce internal neuron distribution changes. They name this layout Inception v3. Szegedy et al. (2017) [53] reduced Inception v3 into Inception v4, influenced by ResNet. They combined the Inception model with back propagation to develop Inception-ResNet. Finally, depthwise separable convolutions replace Inception modules in current research to improve Inception v3 performance [54]. Xception narrowly beats Inception v3 on ImageNet but dominates JFT.

D. ResNet: Learning a residual function that considers a layer's input is more effective than learning its parameters without considering it. They built ResNet, a 152-layer residual network eight times deeper than VGG Nets. CNN networks (e.g., AlexNet, VGG) explicitly train mappings between input and output, whereas the residual network employs multiple variable layers to learn the representation of residuals between input and output. As direct connections expand, the vanishing gradient issue is pushed, feature propagation is increased, feature reuse is encouraged, and system parameters are greatly reduced [55]. Researchers created DenseNet, which combines all layers directly, because convolutional networks are faster and more exact. The DenseNet uses picture characteristics from all preceding layers to create L(L+1)/2 connections instead of L connections in typical convolution networks [56]. Thus, it gives four advantages: The disappearance issue was solved, feature propagation was enhanced, feature reuse was promoted, and parameter requirements were significantly reduced.

E. Regnet: RegNet and its variants excel in computer vision tasks. The shortcut connection technique facilitates gradient movement among construction pieces, but its additive function limits the ability to regularly investigate new possible complementing qualities. In this work, we propose using a regulator module to store complementary characteristics for the ResNet to solve this problem. Convolutional RNNs like LSTMs or GRUs are employed in the regulator module because they can retrieve spatiotemporal information. Our new, controlled networks are called RegNet. Regulation module implementation is simple in any ResNet architecture [57].

III. Comprehensive Analysis of Advanced Machine Learning Architectures for Visual Understanding

Thanks to recent developments in deep learning, computer vision has been utterly transformed, with computers now capable of performing a vast array of complicated visual tasks with unparalleled precision and productivity.

A. Image Categorization

Image Image categorization is a fundamental topic in computer vision and pattern recognition. Image classification is used in biometric identification, human computer interaction, visual data collecting, video surveillance, and online content evaluation [58][59][60][61][62]. In recent years, feature coding, a fundamental part of picture categorization, has garnered interest among numerous coding methodologies. Picture classification frequently starts with extraction of features and continues with categorization. Visual data collection and analysis are essential for picture categorization. Traditional classification uses minimal or intermediate attributes to draw a picture. Human-defined color, texture, shape, location, and grayscale density are utilized to build low-level features. Also called hand-crafted features. Many feature learning and mid-level trait extraction techniques use bag-of-visual-word (BoVW) algorithms [63]. Recently, these algorithms have become increasingly prominent due to their photo retrieval and classification performance. Computer vision uses a classifier (SVM, etc.) to give tag names to object classes after collecting attributes [64]. They utilized three methods to visualize the findings in pixels: To accurately put upper-layer reconstructions,

1) Unpooling: involves noting the maximum locations within each pooling zone.
2) Rectification: After using ReLU non-linearity rebuild the signal.
3) Filtering: Convolutes the image features from the preceding layer using learned filters.

The projections of the trained deconvnet layers show hierarchical Alexnet features. AlexNet's first layer's stride step and receptive window size were reduced to enhance picture classification. LRP improved DNN's transparency and helped users understand the classifier scientifically [65]. For instance, a powerful tool for constructing and outline recognition features fails to detect drawings from photographs without dark outline colors. A fine-tuned DCNN has a far better accuracy rate (96.8%) than previous models, including the first CNN models developed from scratch [66]. A dispositive deep belief network (DisDBN) learns trustworthy and discriminative features to classify high-resolution artificial aperture radar images. The author runs three experiments to establish DisDBN works after training weak classifiers and collecting enhanced differentiating characteristics [67].

B. Identification of Objects

In contrast to classification algorithms, object identification techniques pinpoint the precise location of objects or structures inside the image by displaying their bounding boxes. Localization and similar tasks include detection: While object identification techniques are capable to locate existence and the place of different objects that seem to

be available in image, localization techniques typically only recognize one features of an image. In order to do this, the recognition algorithm will produce a boundary for each objects in an image and, connected with each boundary, the kind of object that it includes (here points indicates for the probability of object to related category) [67]. In computer vision, object identification techniques typically use the following two steps: (1) Patch suggestion, which is the process of removing various patches from a picture in search of possible regions that might contain the subject of interest. With a feature extraction technique or specialized patch suggestion algorithms, the entire image can be examined and segmented into patches to identify the area's most likely to include particular objects. (2) Classifying the collected alterations to create bounding boxes that have a specific likelihood of including an object. There are two approaches which we use in object recognition (One Stage Approach and Two Stage Approach)- The two-step method produces a minimal set of the boundary from the image in the initial stage. The final recognition results are then improved by corrections depending on the boundary region. The single-stage method, in contrast, directly analyzes the image and produces detection results. Although the single-stage recognition is quicker, it has a lesser level of recognition precision. The two-stage strategy, however, does the exact opposite.

In computer vision, object identification techniques typically use the following two steps:

1) Patch suggestion, which is the process of removing various patches from a picture in search of possible regions that might contain the subject of interest. With a feature extraction technique or specialized patch suggestion algorithms, the entire image can be examined and segmented into patches to identify the area's most likely to include particular objects.

2) Classifying the collected alterations to create bounding boxes that have a specific likelihood of including an object.

There are two approaches which we use in object recognition (One Stage Approach and Two Stage Approach)- The two-step method produces a minimal set of the boundary from the image in the initial stage. The final recognition results are then improved by corrections depending on the boundary region. The single-stage method, in contrast, directly analyzes the image and produces detection results. Although the single-stage recognition is quicker, it has a lesser level of recognition precision. The two-stage strategy, however, does the exact opposite.

1) One -Stage -Approach

- *You Only Look Once (YOLO)*

Among the most well-liked object recognition techniques and model designs is called YOLO. The primary reason for its popularity is that it makes use of one of the greatest architectures of neural networks to create highly accurate and overall faster. The YOLO model will be mentioned in the first result of a Google search for object detecting algorithms. The YOLO algorithm seeks to forecast both the category of an object as well as the grid cell defining its position here on source images [68]. Following components are used to identify each bounding box: Center of Grid cell, Height of Grid Cell and Width of Grid Cell. Additionally, Prediction of YOLO represents predicted class number with its correspondence and prediction probability. The researchers note that YOLO can produce 45 frames per second, and the faster version is more effective. In other words, as compared to comparable real-time systems, 155 pixels per second doubles Precision's average mean. Be aware that YOLO's accuracy still trails that of vital detection systems. There are various version of YOLO models with main improvements from version 1 to version 5[69].

- V1: Detection and confidence loss are handled by the grid division.
- V2: Fully Connected Convolutional Network, Two-stage learning approach, and attach with K-means.
- V3: FPN-based multiscale detection.
- V4: Knowledge enhancement techniques, activation function(MISH), mosaic, and GIOU error function (Generalized Intersection over Union).
- V5: Adaptable model size control, use of the activation function (Hardswish), and data improvement.

- *Single Shot Detector (SSD)*

Although SSD is indeed an object recognition technique, what does that mean? Many individuals conflate image categorization and object detection. Simply said, image classification identifies the type of image, whereas object detection identifies the various objects in the image and pinpoints their locations using bounding boxes. Let's move on to SSD now that that is resolved. Since the SSD employs a single-shot multi-box detection method, which is a quicker and more effective technique than the YOLO algorithm. Unlike other models that traverse the frame

multiple times in order to obtain an output detection, the SSD model identifies the item in a quick pass well over source images. However, the SSD variant also appears to have astounding detection accuracy at the same time. The SSD model creates predictions at many scales from extracted features of various scales and explicitly divides forecasts by aspect ratio in order to obtain high detection performance. A technique that does away with the need to generate bounding boxes. Six feature maps are initially processed using their method. The length of the frames on the input signal is generated differently for each anchor box on each feature map. As a result, it may function image features from multiple resolutions to handle objects of varying sizes. When the element size is 300X300, the detecting speed can reach 59 FPS. On the VOC 2007 dataset, adjusting the image size to 512 X 512 results in 76.9% mAP, outperforming the crucial detection approach, a quicker R-CNN [70]. The accuracy rate decreases between 77.5% to 76.4% when try to switch the basis network from VGG to Residual101 [71] based on the SSD. They included a forecasting module for enhancing the networks throughout each task in order to increase accuracy, drawing inspiration from MS-CNN [72]. Deconvolutional Single Shot Detector (DSSD) [73] can effectively guide small objects in the image, even though the final efficiency is close to SSD513, which network is also Residual-101.

- *RetinaNet*

To imbalanced data the root reason of the one-stage- approach's was poor accuracy, and they proposed a new structure called RetinaNet that makes use of focal loss. The backbone of RetinaNet was composed of ResNet and the Feature Pyramid Network (FPN). In order to apply a regulating term to the cross-entropy loss, it used specific single recognition with focused loss. This serves to downplay the many obvious downsides and concentrate learning on difficult situations. This structure outperforms Faster R-CNN on FPN [74], which obtained 36.2 mAP based on the difficult COCO datasets, by 39.1 mAP.

2) Two -Stage -Approach

The two-stage models are the foundation of one subset of object detectors. For extraction of object's first part of the model is used and for classification and localization of a object another part of model is used. Both models are based on the work of R-CNN. Although sharing features have enhanced 2 stage detectors, they now have a similar computation complexity as single-stage identifier. These techniques are known to be moderately slow but quite powerful. These works largely build on the prior process as a foundation and are heavily reliant on earlier developments. Understanding all of the primary algorithms used by two-stage detectors is crucial.

- *R-CNN*

The basics of CNN-based two-stage detector is proposed in the 2014 study [75], it is later enhanced and speeded up in the subsequent studies. The pipeline as a whole is divided into three steps, which are shown in below diagram:

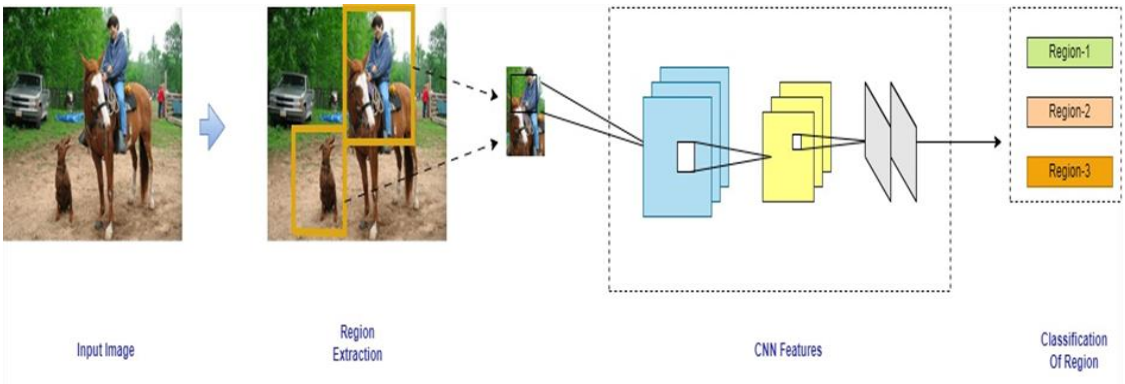

Fig. 1 Architecture of R-CNN

Produce region suggestions: The model needs to depict candidates for the objects in the image, regardless of their category. A fully convolutional network computes parameters from every candidate region in the second stage. A fully linked layer, represented in the study as SVMs, is the last stage. There are several ways to produce region proposals, but the publication opts to employ a selective search to compare it to earlier research. Together, second and third phases can be thought of as a standard CNN that works with cropped region proposals. The region

suggestion is resized by the classification network, which also forecasts class probabilities (includes background) and boundary box improvements.

- *Spatial Pyramid Pooling (SPP-Net)*

The usage of SPP layers, which can operate with photos of any size without having to scale them to fit, is suggested in the research. Resizing photographs to fit a fixed size can cause lost knowledge and distorted images. Convolutions, which are referred to as feature extraction in the CNN, are not the ones limiting the fixed input size; instead, the fully linked classification layers are to blame.

To overcome the network's fixed-size constraint, the authors suggest an unique pooling layer that translates features of various dimensions then delivers them to the fully linked layers, [76]. In essence, the SPP layer combines max-pooling to the result at different scales, according to the size of the image. Any form image can be mapped it into single size using the SPP layer's spatial bins, which have sizes proportionate to the image size. By max-pooling the values in each spatial bin's area, matrix can be maintained. SPP-Net include properties like regardless of the size of the input, produces a fixed-length output, well-known for being resistant to item deformations, can retrieve data from a variety of scales.

- *Fast R-CNN and Faster R-CNN*

Girshick (2015) created the fast R-CNN, a network that can concurrently categorise objects and forecast the placement of bounding boxes. First, it used the selective search technique to provide a set of classifiers for an image. It then used the RoI pooling layer to extract feature information from each region. A softmax classifier and a boundary box regression model, respectively, were fed with the retrieved features and the coordinates of the matching bounding box. This approach, which merges the final two modules in comparison to R-CNN, speeds up training and testing for the S group by 169 times [77]. Using many patches from one image to train efficiently requires only one forward/backward run through the convolution layers. Backpropagation facilitated the learning of convolutional feature extraction.

To reach the objective of real-time object detection, their approach attempted to eliminate the difficulty in producing object suggestions. The Region Proposal Network (RPN) was developed for forecasting object limits and Objectivity ratings. With this technique, each step is integrated into a single neural network. They refer to their approach as a quicker R-CNN as a result\cite [78]. The analysis is not spread throughout the entire page in such case, adding extra work and time. R-FCN, a deep convolution region-based detection with a stance score map, to address this problem. The ROI pooling location across Faster R-CNN layer can have an impact on translation invariance and cause R-FCN to obtain comparable accuracy in 19 times less time.

- *Feature Pyramid Networks (FPN) / Mask R-CNN*

Scale invariance is supported by FPN, which offer multi-scale feature representations useful for object detection [79]. Adjusting the layers of a pyramid can easily compensate for the scale variation of the items, and the model should be able to recognise all dimensions of the image's objects. But because it obviously takes longer to compute the features of numerous levels, pipelines like Fast/Faster R-CNN do not employ it. A structure that uses lateral connections and a top-down approach to integrate high-resolution, semantically weak characteristics with low-resolution, conceptually powerful features. A universal method for creating multi-scale extracted features with rich semantic information is offered by the FPN pipeline. The FPN architecture is used in the Faster R-CNN region-based classification backbone as well as the RPN network for constructing boundary box recommendations when it comes to the object identification pipeline. By substituting the network structure and supplying the FPN output rather than a single feature map, FPN is adapted to RPN.

It is suggested to use Mask R-CNN to address a somewhat different instance segmentation problem. In a nutshell, this issue combines semantic segmentation and object identification. The job is to produce pixel-wise boundaries between objects. The Faster R-CNN pipeline serves as the foundation for Mask R-CNN [80], which represent three outcomes for each object suggestion as opposed to two. The extra branch forecasts K binary object masks that divide up each type of object in the image. The classification branch's outcome is used to choose the final instance segmentation map that has to be drawn. This is known as decoupling class prediction from the mask [81].

C. Track the Visuals

Visual tracking is a crucial topic in computer vision, having applications in surveillance systems, self-driving cars, and human-computer interface robotics. This term refers to the automated prediction of a target item’s trajectory in

future video frames. Initial frames show the target item as a grid cell. Visual tracking involves detecting, recognizing, and extracting objects from video sequences. After identifying an item in a video clip, the tracker will automatically display its position and attribute in subsequent sequences or offer a prompt if the object is not visible [82]. Visual tracking differs greatly from traditional technologies like radar or satellite tracking. Visual tracking relies on computer vision (CV) and image data. Visual tracking involves image analysis, pattern categorization, intelligent systems, and automatic control. It has potential uses in intelligent transportation, medical diagnostics, video encoding, security monitoring, and military direction. The visual tracking methodology encompasses object location, extraction, identification, and organizational structure for data collection and decision-making. All tracking processes are interdependent and restricted. Effective object identification algorithms, tracking efficiency, and durability depend on object separation and expression, requiring a system-level approach. A detection algorithm identifies two types of visual tracking methods: generative and discriminative [83]. The generative technique, like PCA, reduces inaccuracy in object search and characterizes apparent qualities using a generative model. The discriminative approach, with better accuracy and ability to distinguish items from their backdrop, is becoming the dominant tracking methodology. Tracking-by-Detection refers to discriminative approaches, which includes deep learning. One of the most challenging CV topics is visual tracking. Visual tracking is affected by environmental elements such posture changes, lighting, noise, and obstacles in clips. Researchers see multi-idea approaches more. Classification of single-modal and multi-modal tracking approaches using multicue[84]. Kumar et al. (2020) distinguished standard multi-cue object tracking from DL approach and architecture [85]. Addressing object tracking as a learning feature representation challenge. They recommended using multi-layer perceptron autoencoders trained remotely on additional picture data to develop a broad feature representation of images [86]. The encoder is linked to a classification layer for online tracking. Adjustments are made to the encoder and classifier layer to adapt to changes in object appearance. This technology is called deep learning tracker (DLT). Wang et al. (2015) found that generic characteristics cannot reflect temporal invariance, preventing DLT transfer from offline to online learning. A two-layered CNN trained on offline video sequences is proposed to overcome both issues. An online adaptation module applies learned characteristics to a target video sequence [87]. Zhang et al. (2016) proposed a CNN-based online training technique using a lightweight convolutional network topology to solve generic characteristic discrimination and training time complexity. Both basic and complicated layers comprise its structure. The difficult layer addresses position ambiguity, whereas the basic layer includes static filters from the target and adjacent regions. The convolutional network-based tracker (CNT) method achieves an
AUC of 0.545, surpassing the DLT approach by 10.9 points [88]. Qi et al. (2016) proposed a method to enhance trackers by hedging poor trackers from many layers of a pre-trained CNN. They argued that a single layer's characteristics cannot fully utilize CNN's capabilities [89]. The proposed hedged deep detection approach was shown effective by utilizing an online decision-theoretic hedge algorithm to evaluate weak trackers (refer to Fig. 5). Currently, many trackers employ deep neural networks, but Yun et al. (2017) created a tracker with little computation and satisfactory accuracy. Tracking the goal involves repeated actions guided by the well-trained action-decision network (ADNet) using supervised and reinforcement approaches. The tracking system will become more resistant to deformation as it undergoes online modifications. The video retrieval experiment showed that ADNet, MDNet, and C-COT had equal accuracy and success rates (64.6% AUC, Area Under Curve) [90]. Song et al. (2018) have identified two drawbacks of deep categorizing network trackers: (a) positive samples overlap significantly, and (b) positive and negative samples are very imbalanced. Adversarial learning, utilizing the VITAL algorithm, was recommended to address the issue. They employ generative networks to produce masks
randomly and choose the most resilient ones to overcome the first issue. Consider a large financial loss to address the second issue and mitigate negative effects [91]. High accuracy ranking (AR) and resilience rank (RR) of 1.63 and 2.17, respectively, result in an estimated mean overlap (EMO) of 0.323 for VITAL. Lukezic et al. (2020) propose a differentiated single-shot segmentation tracker called D3S. They use specialized networks (GIM and GEM) to develop two modules for segmentation and location solutions. Total AUC climbs 72.8%. The poor depiction of the target in the Bounding Box can affect video segmentation performance, background noise, and long-term resilience [92].

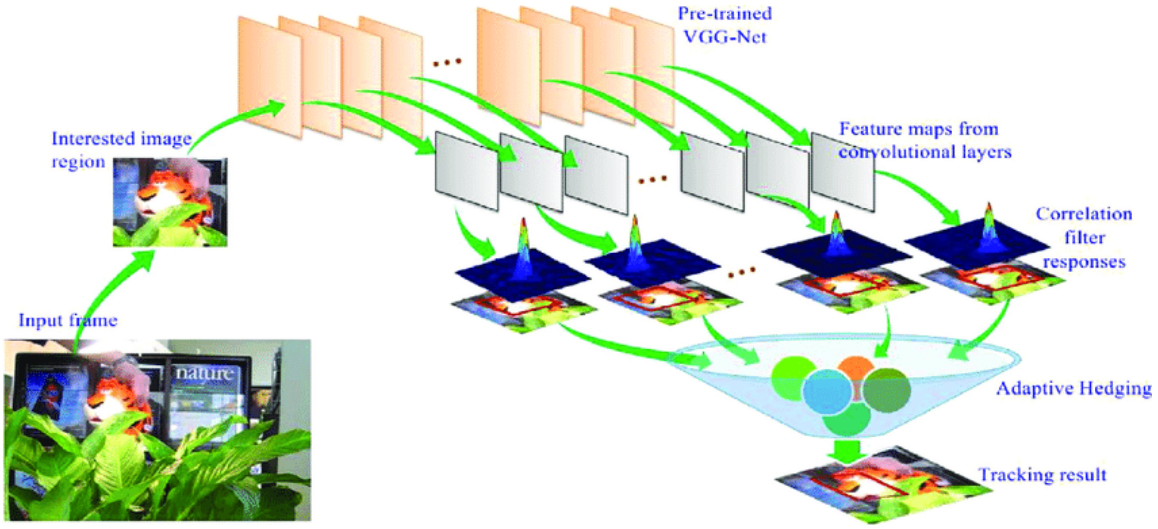

Fig. 2 Processes for tracking results by using DL backbone [89]

D. Semantic Segmentation

CNN is still the most used DL semantic segmentation technique. Long et al. (2015) recommended CNN for end-to-end dense learning. They replace the CNN's last fully connected layer with a 1X1 convolutional layer to create a heatmap [93]. A deconvolutional network was suggested by Noh et al. (2015) for semantic segmentation. The network has a convolution network and a deconvolution network with several deconvolutions and up pooling layers. Deconvolution is the reverse of convolution, while up pooling is the inverse. This network creates a probability map of pixel classifications. Network design was VGG16 [94]. Hong et al. (2015) distinguished semantic segmentation tasks into classification and segmentation. Different CNNs trained for each task. Bridging layers moved class-specific data from classified to segmented networks. Learning class-specific activation maps improves segmentation performance and uses image- and pixel-level class labels. Before training classification networks, they utilize several pictures. They then fix the classifier's parameters and train bridge layers and segmentation networks using a small amount of highly annotated training data. Their DecoupledNet method used VGG16[95]. Pinheiro et al. (2016) [96] developed SharpMask to address the issue of feedforward network upper layers being factor-invariant. It enhanced feedforward networks and joined DeepMask through top-down refining. The experiments showed that the method outperformed the original DeepMask network in both quality (10-20% recall accuracy improvements) and speed (50% faster, say below 0.8 s per image), making it suitable for other pixel-labeling tasks.Using faster R-CNN for instance segmentation, He et al. (2017) improved object detection. Mask R-CNN provides a binary map that displays the bounding box and whether a pixel is an object [97]. On the COCO dataset, the Realigning RoI Pool (RoiAlign) method obtains 37.1AP and 60AP50. Chen et al. (2018) presented DeepLabv3+ to integrate deep neural networks' tighter object boundary capture with encoder-decoder structure's multi-resolution contextual information encoding. They also learned from the Exception model to create a faster and more powerful encoder-decoder network [98]. The technique proved effective with an 89% test set performance in PASCAL VOC 2012. Zhang et al. (2019) claim that Canet, a class-independent segmentation network with few-shot learning, achieves 49.9% mean intersection over union (IOU) on COCO2014 dataset. The suggested k-shot issue solution is more efficient than non-learnable methods [99] which is given in table 2.

Table 2 Semantic Segmentation results on PASCAL VOC and Test Set

Methods	Training	Validation	Test	Descriptions
SDS	VOC extra	53.9	51.9	Applied Region Proposal for Input Images
CFM	VOC extra	60.9	61.8	Applied Region Proposal for Feature Map

FCN-8s	VOC extra	--	62.2	Three Strides in one model
Hyper-column	VOC extra	59	62.6	Applied Region Proposal for Input Images
DeepLab	VOC extra	63.7	66.4	One Stride in one model
DeepLab-MSc-LargeFOV	VOC extra	68.7	71.6	View Field in multi-scale
Piecewise-DCRFs	VOC extra	70.3	70.7	5 models using 3 scales image
CRF-RNN	VOC extra	69.6	72	RNN
BoxSup	COCO+VOC extra	68.2	71	Weakly Supervised Learning
Cross-Joint	COCO+VOC extra	71.7	73.9	Weakly Supervised Learning

E. Image Retrieval

In order to learn an end-to-end mappings between low-resolution (LR) and highresolution (HR) pictures, Dong et al. (2016) [100] created a technique for image superresolution (SR). This technique includes three levels and is based on CNN. A second layer is utilised to map these feature maps from LR patches to HR feature maps after the initial layer extracts the features from the LR patches. By merging the predictions, the final layer reconstructs HR. This system is quick and lightweight enough for online use. To learn the mapping from a noisy image to a noise-free image, Burger et al. (2012) [101] utilised a straightforward MLP. This technique is suitable for less widely researched kinds of noise and keeps up with other cutting-edge denoising techniques [such Dabov et al. (2007) [102]. An original concept was put out by Lehtinen et al. (2018) [103] to recover images by simply glancing at the corrupted samples without using clean training targets. The researchers were able to rebuild the signals from noise to clean by applying basic signal reconstruction techniques using ML to map damaged observations to clean signals. According to the outcome, the noisy targets on the validation data had a mean Peak Signal to Noise Ratio (PSNR) of 31.74 dB. In 31.77 dB, the network trained with clean targets is effective at rebuilding the signal from the noisy to the clean. To improve the effectiveness of deep convolutional networks in image restoration, many researchers [104][105][106] have conducted numerous pre-training sessions using a huge number of realistic images. However, Ulyanov et al. (2018) [107] demonstrated that a convolutional image generator's structure may collect a significant amount of picture statistics even in the absence of learning. Their approach does not need pre-training or a degradation modelling technique. It works well in SR, inpainting, and denoising, though. The process is relatively slow because it necessitates many iterations. In other disciplines, including biological image data, it has trouble finding training datasets. In order to train denoising CNNs, Krull et al. (2019) [108] developed NOISE2VOID (N2V), a training technique that only needs a single noisy acquisition. They suggested a blind-spot network, in which each pixel's receptive area excludes itself, preventing it from figuring out its identity. The authors present a novel method for training the network to adapt towards the fields that acquire limited or realistic low-resolution training datasets, as a result of which N2V also couldn't eliminate the noise effectively if the assumption of independence could not be satisfied. A GANbased combined SR and inverse tone-mapping network (SR-ITM) known as JSI-GAN was created by Kim et al. in 2020[109].

F. Human Pose Estimation

The purpose of human posture estimation is to identify the location of human joints using photographs, sequencing of images, depth images, or skeletons information that is provided by motion-capture hardware [110]. Due to the wide variety of human shapes and looks, difficult lighting, and cluttered background, estimating human stance is a very difficult assignment. Prior to the advent of deep learning, pose estimation relied on body part identification, such as through visual structures [111]. Proceeding on to deep learning techniques for estimating human poses, we can divide them into holistically and part-based approaches based on how the input photos are handled. The holistic processing techniques often complete their purpose in an overall manner without defining a model for each component's particular location and interactions. DeepPose [112] is a comprehensive model which formulates the joint regression problem for such human pose estimation method rather than defining the graphical model or component detectors explicitly. However, due to the difficulty in learning straight regression of complicated pose

vectors from photos, holistic based approaches frequently suffer from inaccuracies in the high-precision region. The part-based processing techniques, on the other hand, concentrate on identifying each unique human body part before using a graphic model to include the spatial data. In order to understand conditional probabilities of both the portion of presence and spatial relationships, the authors in [113] train a CNN utilising local part spots and background patches rather than the entire image. The method described in [114] involves training several smaller CNNs to conduct independent binary body-part classification, followed by a higher-level weak spatial model to weed out strong outliers and enforce global pose consistency. Next, in[115], an implicit graphic model is used to further encourage joint consistency when a multi-resolution CNN is created that perform heat-map likelihood regression for each body component.

G. Advance Machine Learning Architectures

- *Deep Feed Forward Networks*

The simplest deep architecture, a deep feedforward neural network just transfers the connection between the nodes. Deep neural networks are generally referred to as multilayer neural networks that include a large number of hidden layers[116]. Deep Feed-Forward Network with n hidden layers is Compared to shallow design, multiple hidden layers are more effective in modelling complicated nonlinear relationships. Because to the multilevel learning made possible by the numerous levels of nonlinearity, a complicated function can be approximated with fewer computational units than a shallow network that performs similarly [117]. It is a consistently well-liked design among scholars and scientists in practically all engineering areas due to the simplicity of the design and the training in this model. The most popular learning technique used to train this model is backpropagation with gradient descent[118]. The weights are randomly initialised by the algorithm, and then they are adjusted using gradient descent to minimise the error. Several consecutive forward and reverse passes make up the learning process. In forward pass, the input is passed through several hidden layers of nonlinearity on its way to the output, and in the end, the targeted output is compared to desired output with corresponding input. The calculated error with respect to network weights are back propagated during the backward pass to change the weights in order to reduce output error as much as possible. The procedure is repeated repeatedly until the model prediction showed the expected improvement. The layer I output can be expressed by, if input(Xi) and nonlinear activation function(fi).

$$Xi+1 = fi(WiXi + bi) \quad (1)$$

$$W_{new} = \mathbf{W} - \eta \frac{\partial E}{\partial W} \quad (2)$$

$$b_{new} = \mathbf{b} - \eta \frac{\partial E}{\partial b} \quad (3)$$

If a deep neural network is trained naively, many problems, including overfitting, entrapment in local minima, and vanishing gradient problem, might occur. Such troubling problems impacted sluggish

Around the late 1990s, neural network research was conducted. However, a decade later[119] with the introduction of unsupervised pre-training procedures in deep neural networks, neural network research was once more reinvigorated for use in difficult tasks like voice and vision. Many methods have recently been developed, with varying degrees of success, to address the long-standing problems in deep neural network training including L1 and L2 regularisation, dropout, batch normalisation [120], a great gathering of weight initialization methods [121], and a complete selection of activation functions [122].

- *Restricted Boltzmann Machines (RBM)*

Stochastic neural networks can be viewed as RBM . Due to its capacity to learn the input probability distribution both supervised and unsupervised, an RBM is a wellliked deep learning system. Paul Smolensky initially presented it in 1986. However, with the introduction of better training algorithms in 2002[123], Hinton popularised it. It was then extensively used in a variety of applications, including representation learning, dimension reduction [124], and prediction issues. Nonetheless, deep belief network training employing RBMs as building blocks [125] was a very

significant application in the development of RBMs, one that, together with a few other significant developments covered later, helped usher in the era of deep learning. The hidden units(A) and the visible units(B1) are conditionally independent events since the model is a bipartite graph. As a result, the Boltzmann distribution is satisfied in the equation for both H and With inputs B, we can use P(AB1) to obtain A. In a similar manner, we can determine (B2) using P(B2A). The disparity among B1 and B2 can be reduced by modifying the settings, and the resultant A will be a beneficial aspect of B1. P(AB1)=P(A1B1) P(A2B2)..... P(AnBn) Hinton [125] provides a thorough description and a useful technique for training RBMs. Additional research in [126] explored the key challenges associated with training RBMs, their underlying causes, and proposed a novel approach to overcome those challenges. This technique comprises of an adaptable learning rate and an increased gradient. The noisy linear units of binary units are approximately designed to maintain information about relative intensities as information flows through successive layers of feature detectors, which is a well-known advancement of RBM. In addition to working effectively in this model, the refinement is frequently used in several CNN-based techniques [127].

- *Deep Boltzmann Machines (DBM)*

Generative model that is unsupervised, probabilistic, and has connections between layers that are completely undirected. It contains a number of levels of hidden units as well as visible units. Similar to RBM, DBM lacks an intralayer link. Only the units of the adjacent layers can be connected.Symmetrically connected network of stochastic binary units one side of odd layers and the other make-up of even layers a bipartite graph that can be used to arrange DBM. Although the units within the levels are independent of one another, they are dependent on the layers around them. Layer-by-layer pretraining improves learning efficiency; greedy pretraining differs differently from that used in DBM. Back propagation is used to fine-tune DBM after it has learned all binary features in every layer [128]. Both the performance of classification of the deep feature learner and the probability have shown encouraging gains as a result of this cooperative learning. However, a significant drawback of DBMs is that approximate inference has a far higher temporal complexity than DBNs, which makes collaborative optimization of DBM parameters for big datasets difficult. Other researchers developed an approximate inference approach [129] to speed up inference and boost the effectiveness of DBMs. This algorithm uses a separate" recognition" model to establish latent variables's value in all layers. DBMs effectiveness can improve at pre training [130] or at training step.

- *Autoencoders*

For the purpose of discovering effective encodings, the auto - encoder is a specific kind of ANN. An autoencoder is taught to rebuild its own inputs X, so output vectors are of the same dimension as the input vector, as opposed to the network being trained to predict some goal value Y given inputs X. Fig. 12 depicts an autoencoder's overall operation:The learned feature is the corresponding code, which is obtained during the process by optimising the autoencoder by minimising the reconstruction error. The discriminative and representational properties of raw data are typically difficult for a single layer to capture. The deep autoencoder, which transfers the code learned from one auto - encoder to the next, is currently used by researchers to complete their mission. Hinton et al. [124] made the deep autoencoder's initial proposal, and it is still being thoroughly researched in current works [131]. A deep autoencoder is frequently trained using a back-propagation method version, such as the conjugated gradient method. Although while it is frequently rather useful, if there are flaws in the initial levels, the model may become very ineffective. This might teach the network how to recreate the training data's average. Pre-training the network using initial weights that roughly reflect the final result is a suitable strategy for solving this problem. To making the representation as" constant" with respect to changes in input as possible, several autoencoders have also been proposed. Table 3: We list a few well-known autoencoder variations.

Table 3 Comparison of AutoEncoder Variants

AutoEncoder Variant	Description
Sparse Autoencoder	Achieves sparsity in representation by penalizing hidden unit biases or activations. Benefits include: 1) Enhanced separability of categories in high-dimensional representations; 2) Simple breakdown of complex input data; 3) Mimics sparse representations used in biological vision[132].

Denoising Autoencoder	Improves model robustness by learning to retrieve correct input from corrupted versions. This forces the model to identify the underlying structure of the input distribution [133].
Contractive AutoEncoder	Aims to learn reliable representations by adding an analytical contractive penalty to the reconstruction error function. Unlike Denoising Autoencoders which add noise to the training set, CAEs achieve robustness through this penalty [134].
Conventional AutoEncoder	Adapts non-probabilistic and non-generative characteristics for generative modeling. Can be used to produce useful samples from the network[135].
Saturating Autoencoders	Increases reconstruction error for inputs far from the data surface. This limits the ability to reconstruct inputs not close to the data surface[136].

- *Convolutional Neural Networks(CNN)*

A kind of neural networks called CNN are modelled after the visual system of humans. Although LeCun et al.[137] first suggested the notion in 1998, it wasn't until Krizhevsky et al. [138] were successful in winning the ILSVRC-2012 contest with the architecture known as AlexNet that the deep learning community actually got to see it in action. After this amazing victory, the computing community witnessed CNN and its derivatives' astounding classification abilities, ushering in a new artificial intelligence era. Several derivative designs have been presented and are still being explored in the years since. These CNN designs have frequently and easily outperformed human recognition capabilities. An input tensor is abstracted to a feature map using numerous local kernel filters utilising a sequence of convolution layers in the hidden layer of CNN. Convolutional filters are used to the input data's width and height to produce two-dimensional feature maps. The generated feature maps for all filters are stacked to create the convolution layer's output, which is then sent on to the following layer for down sampling. This layer is often a pooling layer that shrinks input's spatial dimension. In order to achieve more abstracted features, the dimension of the data can be further decreased by stacking further convolutions and pooling layers. A fully connected layer may be connected to the generated feature map for classification or regression[139]. Convolutional layers, pooling layers, and fully connected layers are the three primary categories of neural layers found in a CNN. Several layers have different functions. A CNN maps the input data to a 1D feature vector at each layer by converting the input volume to an output volume of neuron activation. This process continues until the final fully linked layers [140].

Convolution Layer. The foundational component of the CNN is the convolution layer. It carries the majority of the computational load on the network. This layer creates a dot product between feature matrix and kernel matrix, Compared to a picture, the kernel is smaller in space but deeper. This indicates that the kernel width and height will be spatially small if the image consists of RGB channels, but the depth will go up to all three channels [141]. Size of Output Volume: -

$$\text{Wout} = \frac{W - F + 2P}{S} + 1 \tag{4}$$

where p is amount of padding and S is stride and W is weight.

Pooling Layer. By calculating an aggregate statistic from the surrounding outputs, the pooling layer substitutes for the network's output at specific locations. This aids in shrinking the representation's spatial size, which lowers the computational time, and weights needed. Each slice of the image is subjected to the pooling operation separately. There are various types of pooling like max pooling [142], min pooling and Average pooling. Size of Output Volume: -

$$\text{Wout} = \frac{W - F}{S} + 1 \tag{5}$$

Fully Connected Layer. As in a conventional FCNN, all of the neurons in this layer are fully connected to all of the neurons in the layer before and after. Because of this, it can be calculated using a matrix multiplication accompanied by a bias effect, as per usual [143]. The representation between both the input and the output is mapped using the FC layer.

Non-Linearity Layers. Non-linearity layers are frequently included right after the convolutional layer to add non-linearity to the activation map because convolution is a linear operation and pictures are anything but linear [144].
Sigmoid: - The sigmoid activation function is primarily utilised because it occurs between (0 to 1). As the probability of anything occurs between 0 and 1, sigmoid is the best option for models where we must anticipate the likelihood as an output. A neural network may become stuck during training as a result of the logistic sigmoid function. As a result, the output layer is where it is most often used. (When a binary classifier is used) • Tanh:- Tanh is a shifted variation of the sigmoid function, with a -1 to 1 range. The data is more centred because the mean of activation functions that emerge from the hidden layer is closer to zero, which facilitates and accelerates learning for the following layer.One drawback of both sigmoid and tanh is that the gradient of this function grows very small and ultimately approaches zero if our weighted sum input(z) seems to be either very large or very small. This may cause a steep descent to lag.
Relu:- Relu is becoming a more popular activation function by default. Use the Relu activation function and one of its variations if you are unsure what to utilise in the hidden layers (we will see them later). The fact that it does not saturate for large input values, unlike the hyperbolic tangent function and logistic function, which saturate at 1, makes it easier to compute than other activation functions and helps gradient descent avoid plateaus. Relu has the drawback that when weighted input is negative, the derivative equals zero. The issue is referred to as the fading Relu. A Relu neuron won't be able to successfully contribute to the training phase if the network weights always result in negative inputs into it.

- *Recurrent Neural Networks (RNN)*

RNNs operate on the premise that the output of one layer is saved and fed back into the input in order to anticipate the output of that layer [145]. A Feed-Forward Neural Network can be changed into RNN as shown below:
A single layer recurrent- neural networks is created by compressing the nodes from several layers of the neural network. The variables of the network are, A,B, and C. The input layer is "x," the hidden layer is "h," and the output layer is "y" in this instance. The network parameters A, B, and C help the model's output to be more accurate. The input at any given time t is a mixture of the input at x(t) and x. (t-1).

$$\mathbf{h}(t) = f_c(\mathbf{h}(t-1), \mathbf{x}(t)) \quad (6)$$

where h(t) is new state, fc is function with parameter c , h(t-1) is old state and x(t) is input vector at time step t. There were a few problems with the feed-forward neural network, which led to the development of RNN:can't deal with consecutive data, merely takes into account current input, and unable to remember earlier inputs, but there are also some problems in RNN i.e problems with disappearing

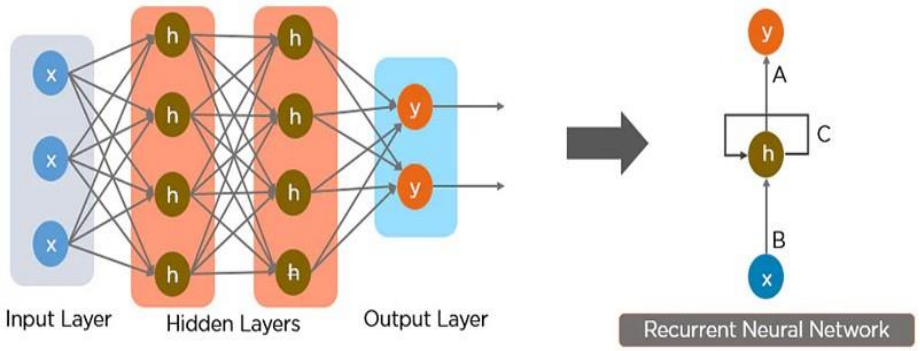

Fig. 3 Architecture of RNN

and exploding gradients so next updated technique is Longest Short-Term Memory Networks (LSTMs).

- *Longest Short-Term Memory Networks (LSTMs)*

The default behaviour of LSTMs, a particular type of RNN, is to recall input for extended durations in order to learn long-term dependencies. Every RNN take the shape of a series of neural network repeating modules [146]. This recurring module in typical RNNs will be made up of just one tanh layer, for example. Although the repeating module of LSTMs also has a chain-like shape, it is slightly different. Four connecting layers are engaging

tremendously in place of a single layer of neural networks. In LSTMs there are 3 steps procedure which is given below[147]:- Step 1:Determine How Much Historical Data It Should Recall , Choosing which data should be excluded from the cell at that specific time step is the first stage in the LSTM. This is decided using the sigmoid function. It computes the function while taking into account the current input xt and the previous state (ht-1).

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \quad (7)$$

(ft) decides which information has to delete which is not important from previous data Step-2:- Assess the Contribution of This Unit to the Present Situation. There are two sections in the second stratum. The tanh function and the sigmoid function are the two. It determines which values in the sigmoid function to allow through (0 or 1). The tanh function determines the importance of the values by giving them weight (-1 to 1).

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \quad (8)$$
$$\mathbf{c}_t = \tanh(\mathbf{W}_c \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c) \quad (9)$$

Step-3:-Choose Which Much of the Current Cell State Is Included in the Output Making a decision regarding the output is the third phase. To determine which components of the cell state are output, we first run a sigmoid layer. Finally, we multiply the cell state by the outcome of the sigmoid gate after pushing the values through tanh to be between -1 and 1.

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \cdot [\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \quad (10)$$
$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (11)$$

In order to train for handwriting recognition, Doetsch et al[148] suggested an LSTM-based training framework made up of sequence chunks that comprise tiny batches. The architecture uses redesigned gating units with layer-specific weights for each gate to reduce runtime by a factor of 3. In order to acquire a comprehensive semantic information of the complete sentence, Palangi et al.[149] built a sentence embedding model utilising LSTM-RNN that successively extracts knowledge from each word and embeds it in a semantic vector until the end of the sentence. The model that can reduce irrelevant words and detect crucial keywords is very helpful in applications for retrieving documents from the web. In order to correlate a series of Part-Of-Speech (POS) tags to a sequence of words, researchers developed a Bi-LSTM architecture, which finds an intriguing use in Natural Language Processing [150]. To provide an end-to-end model for target separation and localisation for visual tracking applications, Gao et al. [151] suggested a novel hierarchy of attentional module with large short-term memory and multi-layer perceptrons.

IV. Conclusion and Future Scope

In this research, we found various cutting-edge deep learning (DL) methods for the CV application scenario, including AlexNet, VGGNet, GoogLeNet and Inception and many more. We looked into where they came from and gave a critical evaluation of several key scientific findings. We concentrated on four essential CV tasks: picture restoration, semantic segmentation [152], visual tracking, and recognition. Also, we looked into and highlighted how well these strategies performed in each circumstance. This paper has mapped out an in-depth evaluation of the latest deep learning architectures with the goal of assisting both novice and experienced practitioners in making well-informed decisions. It has taken things a step further by providing an exploratory analysis of a few of the most popular deep learning application domains. The goal is for readers to find the survey information to be both interesting and simple to grasp, keeping them informed of the most recent advancements in this fascinating sector that has so much potential for the future in improvement of Computer Vision's performance using advance machine learning techniques.

In the last ten years, there has been a substantial advancement in the use of DL in CV. So there are some future trends which plays vital role in CV.

- *Architecture Exploration and network type exploration:-* Networks come in a variety of enriched types. Siamese neural networks (SNN), recurrent neural networks (RNN), generative adversarial networks (GAN), and custom networks are among the many varieties that have been developed. Recent research for CV scenarios have

steadily relied on semi-supervised learning. To put it another way, we saw a progression from supervised methods to semi-supervised learning.

- *Expanding CV's application domains: Crossover* application studies are enriched. In addition to the industrial prediction of petroleum output [153], CV application research have been expanded to the medical arena, such as cancer detection by semantic segmentation [152], and archaeological, such as recovering ancient writings [154].

- *Combinatorial uses of CV with Advanced Machine learning domains*:In addition to just circling their respective fields, studies in CV frequently incorporate with Advanced Machine learning domains for combinatorial applications. For instance, chatbots use much more NLP techniques to increase response accuracy [155], where it is challenging to discern the nature of the conversation. Chatbots face a lot of difficulty simulating real communication by fully comprehending the inner workings of the user. Chatbots can interpret users' emotions from microexpressions using motion analysis and facial expression recognition, and they can even combine these analyses with psychological theories to assess the results.

- *Improvement of more specialised application scenarios:* As CV approaches have improved, specialised application scenarios have emerged, such as the examples of GAN used for 3D semantic segmentation, face and gesture recognition, stylization, and machines creation. As a result, researchers frequently tweak the usual CV procedures and cues to make them more effective for a particular subdivision.

- *Model Interpretability and Visualization:* ML, including DL, is frequently referred to as a" black box." The vast amount of the dataset could not be processed effectively by conventional designs of CV approaches. The end-to-end learning technique that is DL-based allows for less concern over the quantity of the dataset. The DL model could be employed in delicate or unique application settings, such as medical surgery, after being trained on very sizable datasets. Hence, in order for outsiders (such as medical professionals or surgeons) to comprehend the technological foundation for the determination, visualising and interpreting the DL model is required. The first step is to improve DL model visualisation and interpretability.
- *Scalability of the model:* DL models are currently reported in large numbers, and their architectures are becoming more complex. Also, developing the DL model takes time. So, a criterion for judging this approach is if it is easily scalable. A model might be trained with a basic framework using little time and data. The model's cooperation could be improved to satisfy more requirements with higher levels of accuracy.

CONFLICT OF INTEREST

The authors declare no conflict of interest in preparing this article.

DATA AVAILABILITY

The data used to support the findings of this study are included within the article.

FUNDING

The authors declare no funding from any sources in preparing this article.

REFERENCES

[1] Chai, J., Zeng, H., Li, A., Ngai, E.W.: Deep learning in computer vision: A critical review of emerging techniques and application scenarios. Machine Learning with Applications 6, 100134 (2021)

[2] Sengupta, S., Basak, S., Saikia, P., Paul, S., Tsalavoutis, V., Atiah, F., Ravi, V., Peters, A.: A review of deep learning with special emphasis on architectures, applications and recent trends. Knowledge-Based Systems 194, 105596 (2020)

[3] Khamparia, A., Singh, K.M.: A systematic review on deep learning architectures and applications. Expert Systems 36(3), 12400 (2019)

[4] Demush, R.: A Brief History of Computer Vision (and Convolutional Neural Networks). https://hackernoon.com/a-brief-history-of-computer-vision-and-convolutional-neural-networks-8fe8aacc79f3. Accessed: February 27, 2019 (2019)

[5] Marr, B.: Amazing examples of computer and machine vision in practice. Forbes (2019)

[6] Van Herwaarden, A., Angus, J., Richards, R., Farquhar, G.: 'haying-off', the negative grain yield response of dryland wheat to nitrogen fertiliser ii. carbohydrate and protein dynamics. Australian Journal of Agricultural Research 49(7), 1083–1094 (1998)
[7] Rautaray, S.S., Agrawal, A.: Vision-based hand gesture recognition for humancomputer interaction: a survey. Artificial Intelligence Review 43(1), 1–54 (2015)
[8] Zhao, F., Xie, X., Roach, M.: Computer vision techniques for transcatheter intervention. IEEE Journal of Translational Engineering in Health and Medicine 3, 1900331 (2015)
[9] Sigdel, M., Dinc, I., Sigdel, M.S., Dinc, S., Pusey, M.L., Aygun, R.S.: Feature analysis for classification of trace fluorescent labeled protein crystallization images. BioData Mining 10(1), 14 (2017)
[10] Guo, M., Li, J., Sheng, C., Xu, J., Wu, L.: A review of wetland remote sensing. Sensors 17(4), 777 (2017)
[11] Breen, G.-M., Matusitz, J.: An evolutionary examination of telemedicine: A health and computer-mediated communication perspective. Social work in public health 25(1), 59–71 (2010)
[12] Van Gerven, M., Bohte, S.: Editorial: Artificial neural networks as models of neural information processing. Frontiers in Computational Neuroscience 11, 114 (2017)
[13] Hornik, K.: Approximation capabilities of multilayer feedforward networks. Neural networks 4(2), 251–257 (1991)
[14] Cybenko, G.: Approximation by superpositions of a sigmoidal function. Mathematics of control, signals and systems 2(4), 303–314 (1989)
[15] Hanin, B.: Universal function approximation by deep neural nets with bounded width and relu activations. arXiv preprint arXiv:1708.02691 (2017)
[16] Widrow, B., Lehr, M.A.: 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. Proceedings of the IEEE 78(9), 1415–1442 (1990)
[17] Rosenblatt, F.: The perceptron: A probabilistic model for information storage and organization in the brain. Psychological review 65(6), 386 (1958)
[18] McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics 5(4), 115–133 (1943)
[19] Widrow, B., Lehr, M.A.: 30 years of adaptive neural networks: perceptron, madaline, and backpropagation. Proceedings of the IEEE 78(9), 1415–1442 (1990)
[20] Minsky, M., Papert, S.: Perceptrons - an Introduction to Computational Geometry. MIT Press(1969)
[21] Werbos, P.J.: The Roots of Backpropagation: from Ordered Derivatives to Neural Networks and Political Forecasting vol. 1. John Wiley & Sons, ??? (1994)
[22] Theano Development Team: Theano: A python framework for fast computation of mathematical expressions. arXiv preprint arXiv:1605.02688 (2016)
[23] Seide, F., Agarwal, A.: Cntk: Microsoft's open-source deep-learning toolkit. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 2135–2135 (2016)
[24] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
[25] Chollet, F., et al.: Keras. https://github.com/fchollet/keras. Documentation: http://keras.io (2015)
[26] Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch (2017)
[27] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM International Conference on Multimedia, pp. 675–678 (2014)
[28] Tokui, S., Oono, K., Hido, S., Clayton, J.: Chainer: a next-generation open source framework for deep learning. In: Proceedings of Workshop on Machine Learning Systems (LearningSys) in the Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS), vol. 5, pp. 1–6 (2015)
[29] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. Journal of Machine Learning Research 12, 2825–2830 (2011)
[30] Hopfield, J.J.: Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences 79(8), 2554–2558 (1982)
[31] Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. In: Rumelhart, D.E., McClelland, J.L., Group, P.R. (eds.) Parallel Distributed Processing: Explorations in the Microstructure of Cognition vol. 1. MIT Press,(1986)
[32] Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning 20(3), 273– 297 (1995)
[33] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation 9(8), 1735–1780 (1997)
[34] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)

[35] Hinton, G.E., Osindero, S., Teh, Y.-W.: A fast learning algorithm for deep belief nets. Neural Computation 18(7), 1527–1554 (2006)
[36] Voulodimos, A., Doulamis, N., Doulamis, A., Protopapadakis, E.: Deep learning for computer vision: A brief review. Computational Intelligence and Neuroscience 2018 (2018)
[37] Bordes, A., Glorot, X., Weston, J., Bengio, Y.: Joint learning of words and meaning representations for open-text semantic parsing. In: Artificial Intelligence and Statistics, pp. 127–135 (2012). PMLR
[38] Ren, J., Xu, L.: On vectorization of deep convolutional neural networks for vision tasks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29 (2015)
[39] Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. Advances in Neural Information Processing Systems 26 (2013)
[40] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM 60(6), 84–90 (2017)
[41] Deng, L.: A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Transactions on Signal and Information Processing 3, 2 (2014)
[42] Chai, J., Li, A.: Deep learning in natural language processing: A state-of-the-art survey. In: 2019 International Conference on Machine Learning and Cybernetics (ICMLC), pp. 1–6 (2019). IEEE
[43] Adamopoulou, E., Moussiades, L.: Chatbots: History, technology, and applications. Machine Learning with Applications 2, 100006 (2020)
[44] Vickers, N.J.: Animal communication: when i'm calling you, will you answer too? Current Biology 27(14), 713–715 (2017)
[45] Altan, A., Karasu, S., Zio, E.: A new hybrid model for wind speed forecasting combining long short-term memory neural network, decomposition methods and grey wolf optimizer. Applied Soft Computing 100, 106996 (2021)
[46] Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., Lew, M.S.: Deep learning for visual understanding: A review. Neurocomputing 187, 27–48 (2016)
[47] Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamar´ıa, J., Fadhel, M.A., Al-Amidie, M., Farhan, L.: Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. Journal of Big Data 8(1), 1–74 (2021)
[48] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research 15(1), 1929–1958 (2014)
[49] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
[50] Cheng, Y., Wang, D., Zhou, P., Zhang, T.: A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282 (2017)
[51] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
[52] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
[53] Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 31 (2017)
[54] Chollet, F.: Deep learning with separable convolutions. arXiv preprint arXiv:1610.02357 (2016)
[55] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90 . IEEE
[56] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
[57] Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Doll´ar, P.: Designing network design spaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10428–10436 (2020)
[58] Vailaya, A., Figueiredo, M.A., Jain, A.K., Zhang, H.-J.: Image classification for content-based indexing. IEEE transactions on image processing 10(1), 117–130 (2001)
[59] Collins, R.T., Lipton, A.J., Kanade, T., Fujiyoshi, H., Duggins, D., Tsin, Y., Tolliver, D.e.a.: A system for video surveillance and monitoring. VSAM final report 2000(1-68), 1 (2000)
[60] Kosala, R., Blockeel, H.: Web mining research: A survey. ACM Sigkdd Explorations Newsletter 2(1), 1–15 (2000)
[61] Pavlovic, V.I., Sharma, R., Huang, T.S.: Visual interpretation of hand gestures for human-computer interaction: A review. IEEE Transactions on pattern analysis and machine intelligence 19(7), 677–695 (1997)
[62] Jain, A.K., Ross, A., Prabhakar, S.: An introduction to biometric recognition. IEEE Transactions on circuits and systems for video technology 14(1), 4–20 (2004)

[63] Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, ECCV, vol. 1, pp. 1–2 (2004)

[64] Chang, C.-C., Lin, C.-J.: Libsvm: a library for support vector machines. ACM transactions on intelligent systems and technology (TIST) 2(3), 1–27 (2011)

[65] Lapuschkin, S., Binder, A., Montavon, G., Mu¨ller, K.-R., Samek, W.: The lrp toolbox for artificial neural networks. The Journal of Machine Learning Research 17(1), 3938–3942 (2016)

[66] Gando, G., Yamada, T., Sato, H., Oyama, S., Kurihara, M.: Fine-tuning deep convolutional neural networks for distinguishing illustrations from photographs. Expert Systems with Applications 66, 295–301 (2016)

[67] Garc´ıa Ocan˜a, M.I., L´opez-Linares Rom´an, K., Lete Urzelai, N., Gonz´alez Ballester, M.A., Mac´ıa Oliver, I.: Medical image detection using´ deep learning. Deep Learning in Healthcare: Paradigms and Applications, 3–16 (2020)

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

[69] Jiang, P., Ergu, D., Liu, F., Cai, Y., Ma, B.: A review of yolo algorithm developments. Procedia Computer Science 199, 1066–1073 (2022)

[70] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I, vol. 9905, pp. 21–37 (2016). Springer International Publishing

[71] Isola, P., Xiao, J., Torralba, A., Oliva, A.: What makes an image memorable? In: CVPR 2011, pp. 145–152 (2011). IEEE

[72] Cai, Z., Fan, Q., Feris, R.S., Vasconcelos, N.: A unified multi-scale deep convolutional neural network for fast object detection. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV, vol. 9910, pp. 354–370 (2016). Springer International Publishing

[73] Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: Dssd: Deconvolutional single shot detector. arXiv preprint arXiv:1701.06659 (2017)

[74] Lin, T.-Y., Goyal, P., Girshick, R., He, K., Doll´ar, P.: Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988 (2017). IEEE

[75] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)

[76] He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence 37(9), 1904–1916 (2015)

[77] Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)

[78] Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 6517–6525 (2017). https://doi.org/10.1109/TPAMI.2016.2577031 . IEEE

[79] Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS), pp. 379–387 (2016). NIPS

[80] Lin, T.-Y., Doll´ar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)

[81] He, K., Gkioxari, G., Doll´ar, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)

[82] Yilmaz, A., Javed, O., Shah, M.: Object tracking: A survey. ACM Computing Surveys 38(4), 1–45 (2006)

[83] Raina, R., Shen, Y., McCallum, A., Ng, A.Y.: Classification with hybrid generative/discriminative models. In: Advances in Neural Information Processing Systems, vol. 16 (2003)

[84] Walia, G.S., Kapoor, R.: Recent advances on multicue object tracking: A survey. The Artificial Intelligence Review 46(1), 1–39 (2016)

[85] Kumar, A., Walia, G.S., Sharma, K.: Recent trends in multicue based visual tracking: A review. Expert Systems with Applications 162, 113711 (2020)

[86] Wang, N., Yeung, D.-Y.: Learning a deep compact image representation for visual tracking. In: Advances in Neural Information Processing Systems, vol. 26 (2013)

[87] Wang, C., Huang, K., Ren, W., Zhang, J., Maybank, S.J.: Large-scale weakly supervised object localization via latent category learning. IEEE Transactions on Image Processing 24(4), 1371–1385 (2015)

[88] Zhang, K., Liu, Q., Wu, Y., Yang, M.-H.: Robust visual tracking via convolutional networks without training. IEEE Transactions on Image Processing 25(4), 1779–1792 (2016)

[89] Liu, H., Tian, Y., Yang, Y., Pang, L., Huang, T.: Deep relative distance learning: Tell the difference between similar vehicles. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2167–2175 (2016)

[90] Yun, S., Choi, J., Yoo, Y., Yun, K., Choi, J.Y.: Action-decision networks for visual tracking with deep reinforcement learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2711–2720 (2017)
[91] Song, Y., Ma, C., Wu, X., Gong, L., Bao, L., Zuo, W., Shen, C., Lau, R.W.H., Yang, M.-H.: Vital: Visual tracking via adversarial learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8990–8999 (2018)
[92] Lukezic, A., Matas, J., Kristan, M.: D3s-a discriminative single shot segmentation tracker. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7133–7142 (2020)
[93] Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440 (2015)
[94] Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1520–1528 (2015)
[95] Hong, S., Noh, H., Han, B.: Decoupled deep neural network for semi-supervised semantic segmentation. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
[96] Pinheiro, P.O., Lin, T.-Y., Collobert, R., Doll´ar, P.: Learning to refine object segments. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I, vol. 9905, pp. 75–91 (2016). Springer International Publishing
[97] He, K., Gkioxari, G., Doll´ar, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988 (2017). https://doi.org/10.1109/ICCV.2017.322
[98] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 801–818 (2018)
[99] Zhang, C., Lin, G., Liu, F., Yao, R., Shen, C.: Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5217–5226 (2019)
[100] Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE transactions on pattern analysis and machine intelligence 38(2), 295–307 (2015)
[101] Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: Can plain neural networks compete with bm3d? In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 2392–2399 (2012). IEEE
[102] Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on image processing 16(8), 2080–2095 (2007)
[103] Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2noise: Learning image restoration without clean data. arXiv preprint arXiv:1803.04189 (2018)
[104] Dosovitskiy, A., Brox, T.: Inverting visual representations with convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4829–4837 (2016)
[105] Lai, W.-S., Huang, J.-B., Ahuja, N., Yang, M.-H.: Deep laplacian pyramid networks for fast and accurate super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 624–632 (2017)
[106] Ledig, C., Theis, L., Husz´ar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4681–4690 (2017)
[107] Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. International Journal of Computer Vision 128(7), 1867–1888 (2020)
[108] Krull, A., Buchholz, T.-O., Jug, F.: Noise2void-learning denoising from single noisy images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2129–2137 (2019)
[109] Kim, S.Y., Oh, J., Kim, M.: Jsi-gan: Gan-based joint super-resolution and inverse tone-mapping with pixel-wise task-specific filters for uhd hdr video. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 11287–11295 (2020)
[110] Kitsikidis, A., Dimitropoulos, K., Douka, S., Grammalidis, N.: Dance analysis using multiple kinect sensors. In: Proceedings of the 9th International Conference on Computer Vision Theory and Applications, VISAPP, pp. 789–795 (2014)
[111] Felzenszwalb, P.F., Huttenlocher, D.P.: Pictorial structures for object recognition. International Journal of Computer Vision 61(1), 55–79 (2005)
[112] Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. In: Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 1653–1660 (2014). IEEE
[113] Chen, X., Yuille, A.L.: Articulated pose estimation by a graphical model with image dependent pairwise relations. In: Proceedings of the NIPS (2014)
[114] Jain, A., Tompson, J., Andriluka, M.: Learning human pose estimation features with convolutional networks. In: Proceedings of the ICLR (2014)

[115] Tompson, J.J., Jain, A., LeCun, Y., et al.: Joint training of a convolutional network and a graphical model for human pose estimation. In: Proceedings of the NIPS (2014)
[116] Deng, L., Yu, D.: Deep learning: methods and applications. Foundations and trends® in signal processing 7(3–4), 197–387 (2014)
[117] Bengio, Y.: Learning deep architectures for ai. Foundations and trends® in Machine Learning 2(1), 1–127 (2009)
[118] Bengio, Y., Lamblin, P., Popovici, D., Larochelle, H.: Greedy layer-wise training of deep networks. In: Advances in Neural Information Processing Systems, vol. 19 (2006)
[119] Hinton, G.E., Osindero, S., Teh, Y.-W.: A fast learning algorithm for deep belief nets. Neural computation 18(7), 1527–1554 (2006)
[120] Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International Conference on Machine Learning, pp. 448–456 (2015). PMLR
[121] Kumar, S.K.: On weight initialization in deep neural networks. arXiv preprint arXiv:1704.08863 (2017)
[122] Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. ICML, vol. 30, p. 3 (2013)
[123] Hinton, G.E.: Training products of experts by minimizing contrastive divergence. Neural computation 14(8), 1771–1800 (2002)
[124] Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science 313(5786), 504–507 (2006)
[125] Larochelle, H., Bengio, Y.: Classification using discriminative restricted boltzmann machines. In: Proceedings of the 25th International Conference on Machine Learning, pp. 536–543 (2008)
[126] Salakhutdinov, R., Mnih, A., Hinton, G.: Restricted boltzmann machines for collaborative filtering. In: Proceedings of the 24th International Conference on Machine Learning, pp. 791–798 (2007)
[127] Zeiler, M.D., Fergus, R.: Stochastic pooling for regularization of deep convolutional neural networks. arXiv preprint arXiv:1301.3557 (2013)
[128] Salakhutdinov, R., Larochelle, H.: Efficient learning of deep boltzmann machines. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pp. 693–700 (2010). JMLR Workshop and Conference Proceedings
[129] Cho, K.H., Raiko, T., Ilin, A.: A two-stage pretraining algorithm for deep boltzmann machines. In: Proceedings of the ICANN (2013)
[130] Montavon, G., Mu¨ller, K.R.: Deep boltzmann machines and the centering trick. In: Neural Networks: Tricks of the Trade, pp. 621–637. Springer, ??? (2012)
[131] Zhou, Y., Arpit, D., Nwogu, I., Govindaraju, V.: Is joint training better for deep auto-encoders? arXiv preprint arXiv:1405.1380 (2014)
[132] Zou, W.Y., Ng, A.Y., Yu, K.: Unsupervised learning of visual invariance with temporal coherence. In: Proceedings of the NIPS Workshop (2011)
[133] Vincent, P., Larochelle, H., Bengio, Y.: Extracting and composing robust features with denoising autoencoders. In: Proceedings of the ICML (2008)
[134] Rifai, S., Vincent, P., Muller, X.: Contractive auto-encoders: explicit invariance during feature extraction. In: Proceedings of the ICML (2011)
[135] Mesnil, G., Dauphin, Y., Glorot, X.: Unsupervised and transfer learning challenge: a deep learning approach. In: Proceedings of the ICML (2012)
[136] Jiang, X., Zhang, Y., Zhang, W.: A novel sparse auto-encoder for deep unsupervised learning. In: Proceedings of the ICACI (2013)
[137] Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE 86(11), 2278–2324 (1998) https://doi.org/ 10.1109/5.726791
[138] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM 60(6), 84–90 (2017)
[139] Nasir, V., Sassani, F.: A review on deep learning in machining and tool monitoring: methods, opportunities, and challenges. The International Journal of Advanced Manufacturing Technology 115(9-10), 2683–2709 (2021)
[140] Tygert, M., Bruna, J., Chintala, S., LeCun, Y., Piantino, S., Szlam, A.: A mathematical motivation for complex-valued convolutional networks. Neural computation 28(5), 815–825 (2016)
[141] Szegedy, C., Liu, W., Jia, Y.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, Mass, USA, pp. 1–9 (2015)
[142] Szegedy, C., Liu, W., Jia, Y.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, Mass, USA, pp. 1–9 (2015)
[143] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS), Lake Tahoe, Nev, USA, pp. 1097–1105 (2012)

[144] Dubey, S.R., Singh, S.K., Chaudhuri, B.B.: Activation functions in deep learning: A comprehensive survey and benchmark. Neurocomputing (2022)
[145] Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: Lstm cells and network architectures. Neural Computation 31(7), 1235–1270 (2019)
[146] Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
[147] Gers, F.A., Schmidhuber, J.: Recurrent nets that time and count. In: Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, vol. 3, pp. 189–194 (2000). https://doi.org/10.1109/IJCNN.2000. 861302
[148] Doetsch, P., Kozielski, M., Ney, H.: Fast and robust training of recurrent neural networks for offline handwriting recognition. In: 2014 14th International Conference on Frontiers in Handwriting Recognition, pp. 279–284 (2014). IEEE
[149] Palangi, H., Deng, L., Shen, Y., Gao, J., He, X., Chen, J., Song, X., Ward, R.: Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval. IEEE/ACM Transactions on Audio,
[150] Speech, and Language Processing 24(4), 694–707 (2016)
[151] Pota, M., Marulli, F., Esposito, M., De Pietro, G., Fujita, H.: Multilingual pos tagging by a composite deep architecture based on character-level features and on-the-fly enriched word embeddings. Knowledge-Based Systems 164, 309–323 (2019)
[152] Gao, P., Zhang, Q., Wang, F., Xiao, L., Fujita, H., Zhang, Y.: Learning reinforced attentional representation for end-to-end visual tracking. Information Sciences 517, 52–67 (2020)
[153] Mehrotra, R., Ansari, M.A., Agrawal, R., Anand, R.S.: A transfer learning approach for ai-based classification of brain tumors. Machine Learning with Applications 2, 100003 (2020)
[154] Al-Shabandar, R., Jaddoa, A., Liatsis, P., Hussain, A.J.: A deep gated recurrent neural network for petroleum production forecasting. Machine Learning with Applications 3, 100013 (2021)
[155] Dambrogio, J., Ghassaei, A., Smith, D.S., Jackson, H., Demaine, M.L., Davis, G., Mills, D.e.a.: Unlocking history through automated virtual unfolding of sealed documents imaged by x-ray microtomography. Nature communications 12(1), 1184 (2021)
[156] Adamopoulou, E., Moussiades, L.: Chatbots: History, technology, and applications. Machine Learning with Applications 2, 100006 (2020)