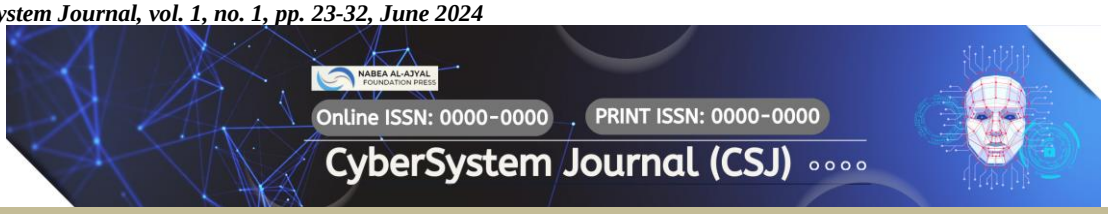

CyberSystem Journal *Review Article*

Deep Learning for Computer Vision: Innovations in Image Recognition and Processing Techniques

Akeel Sh. Mahmoud[1,2*], and Sahar Hamad Ahmed[3]

[1] Computer Center, University of Anbar, Anbar, Iraq
[2] National Engineering School of Tunis, University of Tunis El Manar, Tunis, Tunisia
[3] University Headquarter, University of Anbar, Anbar, Iraq

***** Corresponding Author: Akeel Sh. Mahmoud, Email: akeelab2000@uoanbar.edu.iq

***Abstract*:** Deep learning is a key area of research in the field of computer vision, image processing and bioinformatics. The techniques of deep learning generally are divided into three categories namely Convolutional Neural Networks (CNN), Restricted Boltzmann Machines (RBM), Stacked RBM and HOG (Histograms of oriented Gradient) feature extraction, Convolutional Neural Networks as a Database (CNN as D). Additionally, one in few deep learning architectures which is gaining popularity and is frequently used in the field of computer vision and image processing is Extreme Learning Machine and ensemble of Extreme Learning Machine and CNN. It attempts to survey the recent advances in deep learning researchers and the application of these algorithm in the field of computer vision. Mainly focusing on the deep learning methods and algorithms rather than image processing and computer vision methods, this work inspects deep learning techniques which are widely and commonly used in the field of computer vision image detection and processing like CNN, DBN, RBM and HMM as well as various applications of these techniques. Applications of deep learning techniques in computer vision are image classification, object recognition and detection. Along with the recent works and the future scope for deep learning methods in the field of computer vision and image processing is presented.

Access this article online

***Keywords*:** Architectures, Computer Vision, Convolutional Neural Networks, Restricted Boltzmann Machines (RBM)

1. Introduction to Deep Learning in Computer Vision

Deep learning has become one of the most widely used and researched topics in computer science over the last decade. Most common applications are in the field of artificial intelligence, computer vision, speech recognition, bioinformatics, natural languages processing and many more. Deep learning basically deals with learning high level and more abstract features from data, training large deep neural networks architecture having multiple hidden layers using large training dataset, thus necessitating use of faster hardware and low-level programming languages [1]. This paper focuses largely on the various approaches of deep learning techniques followed by description of its applications in computer vision domain with relevant papers as references.

1.1 Fundamentals of Deep Learning

Deep Learning is a paradigm of Machine Learning that trains multi-layered neural networks to learn data representation with multiple levels of abstraction. Convolutional Neural Networks, Restricted Boltzmann Machines, and Deep Belief Networks are some of the deep learning networks that have been employed and provided satisfactory results on Image and Video framing. These

Received March 25, 2024; Revised April 23, 2024; Accepted May 28, 2024; Published June 30, 2024
https://doi.org/10.57238/n65d0p57

networks automatically learn features without any external feature extraction and preprocessing steps [2]. Apart from CNN, deep learning approaches such as Deep Boltzmann Machines (DBN), Autoencoders, and Recurrent Neural Networks are other popular architectures. Deep learning has recently become one of the most popular sub-fields of machine learning owing to its importance in representation learning with distributed data representation and multiple levels of abstraction. State-of-art deep learning algorithms from various approaches are reviewed, followed by a description of their applications in several conventional computer vision problems.

Vision is one of the most prominent and widely used sensing modalities to analyze surrounding space and objects in order to make very essential intelligent decisions. Human visual recognition system is very complex. With the basic understanding of how human brain perceives, other approaches have also been adopted for visual interpretation such as models of eye physiology, computerized image interpretation, and computational models based on the understanding of brain anatomy. Initial computational models were simple mathematical models of neurons where intellect is presumed to be encoded by the firing rates of these neurons. Deep learning attempts to model high level abstractions in data by using a group of processing layers, with representations between them, where each representation is a nonlinear transformation of its predecessor [3]. On the other hand, Image Recognition or Object Recognition is another sub-field of computer vision having the intent to classify, recognize, and locate objects in the image or video stream.

1.2 Applications of Deep Learning in Computer Vision

Human beings excel at understanding images, even in the midst of a chaotic or mentally stressed environment. However, this is only a very tiny part of the capabilities which biological systems have deployed on their vivid input image data. On the contrary, mostly due to the lack of powerful hardware, traditional computer vision systems fall short of expectations. They employ complex mathematical hand-crafted rules but fail at detecting elements from the images that humans would be able to recognize easily. Nevertheless recent progress in machine learning, on models inspired by the human cortex, deep networks have shown a remarkable ability to analyze fingerprint images 1 and to recognize anonymously scribbled 2D images. The breakthrough and so-called natural image processing abilities of these models are attributed to their deep hierarchical architecture and a convergence of learning algorithms suitable for large sets of parameters.

Deep Learning, along with other computational intelligence algorithms, has attracted great interest from industries and government agencies, with far-reaching consequences for how companies and organizations interact with the world. Industries with a large count of temporary low-paying jobs, as call center job and factory workers, are seen as the most threatened by the apparent exponential growth of this technological field. Nevertheless, a reality is that its technical machinery is also addressing more complex tasks and is going to change the picture of several areas including engineering design, software programming, and white-collar public agencies [4]. Consequently, it is necessary to provide the basis for a public understanding of Deep Learning in the area of computer vision, and also to raise awareness of its consequences for the future society as shown in Figure 1.

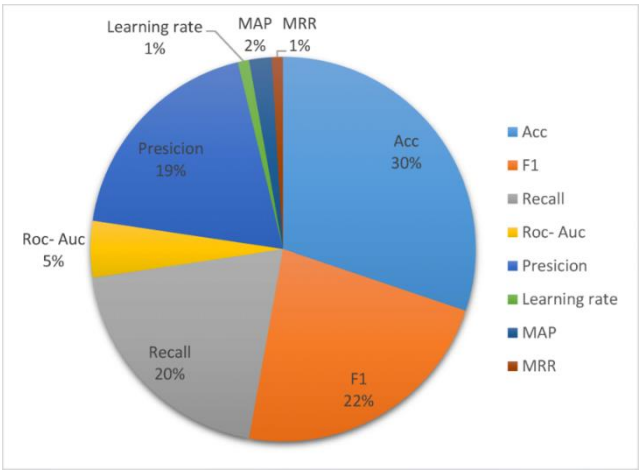

Figure 1. Chart shows the evaluation criteria used in the deep learning

2. Convolutional Neural Networks (CNNs)

Deep learning has led to a remarkable advancement of artificial neural networks (ANN) performance in image recognition and image driven processing tasks. Prominent achievements have been made by a class of ANN known as Convolutional Neural Networks (CNN). CNNs are specifically conceived and designed for vision-related tasks. Primarily because pixels are sparse and necessitate a special form of information processing model. CNNs solve some of the most recurring difficult image-driven pattern recognition tasks that repeatedly occur in different disciplines. There are various forms of CNN architecture, vision-related software applications developed based on CNNs, and several suggestions to improvement aspects of CNN paradigms [5]. Several model architectures accepted by the community as platforms to work from, such as AlexNet or GoogLeNet, are illustrated. Training of these model architectures is done using very large images data sets. Furthermore, suggestions regarding the optimization of certain trained model parameters are provided as initial means of evaluation and experimentation with CNN character. With the fast development of affordable deep learning computation resources, valid predictions can be

obtained with success rates comparable to and often superior to these of human experts [6].

2.1 Architecture of CNNs

Convolutional Neural Networks (CNNs) are a unique architecture of artificial neural networks (ANNs) explicitly designed to process grid-like topology data, especially images. CNNs have gained immense popularity in the research and academic arenas thanks to their ability to learn invariant representation without identifying sets of features in data beforehand. With CNNs, participants do not have to decide which features to extract from the training datasets manually, as the architecture discovers the patterns by itself, ultimately classifying the input data [5]. This section will provide a complete overview of the architectural design and principles of CNNs. The goal is to understand the structure of CNNs and how they work, which is vital for comprehending the architectures of CNNs and their applications [7].

The inner working of CNNs is commonly broken down into five principal layers common to many CNN architectures:

1. The input layer
2. The convolutional layer
3. The activation layer
4. The pooling layer
5. The fully connected layer (FC).

The input layer is the given image of a recognizable object commonly expressed in RGB color space, where each channel is represented by a 2D matrix of floating-point numbers corresponding to brightness levels. CNN's purpose is to learn how to classify input images and recognize the object in them. Each image has many characteristics: color, shape, contrast, brightness, and texture. These attributes have a distance from one object to another with respect to other generic characteristics of objects (e.g., an apple is round and red). Hence, a CNN's task is to learn these characteristics and attribute distances afterward. The architecture of CNNs can be understood as a hierarchy of these attributes.

2.2 Training and Optimization Techniques

An overview of training and optimization techniques specific to CNNs, the most prolific and successful class of neural networks and, as of 2023, the dominant architecture in computer vision. These techniques will equip readers with the knowledge required to effectively train CNNs to undertake any of the computer vision tasks described in Section 2.1, using any computer vision datasets. The general and CNN-specific loss functions and metrics of performance classification tasks are defined first, before introducing the optimization methods that ensure the networks yield optimal results despite the numerous free parameters that require adjustment. The many CNN-specific techniques related to training and optimization that exist are also discussed, although a smaller subset of the same techniques is likely sufficient to yield optimal results for any undertaken tasks (Table 1).

Table 1. Training Labels the Data

Al Agent Levels	Techniques and Capabilities+Tools (Perception + Actions)
L01	Techniques and Capabilities+Tools (Perception + Actions)
L02	LLM-based Al+Tools (Perception +Actions)+Reasoning and Decision Making
L03	LLM-based Al + Tools (Perception)+ Actions + Reasoning & Decision Making
L04	LLM-based Al + Tools (Perception) + Actions + Reasoning & Decision Making + Memory+ Reflection + Autonomous Learning + Generalization
L05	Personality (Emotion + Character) + Collaborative behavior (multi-Agents)

An introduction to the general and CNN-specific training and optimization methods, as of 2023, available to effectively fit CNN free parameters to computer vision data. Any optimization or other training technique labelled as CNN-specific or CNN-related is likely to be based in some way on particular aspects of CNNs, especially the architecture, it is expected that most current ongoing research in this area will be devoted to developing such techniques. Thus, it is hoped that computational computer vision researchers will be encouraged to stay abreast of developments in such techniques and/or pursue alternative lines of research on CNN architecture designing [8]. Training labels the data on which CNN free parameters are fit and refers to all necessary processing steps taken independently of the data. A task, such as image classification or object detection, defines how the CNNs are to be used after their free parameters have been suitably adjusted. It will be assumed throughout this section that tasks have been undertaken using datasets to train CNNs. Using qualitative terms without clear definitions would hazard ambiguity in conveying what was otherwise concrete information, so size and complexity of the tasks and datasets are specifically quantified.

3. Advanced CNN Architectures

The seminal work in CNNs served remarkably well in image classification and understanding fine-grained visual categorization, driven greatly by prominent particulates and structures in the low-resolution visual stimuli. There is continued active research to build and enhance existing architectures for recognizing more challenging classes of the visual dataset, where complex objects persist essentially. The endeavoring ambition to recognize such complex objects typically results into a deep architecture that is severely over-parameterized. To counteract the problem, different approaches of utilizing limited observed visual area through the pooling operation, and semi-supervised learning with Generative Adversarial Networks (GANs) are proposed. But the methodologies introduce their own limitations and fail completely at training much deeper architectures. So, as an alternative approach, it is proposed to utilize the bypassing connection in an additive manner from every layer to the posterior layer without a concern on the network depth, maintaining the benefit of building deeper architectures without the issue of degradation [9].

There are infinitely many neural networks corresponding, as show in Table 2 to an architecture and there exists a special one whose output is identical to the output of the deep network. Identifying an appropriate set of parameters for a generic neural network is essential to solve any interesting problem. CNN is a specific neural network and the most popular architectures are openly available along with pretrained parameter sets. Given an architecture and dataset, the question of determining all parameters of this network that result in a desired output is computationally intensive. Directly estimating the weights is infeasible for any significant network because of the ill-posed nature of the problem and sheer computational burden. Nevertheless, an approximate solution can be incrementally derived via gradient-based optimization algorithm [10].

Table 2. Deep Learning Based MS Classification

CNN based	CNN CNN with multimodal data Graph CNN Wivelet CNN
Hybrid CNN+ Classifier	CNN+ Multilayer Neural Network CNN+ Random Forest CNN+ Random Forest Regressor and Manifold Learning
Deep Transfer Learning	Using AlexNet Using DenseNet Using VGG16

3.1 Residual Networks (ResNets)

Deep networks can suffer from degradation, wherein accuracy gets saturated or even drops with the increase in depth. The straightforward solution of using a deeper model is not attractive. Residual networks (ResNets) can avoid this problem by using a shortcut connection that skips one or more layers. To learn the residual mapping, the stacked nonlinear layers fit a residual mapping $h(x) = F(x) + x$ instead of the original mapping $y = h(x)$. The identity mapping is preserved by the shortcut connection without learning, allowing the flow of information toward the input of the network, and facilitating the signal propagation along the deep net. Experimental results reveal that ResNets are very efficient in training deep architectures. For example, a ResNet with 152 layers has about 60 million parameters, and is trained in 3 days. With all the architectural variants and matching hyper-parameters, ResNets also achieve reasonable performance and reasonable training time across a wide range of depth, such as 34, 50, 101, 152, 1107 layers. Beyond 1000 layers, ResNets are the only networks that converge [11]. There are several interesting aspects of ResNets. First, the deeper the net, the more accurate the classification results. ResNets with different depths outperform the VGG networks with millions more parameters. Second, ResNets significantly improve the performance in both the deep regime and the shallow regime, indicating their versatility. Finally, ResNets produce reasonable results on datasets, such as ILSVRC 2012 image classification and COCO detection, without extensive hyper-parameter tweaking, indicating their robustness [9].

3.2 Inception Networks

Inception networks, so named for the connection of their architecture to the famous dream sequences from the movie “Inception”, begin by performing convolutions of multiple sizes on the same layer which feed into concatenation, allowing the network to learn features from all sizes simultaneously [12]. The first layer would simply operate like a 1 × 1 convolution layer, but numerous pooling convolutions would be performed on top of it. Pooling of different sizes will learn features representing small to large edges in one layer, furthermore helping the learning of the hidden layers through better representation of the input data. Because the mini-batch size in the training phase can dramatically affect the optimization process, if this approach is pursued, for each hidden layer, the concatenation layer would need to be computed from the input of that layer, resulting in a large computational overhead and a complex architecture that would be prone to error and inefficiency.

In this architecture, pooling is replaced with convolutions of strides larger than 1 and the size of the filter is doubled. The 3 × 3 convolution on top of that operates with a stride of 1, and thus it is reminiscent of a 1 × 1 pooling to increase the number of parameters passed to the pooling convolution but not in terms of processing 10. In general, the evolution of the Inception network architectures

proceeds from very high dimensional feature vectors and pooling layer to smaller ones, more appropriate for classification. Inception v3 is part of the Inception Net researches performed by Google. This network aims at balancing the growing width and depth of the architecture with its computational costs and efficacy.

4. Object Detection and Recognition

The objective of object detection is to determine whether there are any instances of objects from a given set of categories in an image and to return the spatial location and extent of each object instance [13]. Object detection forms the basis for solving complex vision tasks such as segmentation, scene understanding, object tracking, and activity recognition. It is a critical component for vision-based software systems. Widespread applications include security, autonomous driving, understanding human interactions, and intelligent video surveillance. Over the past few decades, a large number of object detection algorithms have been proposed, with extensive ongoing research [14]. Recently, deep learning techniques have emerged as a powerful method for learning feature representations automatically from data.

Two major classes of detectors are currently used: detection of broad categories (e.g., cars, people, animals, chairs, etc.) and detection of specific instances (e.g., a given car with license plate number "XYZ 123A"). Historically, a great deal of effort has focused on detecting a single category or a few specific categories while the research community has started moving towards building general purpose object detection systems. The focus in computer vision has largely shifted to deep learning methods including generic object detection. Many deep learning approaches to generic object detection have been proposed recently. Here, first, to build a strong foundation for understanding modern object detection algorithms, a detailed survey is conducted of recent and most important object detection literature. After that, two key methodologies dubbed SSD class of single-shot detectors and Faster-RCNN class of two-shots detectors are focused on more deeply.

4.1 Single Shot MultiBox Detector (SSD)

The Single Shot MultiBox Detector (SSD) is one of the best object detection algorithms that is able to provide high accurate object detection performance in real time. It uses a single deep neural network to perform object detection directly under the full support of convolutional features, and no proposals or extra services are required. In order to provide detection at multiple scales, a set of default boxes with different aspect ratios and scales is generated for each convolutional feature map. The bounding box refinement and classification are then simultaneously performed for each default box in a single network forward pass. This elegant design makes it possible to achieve high speed, and SSD is able to run at 59 frames per second on a Titan, and this high speed is retained even when using larger models [15].

However, SSD shows relatively poor performance on small object detection because its shallow prediction layer (i.e., conv4_3 layer), which is responsible for detecting small objects, lacks enough semantic information (i.e., deep features). To overcome this problem, SKIPSSD, an improved SSD with a novel skip connection of multiscale feature maps, is proposed to enhance the semantic information and the details of the prediction layers through skippingly fusing high-level and low-level feature maps. A new side output is added to the lower order feature map of conv8_2, and the skip connection is made between this new side output and the side output of conv4_3. The feature maps are fused with respect to dimension by means of concatenation, and thus not only the details of the higher order feature map of conv8_2 are kept but also the semantic information of conv4_3 is retained. In this architecture, two predictive layers instead of one predictive layer are produced through side output. Extensive experiments show that SKIPSSD significantly improves the detection performance of small objects and outperforms lots of state-of-the-art object detectors.

4.2 Region-Based CNNs (R-CNNs)

R-CNNs are a family of Region-based CNNs that diagnose and recognizes objects within images. CNNs consider an entire image as singular input and outputs pixel-wise classification or regression models. Starting from the R-CNN, introduced as a two-stage detection algorithm, there are two unique characteristics of using R-CNNs: generating region proposals and the follow-up end-to-end refinements of these proposals from a distinct set of regions. R-CNN proposals are either background or an object. Using a region cause increases in hidden layer size changes considers regressions in pixel switches. Running an entire net 2000 times becomes computationally expensive as inference time of R-CNN has to run 2000 region proposals after the regions have been generated. Finally, the net uses SVM, which is very different from CNNs they work by basic functions rather kernel methods [16].

The final output of an R-CNN is a bunch of boxes/proposals that will be examined by a classifier and a regressor. Additionally, the R-CNN hypothesis proposes that there is a proposal with high probability of containing a detected object. This hypothesis is used by conditioning the test with more complex neural networks. The overall loss of an R-CNN is a combination of the classification loss (for each box to predict whether the corresponding object) and the regression loss (box refinement). After the RPN, the network yields proposed regions but the resulting boxes that models were trained on are of different sizes. Towards handling unevenly sized input images, most architectures (like VGG, ResNet) use pooling layers [17].

5. Semantic Segmentation

With the increasing popularity of deep learning, semantic segmentation has received significant attention in recent years. Early work on semantic segmentation adopted a two-step pipeline, in which a classifier is first trained to classify the object and then a multi-classification method is applied to assign class labels to each pixel. Some of the early foundations of semantic segmentation were explicitly conditional random fields. The recent success of convolutional neural networks (CNNs) in image classification tasks has led to great progress in visual recognition tasks, such as image classification and object detection. Fully Convolutional Networks (FCNs) are designed to take a standard classification network as a starting point, ignoring the fully connected layers and upsampling the feature maps until the dimensions of the input image are restored. This section first discusses fully convolutional networks (FCNs) and then goes on to discuss a popular architecture for biomedical image segmentation known as U-Net.

FCNs rely on an off-the-shelf classifier, such as VGG16, GoogLeNet, or ResNet, as a backbone. Initially, the input image is preprocessed and fed into the FCN, which extracts the feature maps from the last layer of the classifier. Because each pixel is classified independently, the task is converted to a per-pixel problem. The classifier is initialized with the pre-trained model of the input image and trained with pixelwise annotations. Since the resolution of the feature maps diminishes successively from the input image, the high-resolution pixels are coarsely segmented. To recover the input image size without losing spatial information, skip connections are introduced. By providing help from higher-resolution layers, this kind of architecture emphasizes both object categories and pixel localization.

5.1 Fully Convolutional Networks (FCNs)

Fully Convolutional Networks (FCNs) are convolutional networks where all the pixels have scoring, and are trained end-to-end, pixels-to-pixels. FCNs have been applied successfully to a range of dense prediction problems including image segmentation, diseased plant detection, demosaicking, depth prediction, etc. In the context of image segmentation tasks, there are regions for which the output produces either a different class or a low score at the boundary. To address this problem, the incorporation of dense Conditional Random Fields (CRF), a structured prediction model, can be used [18]. In FCN architecture, a classical encoder-decoder structure is utilized for pixel-wise feature extraction and classification. Downsampling is implemented in the encoder using pooling and strided convolutions. Upsampling gradually recovers spatial resolution in the decoder and is achieved by unpooling layers, as in deconvolutional networks, or learned bilinear interpolation [19]. Skip connections from the encoder to the decoder resulting in a U-Net-like architecture enable the combination of low resolution and high resolution feature maps and retain spatial information during decoding. This coarse-to-fine design pattern has proven useful in many segmentation tasks and is encouraged to be used. FCN models that are pre-trained on ImageNet can be utilized and fine-tuned on FuCo dataset to lessen the burden of training segmentation models from scratch.

5.2 U-Net Architecture

U-Net is one of the most widely used fully convolutional network architectures for semantic segmentation [20]. From its first introduction in 2015 as a deep learning framework for biomedical image segmentation, it has been developed and applied both within and outside the area of biomedicine. Its particular attribute is the U-shaped design with contracting and expansive paths, where feature maps from the contracting paths are concatenated to the corresponding expansive paths for informing about the high-resolution image details.

The architecture of U-Net is illustrated in the (Figure 2). It follows the general idea of fully convolutional networks, but it is implemented particularly to be used for segmentation problems. There are two conventional components of the U-Net architecture: the contracting path (the left) and the expansive path (the right). The basic building blocks of the contracting part are convolution and max-pooling operations, whereas a symmetric up-convolution operation is exploited for up-sampling in the expansive path. The layers in the contracting paths act as a contextual information extractor, in which the spatial information in the image is lost, resulting in low-resolution feature maps [21]. The up-sampling operations of the expansive paths recover uncompacted feature maps with a high-resolution image representation, in which the segmentation is performed. The concatenation from the contracting path to the expansive path facilitates the localization and improves the segmentation accuracy because the detailed features from the high-resolution image representation are combined to the coarser feature maps.

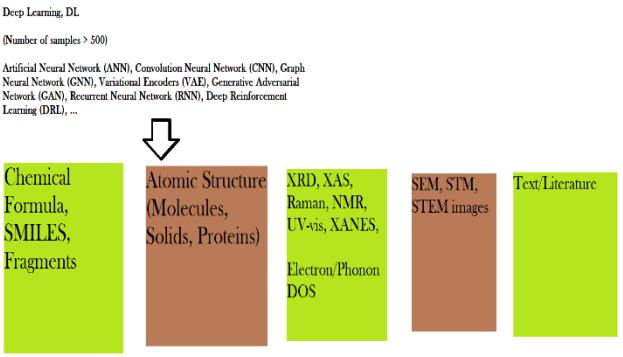

Figure 2. Artificial Intelligence, AI

6. Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) have become a groundbreaking concept in computer vision, particularly in the domain of image generation. Introduced by Ian Goodfellow and colleagues in 2014, GANs consist of two neural networks, a generator and a discriminator, engaged in a competitive game. The generator creates fake images, while the discriminator distinguishes between real and fake images. Through adversarial training, GANs learn to generate increasingly realistic images, making them a powerful tool for creative applications in computer vision.

Deep learning has revolutionized computer vision, enabling machines to excel in image recognition, understanding, and processing tasks traditionally performed by humans. Inspired by the human visual perception system, researchers have developed layered neural network architectures, with Convolutional Neural Networks (CNNs) at the forefront. CNNs have demonstrated superior performance in image classification tasks compared to traditional handcrafted feature-based models. Numerous innovative CNN architectures, such as AlexNet, ResNet, GoogLeNet, and VGGNet, have fueled advances in visual recognition and processing techniques across various practical applications [22].

Alongside breakthroughs in image understanding and processing, efforts have been made to explore different computer vision applications using deep learning models [23]. One such emerging application, GANs, has gained immense popularity in recent years. Generative models learn the distribution of training data from the training set and generate new instances from that learned distribution, while discriminative models learn the boundary between classes.

6.1 Introduction to GANs

Generative Adversarial Networks (GANs) were first proposed in 2014 by a team of researchers at the University of Montreal, including Ian Goodfellow, Yoshua Bengio, and others, and were published in a paper, "Generative Adversarial Networks." In this work, GANs were introduced as an innovative and powerful tool for generative modeling, a subfield of machine learning. Unlike other approaches, GANs are based on game theory and consist of two neural networks that work against each other. Then, one network, the generator (G), aims to produce fake data samples as similar as possible to the real data samples in a dataset, while the other one, the discriminator (D), recognizes the fake samples generated by G and classifies them as "fake." The two networks compete with each other: if D gets better at classifying the samples as "real" or "fake", then G must improve to produce better samples. If G gets better at generating realistic samples, then D must enhance its ability to distinguish between the real and fake samples. Ultimately, if the GAN model is trained properly, it finds a Nash equilibrium, and G produces samples that are indistinguishable from the samples in the dataset, meaning G can match the real data distribution Pdata [23].

GANs have unique characteristics. To begin with, GANs focus exclusively on modeling the underlying probability distribution of a dataset. This allows them to generate samples with the same distribution as the dataset. Furthermore, GANs do this in a non-parametric fashion. Instead of estimating a set of parameters (like the mean and variance of a multivariate Gaussian), GANs attempt to "learn" a generative model that can create new data points. This model could be viewed as the function mapping a variable set of points (in a low-dimensional space) to the data points in the dataset (in a high-dimensional space) [22]. It is also important to mention that GANs are inherently robust to the curse of dimensionality. GANs allow modeling complex distributions by providing the generator network with a sample from a simple, well-known distribution, e.g., a uniform distribution over a hypersphere or the standard normal distribution.

6.2 Applications in Image Generation

One of the important applications of GANs is the image generation of facial images from sketches. Hwang and Ouhyoung, in 1998, proposed an image generation model that generates facial images from arbitrary input sketches [22]. At first, an expression for the input sketch was estimated using a sketch-to-image model. This expression was smashed with a constrained expression of the GAN model to create the final facial image without any loss of semantic information. However, it sometimes deformed the drawn contour line. In 2017, the image generation model from input facial sketches was resumed by a GAN-conditional model (cGAN). The proposed model has a fully connected GAN model having sketch features as input and facial images as output [23]. In 2020, a facial image generation model from input sketches with kernel-based texture refinement was proposed. The input sketch was converted to tone control with rough texture attributes filtered by various sizes. The formal expression was created through image-to-image conversion for concealing application, and finally, kernel-based texturing was incorporated into the generated image. In this case, only single facial sketches were required.

Another application of GANs for image generation is generating facial images from contour images. Cho and others presented a color transformation model from the single contour image to two separate models, one illuminating a face with certain lighting directions and the other showing the shaven face (for bearded images, they also required to input the contour of lower faces). In the case of a color-rendered contour, a two-stream CM for illuminating was introduced. Note that the two facial images rendered under novel colors from contour images could be generated simultaneously in this case. Cho et al. also modeled the fashion data. The proposed model creates a

variety of fashion images that contain different matching cloth combinations between the input clothing and the background. The system integrated cGAN and transferability of domain adaptation by designing two generators, one for attempting each clothing selection list as background and the other for generating images using data for fashion-type clothing event transformations.

7. Transfer Learning in Computer Vision

Transfer learning is a machine learning methodology in which a model trained on one problem (source domain) is reused as the initial starting point on a second, different problem (target domain). The source domain is defined by a particular feature space and a probability distribution over the domain's instances. The domain's feature space consists of the corresponding set of features that are measurable for the instances within that domain. The target domain is the one for which it is desired to learn a new task; it has its own feature space and probability distribution over the instances that is intended to learn a new task [24].

Transfer learning is widely categorized from four perspectives:

1. Availability of labeled data: Transfer learning problems can occur in two situations based on whether the labeled data is available for the source domain and/or the target domain. According to this perspective and with respect to the availability of labeled data, transfer learning can be categorized into three groups:

- Inductive transfer learning.
- Unsupervised transfer learning.
- Transductive transfer learning.

2. Similarity of source and target feature spaces: Transfer learning problems can occur in two situations based on whether the source and target domains have some similarities and/or dissimilarities in feature space. According to this perspective and with respect to the similarity/dissimilarity of source and target feature spaces, transfer learning can be categorized into two groups:

- Homogeneous transfer learning
- Heterogeneous transfer learning.

3. Similarity of source and target conditional probability distributions: Transfer learning problems can occur in two situations based on whether the conditional probability distributions of source and target domains are the same. According to this perspective and with respect to the similarity/dissimilarity of source and target conditional probability distributions, transfer learning can be categorized into two groups:

- Covariate shift
- Prior shift.

Transfer learning is a collection of methodologies that aim to effectively leverage previously acquired knowledge from one or several tasks (source) to aid target tasks [25]. On the one hand, these methodologies may be required due to the scarcity of data (and possibly labels) for the target task. On the other hand, handling the target task may be computationally costly and/or time-consuming. Transfer learning methodologies may be used at different steps of the overall learning process, including model selection, data selection, and data generation. It can involve based approaches (i.e., acting on the physical components of the applied learning algorithm) and unbased approaches (i.e., acting on the data at the input of the learning algorithm).

8. Deep Reinforcement Learning for Vision Tasks

Deep reinforcement learning (DRL) combines deep learning and reinforcement learning (RL) into a single, cohesive architecture. It consists of a representation learning strategy, leveraging deep learning to learn functional representations, often neural networks, and an action learning strategy to solve the control problem, often using RL techniques [26]. The representation learning strategy is what distinguishes DRL from previous approaches that used hand-crafted features with either linear or non-linear classifiers. It is the first to convincingly demonstrate that RL agents could be trained on raw, high-dimensional observations like pixels, with no further feature engineering done by a human.

More importantly, DRL was able to do so in very complex environments, solely based on a reward signal, thus closing the loop between perception and decision making. Applications span from relatively simple domains like Atari games and board games to physically realistic 3D simulated environments 1. DRL algorithms have been applied to a wide range of problems. Not only in simulation, such as robotics, where control policies for robots can now be learned directly from camera inputs in the real world.

9. Future Directions and Emerging Trends

The field of computer vision technologies, encompassing areas such as object recognition, image classification, image analysis, occurrence estimation and semantic segmentation, has recently experienced a resurgence in popularity, widespread activity and high expectation. Much of the current interest in this area stems from the major and

enthusiastic successes of Deep Learning methods, particularly multilayered Convolutional Neural Networks (CNNs) and similar representations, on large, challenging datasets, such as ImageNet, and on influential benchmark problems, such as PASCAL VOC and MS COCO 1. With a growing emphasis on Deep Learning methods, a question arises; what is the nature of the future in computer vision? Where will this field of study go from here?

With this question in mind, an attempt is made to describe and inquire into some major and influential emerging trends or potential directions in computer vision technologies and systems. As computer vision tasks and datasets become larger, more challenging and closer to everyday life, the solutions to these tasks and the technologies, architectures, algorithms and approaches that underpin them are described. Readers are invited to parse their own thoughts on these potential areas or trends as they consider the future of computer vision 2.

10. Conclusion

In this paper, we have proposed a new scheme for switching the primary path and an alternative path within the SCTP protocol for MTs that are moving at various speeds between networks. The proposed scheme achieves a better overall performance than other existing schemes as shown by performance simulation. This is due to the fact that the proposed scheme utilizes the performance trade-off relationship between RTT and *cwnd* and performs switches between the current primary path and the alternative path according to RTT, as well as the velocity of movement of MT.

References

[1] J. Ruiz-del-Solar, P. Loncomilla, and N. Soto, "A survey on deep learning methods for robot vision," *arXiv:1803.10862*, pp. 1-43, 2018.

[2] R. K. Sinha, R. Pandey, and R. Pattnaik, "Deep learning for computer vision tasks: a review," in *2017 International Conference on Intelligent Computing and Control (I2C2)* 2018: arXiv:1804.03928, p. 5.

[3] P. Shah, V. Bakrola, and S. Pati, "Optimal approach for image recognition using deep convolutional architecture," in *Recent Findings in Intelligent Computing Techniques*, 2018: Springer, pp. 535-545. doi: https://doi.org/10.1007/978-981-10-8633-5_53

[4] A. Luckow, M. Cook, N. Ashcraft, E. Weill, E. Djerekarov, and B. Vorster, "Deep learning in the automotive industry: Applications and tools," in *2016 IEEE International Conference on Big Data (Big Data)*, Washington, DC, USA, 2016: IEEE, pp. 3759-3768, doi: https://doi.org/10.1109/BigData.2016.7841045

[5] K. O'shea and R. Nash, "An introduction to convolutional neural networks," *arXiv:1511.08458*, pp. 1-11, 2015.

[6] A. Celeghin *et al.*, "Convolutional neural networks for vision neuroscience: significance, developments, and outstanding issues," *Frontiers in Computational Neuroscience*, vol. 17, p. 1153572, 2023, doi: https://doi.org/10.3389/fncom.2023.1153572

[7] F. Iandola, "Exploring the design space of deep convolutional neural networks at large scale," PhD Thesis, University of California, Berkeley, 2016.

[8] G. Stathopoulos, "Application of Recognition Input Squinting and Error-Correcting Output Coding to Convolutional Neural Networks," MSc Thesis, Concordia University, 2011.

[9] G. Kobayashi and H. Shouno, "Interpretation of ResNet by Visualization of the Preferred Stimulus in Receptive Fields," in *Advances in Parallel & Distributed Processing, and Applications*, 2021: Springer, pp. 769-779, doi: https://doi.org/10.1007/978-3-030-69984-0_56

[10] H. Qassim, D. Feinzimer, and A. Verma, "Residual squeeze vgg16," *arXiv:1705.03004*, p. 11, 2017.

[11] I. Cosmin Duta, L. Liu, F. Zhu, and L. Shao, "Improved residual networks for image and video recognition," *arXiv e-prints*, pp. 1-22, 2020, doi: https://ui.adsabs.harvard.edu/link_gateway/2020arXiv200404989C/doi:10.48550/arXiv.2004.04989

[12] Y. He, "GoogLe2Net: Going transverse with convolutions," *arXiv:2301.00424*, p. 33, 2023.

[13] L. Liu *et al.*, "Deep learning for generic object detection: A survey," *International Journal of Computer Vision*, vol. 128, pp. 261-318, 2020, doi: https://doi.org/10.1007/s11263-019-01247-4

[14] K. Singh Chahal and K. Dey, "A Survey of Modern Object Detection Literature using Deep Learning," *arXiv e-prints*, p. 15, 2018, doi: https://ui.adsabs.harvard.edu/link_gateway/2018arXiv180807256S/doi:10.48550/arXiv.1808.07256

[15] X. Zhang, Y. Gao, F. Ye, Q. Liu, and K. Zhang, "An approach to improve SSD through skip connection of multiscale feature maps," *Computational Intelligence and Neuroscience*, vol. 2020, pp. 1-13, 2020, doi: https://doi.org/10.1155/2020/2936920

[16] R. Simhambhatla, K. Okiah, S. Kuchkula, and R. Slater, "Self-driving cars: Evaluation of deep learning techniques for object detection in different driving conditions," *SMU Data Science Review*, vol. 2, no. 1, p. 23, 2019.

[17] F. Sultana, A. Sufian, and P. Dutta, "A review of object detection models based on convolutional neural network," in *Intelligent computing: image processing based applications*: Springer, Singapore, 2020, pp. 1-16.

[18] J. Wang, "Semantic Image Segmentation via a Dense Parallel Network," PhD Thesis, Syracuse University, 2019.

[19] B. Shuai, T. Liu, and G. Wang, "Improving fully convolution network for semantic segmentation," *arXiv:1611.08986*, p. 9, 2016.
[20] V. Ummadi, "U-Net and its variants for Medical Image Segmentation: A short review," *arXiv:2204.08470*, p. 5, 2022.
[21] L. Chen, P. Bentley, K. Mori, K. Misawa, M. Fujiwara, and D. Rueckert, "DRINet for medical image segmentation," *IEEE Transactions on Medical Imaging*, vol. 37, no. 11, pp. 2453-2462, 2018, doi: https://doi.org/10.1109/TMI.2018.2835303
[22] L. Jin, F. Tan, and S. Jiang, "Generative adversarial network technologies and applications in computer vision," *Computational Intelligence and Neuroscience*, vol. 2020, pp. 1-17, 2020, doi: https://doi.org/10.1155/2020/1459107
[23] H. Navidan *et al.*, "Generative Adversarial Networks (GANs) in networking: A comprehensive survey & evaluation," *Computer Networks*, vol. 194, p. 108149, 2021, doi: https://doi.org/10.1016/j.comnet.2021.108149
[24] A. Farahani, B. Pourshojae, K. Rasheed, and H. R. Arabnia, "A concise review of transfer learning," in *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, Las Vegas, NV, USA, 2020: IEEE, pp. 344-351, doi: https://doi.org/10.1109/CSCI51800.2020.00065
[25] A. Tormos, D. Garcia-Gasulla, V. Gimenez-Abalos, and S. Alvarez-Napagao, "When & How to Transfer with Transfer Learning," *arXiv:2211.04347*, pp. 1-10, 2022.
[26] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A brief survey of deep reinforcement learning," *IEEE Signal Processing Magazine*, pp. 1-16, 2017.
[27] K. Huang, Y. Teng, C. Yang, and Y. Wang, "From Pixels to Principles: A Decade of Progress and Landscape in Trustworthy Computer Vision," *Science and Engineering Ethics*, vol. 30, p. 26, 2024, doi: https://doi.org/10.1007/s11948-024-00480-6
[28] S. Goree, G. Appleby, D. Crandall, and N. Su, "Attention is All They Need: Exploring the Media Archaeology of the Computer Vision Research Paper," *arXiv:2209.11200*, pp. 1-25, 2022.

How to cite this article
A. S. Mahmoud and S. H. Ahmed, "Deep Learning for Computer Vision: Innovations in Image Recognition and Processing Techniques," *CyberSystem Journal*, vol. 1, no. 1, pp. 23-32, 2024. doi: 10.57238/n65d0p57

Access this article online