DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

Architecture of Deep Learning and Its Applications

Afrah Salman Dawood[1], Zena Mohammed Faris[2]
[1]*Communications Engineering Department, University of Technology, Baghdad, Iraq*
[2]*Ministry of Youth and Sport, Directorate General of Scientific Welfare, Iraq*
[1]*afrah.s.dawood@uotechnology.edu.iq*, [2]*zeenamhmmd@gmail.com*

Abstract— *Recently, Deep Learning (DL) has accomplished enormous prosperity in various areas, like natural language processing (NLP), image processing, different medical issues and computer vision. Both Machine Learning (ML) and DL as compared to traditional methods, can learn and make better and enhanced use of datasets for feature extraction. This paper is divided into three parts. The first part introduces a detailed information about different characteristics and learning types in terms of learning problems, hybrid learning problems, statistical inference and learning techniques; besides to an exhausted historical background about feature learning and DL. The second part is about the major architectures of DL with mathematical equations and clarified examples. These architectures include Autoencoders (AEs), Generative Adversarial Networks (GANs), Deep Belief Networks (DBNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and Recursive Neural Networks. The third part of this work represents an overview with detailed explanation about different applications and use-cases. Finally, the fourth part is about hardware/ software tools used with DL.*

Index Terms— *Deep learning, Machine Learning, Neural Network, Network architecture.*

I. INTRODUCTION

The broad term artificial intelligence (AI) includes any technique that is able to mimic human behavior. ML is a division of AI and it includes the use of data and algorithms that simulate the way humans think. Going further is DL, a branch of ML, that simulates the behavior of the human brain using neural networks with the ability to process massive amounts of data, continuously learn from it and then provide highly precise results.

DL (utilizing either deep architecture of learning or progressive learning approaches) is an ML technique, developed essentially by Geoffrey E. Hinton and his co-authors from 2006 [1][2], where highly considerable Neural Networks (NN) learn from the major quantity of data and convey highly accurate outputs. The more the data, the better the DL model learns and the more accurate the output. Generally, learning is a process for evaluating parameters of the model so that the learned model (i.e., algorithm) can perform a specific task [3]. Multiple layers are used in deep learning between input and output layers to symbolize the abstraction of non-linear information processing units with hierarchical architectures to construct a computational model or classification. Some researchers have depicted DL as a global learning tactic that is able to solve almost all kinds of issues in various application domains (i.e., DL is not task specific) [4]. DL is sometimes called universal learning [5] since it can be utilized to nearly any application domain (see *Fig. 1*). Additionally, it doesn't need the plan of highlights early. Other important features are the automatically learned features and the high scalability where a DL model can identify, classify and

Received 20/April/2022; Accepted 15/May/2022
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

analyze structured data, images, text or sound. In a 2015 paper, Microsoft described a network known as ResNet [6]. It consists of 1202 layers and is usually executed at a supercomputing scale. There are many useful uses of DL.

1. In the shortfall of a human master (route on Mars).
2. When the person is incapable of showing his experience (speech recognition).
3. When the answer for the issue changes extra time (following, climate expectation, cost forecast).
4. When arrangements should be adjusted to the specific cases (biometrics).
5. When the issue size is excessively huge for our restricted thinking capacities (managing ads on social media).

FIG. 1. WHERE TO USE DL.

Over more than 100 years ago, many methods for data representation learning have been proposed. K. Pearson in 1901 (referred to Table I below) proposed principal component analysis (PCA) to learn low dimensional representations of data with a linear projection; while linear discriminant analysis (LDA) was proposed by R. Fisher in 1936 [7]. Generally, feature learning [8] is the technique that allows the system to automatically detect the features from a set of row data and use them for a specific task. Although in practical terms a DL is just a subset of ML, the main distinction between traditional ML and DL is in how features are extracted. Manual features are used by conventional ML tactics by placing different characteristic deposition algorithms including Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), Local Binary Pattern (LBP), Empirical mode decomposition (EMD) for speech analysis, and many more. Table II below, illustrates the relationship between different data representation learning algorithms. Features in DL are learned in an automatic manner and are symbolized hierarchically in multiple levels. This is the powerful stage of DL versus traditional ML tactics. In abridgment for the above explanation ML needs some guidance while DL seems to have its own brain. Recently, the use of the concept "Shallow Learning" expanded. In fact, it is not standardized where it refers to traditional ML [23][24].

Despite of its benefits and uses, DL has a number of challenges, the following are [25] [26] [27]:

1. Deep Learning for Big Data Analytics: DL can deal with key criteria of the big data challenge, including volume, velocity, diversity, and validity.
2. Mobile intelligence, FPGAs, and other energy-efficient solutions for special-purpose devices.
3. Multi-tasking and transfer learning (generalization) or multi-module learning are examples of multi-tasking and transfer learning. This entails combining knowledge from diverse fields or models.
4. DL methods' scalability: Most large-scale problems are solved using a High-Performance Computing (HPC) system (supercomputing, clustering, and cloud computing), which has enormous prospects for data-intensive commercial computing. While data grows in velocity, diversity, veracity, and volume, scaling compute performance using business level servers and storage to keep up becomes increasingly difficult.

Received 20/April/2022; Accepted 15/May/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

5. Capability to produce data, which is critical when source data is unavailable for system learning (especially for computer vision tasks such as inverse graphics).

TABLE I. HISTORICAL BACKGROUND OF LEARNING FEATURES

Year	Action
1957	found the perceptron, a two-layer NN for parallel grouping [9].
1982	The idea that RNNs are extraordinary as they permit activity over a succession of vectors after some time [10].
1985	● Described an overall parallel inquiry strategy, in view of measurable mechanics, and showed how it prompts an overall learning rule for altering the association qualities to fuse information about the undertaking space in a proficient manner. ● Described some simple examples in which the learning algorithm creates internal representations [11]
1986	Presented that the Bp calculation can create helpful inner portrayals of information in secret layers of NNs [12].
1998	Applied an inclination-based learning calculation to CNNs and got successes for the written by hand digit characterization issue [13].
2000	Proposed the most useful model in RNN called Long Short-Term Memory (LSTM) [14].
2003	Discovered that LSTM, compounded by "peephole connections" from the inside cells to the multiplicative gates, is capable of learning the smooth difference between series of spikes spaced either 50- or 49-time steps away without assistance of any short training samples. Their LSTM variation additionally figures out how to create stable surges of unequivocally planned spikes and other exceptionally nonlinear occasional examples [15].
2006	DNNs have been successfully applied to reduce dimensionality and the concept of "deep learning" has been proposed [1][2].
2015	Presented Visual Geometry Group (VGG) architecture which shows that the profundity of an organization is a basic part to accomplish better acknowledgment or characterization precision in CNNs [16].
2016	Developed Residual Network (ResNet) with the goal of planning super profound organizations that didn't experience the vanishing effects of the evaporating slope issue that ancestors had and gain the accuracy from increased depth (till 152 layers - 8× deeper than VGG nets) likewise present investigation on CIFAR-10 with 100 and 1000 layers [6]. Many alternatives to this network have been introduced in that time.
2017	Developed Densely Connected Network (DenseNet) which associates each layer to each and every layer in a feed-forward design. For each layer, the component guides of all previous layers are utilized as sources of info, and its own element maps are utilized as contributions to every single resulting layer [17].
2018	A novel meta-learning algorithm was proposed to learn assigning weights to preparing models in light of their inclination headings. The technique plays out a meta slope plummet step on the present scaled down group model loads (which are instated from nothing) to limit the misfortune on a clean fair approval set [18].
2019	Presented SAUCIE, a DL network that joins parallelization and versatility presented by NNs, with the profound portrayal of information that can be advanced by them to perform many single-cell information examination assignments. It permits data to be gotten from the inner layers of the organization, which gives extra unthinking comprehension that can be utilized to additional tune information investigation [19].
2019	Two various element learning systems for the combination of hyperspectral thermal infrared (HTIR) and visible remote sensing data have been presented. First, a Deep Convolutional Neural Network (DCNN)- Support Vector Machine (SVM) was used on features with datasets to supply the class labels. Then, a shallow feature scheme utilized to approve the outcomes with other learning procedures [20].
2020	Introduced PyTorch3D library. It includes a fast, modular differentiable 3D DL operator's renderer for meshes and point clouds, enabling analysis-by-synthesis tactics [21].

Received 20/April/2022; Accepted 15/May/2022
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

2021 A deep-learning-empowered breast cancer auxiliary diagnosis blueprint has been developed for remote e-health propped by 5G technology. First, breast pathology images are received from the hospital through 5G communication, and a DL scheme is subjected based on the Inception-v3 network to transfer learning to gain a diagnostic model [22].

TABLE II. TYPES OF FEATURE LEARNING

Learning Problems	**Hybrid Learning Problems**	**Statistical Inference**	**Learning Techniques**
Supervised Learning $traning\ data = input\ vector + target\ vector$ [28]	Semi-Supervised Learning $traning\ data = few\ labled\ examples + large\ collection\ fo\ unlabled\ examples$ [29], [30]	Inductive Learning $model = learned\ from\ past\ examples$ $induction\ conclusion = draw\ general\ future\ conclusions\ from\ past\ observations$ induction is bottom-up reasoning uses available data to find output [31]	Multi-Task Learning $multiple\ related\ models = borrow + shared\ statistical\ strength\ from\ large\ data\ task\ with\ little\ data$ [31]
Unsupervised Learning $data\ relationships = learned\ by\ clustering\ density\ estimation$ [32]	Self-Supervised Learning $pretext\ learning\ task = supervised\ learning\ algorithm\ solves\ unsupervised\ learning\ problem = predicting\ some\ hidden\ portion\ of\ data$ [33]	Deductive Inference $deduction \propto \frac{1}{induction}$ Deduction is top-down reasoning that requires all premises met before determining the conclusion [34]	Active Learning $training\ examples{=}adaptively\ collected\ by\ querying\ an\ oracle\ to\ request\ labels\ for\ new\ points$ [35]
			Online Learning used with streaming data $adaptive\ model = observations + probability\ distributio$ [31]
Reinforcement Learning $goal = map\ situations\ to\ actions + feedback$ [36]	Multi-Instance Learning $bag = group\ of\ instances$ $goal\ of\ learning = how\ class\ infered\ from\ instances$ [37]	Transductive Learning $transduction = derving\ values\ of\ unknown\ function\ for\ point\ of\ interest\ form\ given\ data$ $general\ rules = learned\ from\ specific\ examples$ [38]	Transfer Learning $factors\ for\ different\ tasks = p1\ variations \propto p2\ variations\ needed\ to\ be\ captured\ for\ training$ Where $p1\ and\ p2$ are tasks' factors [39]
			Ensemble Learning $composite\ predictor = developing\ a\ populatio\ of\ base\ learners\ from\ training\ data + combining\ them$ [40]

Received 20/April/2022; Accepted 15/May/2022
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

II. MAJOR ARCHITECTURES OF DEEP LEARNING

Basically, NN is a computational model derived from the architecture of animals' brain cells where many simple units (also called node, neuron, perceptron) work in parallel to perform a specific behavior without a centralized control unit. *Fig. 2* illustrates the working model of a basic nonlinear perceptron in detail, where x_1 , x_2, $\cdots$ x_n are input signals, w_{k1}, w_{k2}, $\cdots w_{kn}$ are learning weights, φ_j is the step (activation) function which is chosen to be the sigmoid function ($\varphi(j) = \frac{1}{1+e^{-z}}$) for the current status and derivatives. The output of this function will have the range [0,1], which is a similar result to the strategic relapse function. The network architecture of NN is composed mainly of the number of neurons, layers, and types of associations among layers. The perceptron functioning can be described in the following equation:

$$y_k = \varphi(\sum_{i=1}^{n} x_i w_i + b_k) \quad (1)$$

Where b_k is the bias (input value doesn't influence its worth; shifts choice limit away from beginning) which is added as a linear combiner of outputs of the summation. $x_i w_i$is a dot product. $n$is the number of inputs to the perceptron. The basic most understood NN is the feed-forward multilayer NN which consists of a single I/O layer and one or multi hidden layers as shown in *Fig. 3*. The weight values on the associations between the layers are the way NNs encode the gained data separated from the crude preparation information. Hidden layers are the way to permit NNs to show nonlinear capacities. The perceptron is by and large prepared by a learning calculation named Bp learning which is very important for reducing errors in NN. This algorithm uses gradient descent on the weights of the associations in NN for limiting the blunder on the result of the organization. Historically, Bp has been considered slow because of the local minima problem, although latent improvements in computational powered by parallelism and Graphics Processing Units (GPUs) have refreshed concern in NNs. The mathematical expression of a multilayer perceptron is as follows [3]:

$$y = \varphi\ (w_l \ldots \varphi\ (w_2 \varphi\ (\ w_1 x + b_1) + b_2) \ldots + b_L) \quad (2)$$

Where, L is the number of layers, y is the network output. $h_l = \varphi_l(W_l^T h_{l-1} + b_l)$ is the output form hidden layers for the t^{th}round of training. If we have only two layers:

$$y = \varphi\ (w_2 \varphi\ (\ w_1 x + b_1) + b_2) \quad (3)$$

Then the mathematical expression will be as follows:

$$y = f(g(x)) \quad (4)$$

According to the chain rule [41], the derivative will be:

$$\frac{\partial y}{\partial x} = \frac{\partial f(x)}{\partial x} = f'(g(x)).\, g'(x) \quad (5)$$

Where $g(x)$ represents the inner function ($w_1 x + b_1$) and $f(x)$ (i.e. the composite function) represents $\varphi\ (w_2 \varphi\ (\ w_1 x + b_1) + b_2)$ the stochastic gradient descent (SGD) with the momentum method (its main purpose is to utilize moving normal of the slope rather than the present genuine worth simply to forestall the organization form stacking in the local minima) formulas [42] [43]:

$$\tilde{y}_i = F(\theta, x_i) \quad (6)$$

$$\theta = \theta - \frac{\eta.1}{L \sum_{i=1}^{L} \partial \varepsilon(y_i, \tilde{y}_i)/\partial \theta} \quad (7)$$

$$v_t = \gamma v_{t-1} - \eta \Delta F(\theta_{t-1}) \quad (8)$$

$$\theta_t = \theta_{t-1} + v_t \quad (9)$$

Where ε is the loss function (explained later in this section), η is the learning rate, X, y is the dataset, $F(\theta, x)$is the model to be optimized, γ is the momentum which is in the range (0,1] and usually set to 0.5

Received 20/April/2022; Accepted 15/May/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

until the initial training stabilized then increased to 0.9 or higher because setting it to higher values overshoots its minimum and make the network unstable [42], $\Delta F(\theta_{t-1})$ is the gradient, and v_t is the velocity vector of the t^{th} round of training. If we replace the above mathematical expression with the perceptron variables ($y \rightarrow h_l, x \rightarrow w_l, b_l, \Delta F(\theta_{t-1}) \rightarrow \delta$) then we will have the following arrangement of the Bp:
The gradient for the present layer:

$$\frac{\partial \varepsilon(y,\tilde{y})}{\partial w_l} = \frac{\partial \varepsilon(y,\tilde{y})}{\partial h_l}\frac{\partial h_l}{\partial w_l} = \delta \frac{\partial h_l}{\partial w_l} \tag{10}$$

$$\frac{\partial \varepsilon(y,\tilde{y})}{\partial b_l} = \frac{\partial \varepsilon(y,\tilde{y})}{\partial h_l}\frac{\partial h_l}{\partial b_l} = \delta \frac{\partial h_l}{\partial b_l} \tag{11}$$

Applying the gradient ($\frac{\partial \varepsilon(y,\tilde{y})}{\partial w_l}$) and ($\frac{\partial \varepsilon(y,\tilde{y})}{\partial b_l}$) descent to the lower layer (i.e., back propagate):

$$\delta \leftarrow \frac{\partial \varepsilon(y,\tilde{y})}{\partial h_l}\frac{\partial h_l}{\partial h_{l-1}} = \delta \frac{\partial h_l}{\partial h_{l-1}} \tag{12}$$

The pseudo code of the basic Bp according to above derivation and *Fig. 3* is given in Example 1.

Example 1: Basic Pseudo code for Bp Algorithm

$n_t = network\ topology$, $w_p = weight\ parameters$, $a_f = activation\ function$
$s_t = stop\ condition$, $c_p = compute\ prediction$, $op = actual\ output$

Initialize n_t , w_p , a_f
While (s_t not reached)
{
For all $items$ do:
{
$c_p = NN_{output}$ (n_t, w_p) //feedforward input to hidden layers
op = observed result associated with $items$ //feedforward hidden layers to output layer
Update w_p based on $items$,c_p, op, a_f //backward output to hidden layers to input layer
}
Check s_t //returns the correctly classified $items$or hit unexpected error
}
Return op

Activation functions (see Table III) [44] are critical part of designing NN and used to propagate the output of one perceptron to the next succeeding one till the output layer; usually, they are one of sigmoid family functions. The choice of activation function in the hidden layer will control how well the network model learns the training dataset. The choice of activation function in the output layer will define the type of predictions the model can make. Below are the most important types of activation functions:

Received 20/April/2022; Accepted 15/May/2022
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

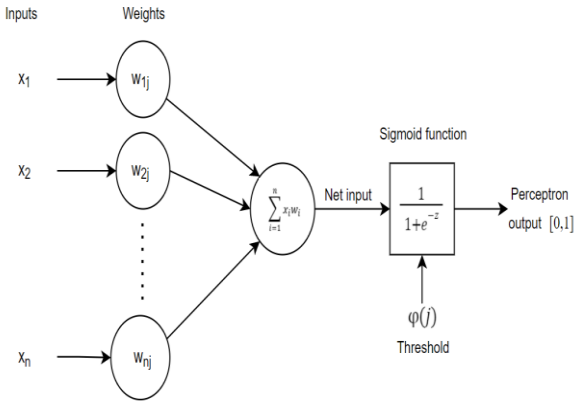

FIG. 2. ARCHITECTURE OF PERCEPTRON.

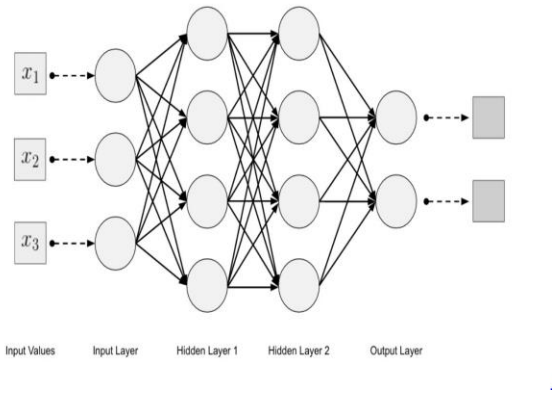

FIG. 3. BASIC FEEDFORWARD ARCHITECTURE.

Activation functions can be different for hidden and output layers (*Fig. 4* and *Fig. 5*) [45]. Regularly, a differentiable nonlinear enactment capability is utilized in the secret layers of an NN. This permits the model to learn more intricate capabilities than an organization prepared utilizing a direct enactment capability. To gain admittance to a lot more extravagant speculation space that would profit from profound portrayals, a non-linearity, or enactment function is needed [46][47]. Probably, three activation functions are considered for use in hidden layers: Rectified Linear Activation (ReLU), Logistic (Sigmoid) and Hyperbolic Tangent (Tanh). For the output layer, there are possibly three types of activation functions which are: Linear, Logistic (Sigmoid) and Softmax.

TABLE III. TYPES OF ACTIVATION FUNCTIONS

Function's Name	**Arithmetic Expression**
Binary Step	$f(x) = 0\ , x < 0\ \ ;\ \ 1, elsewhere$
Linear	$f(x) = bx$, where b is a constant value
Sigmoid	$f(x)\ = \frac{1}{(1 + e^{-x})}$
Tanh	$tanh(x) = 2\ sigmoid\ (2x) - 1 = 2(1 + e^{-2x}) - 1$
Hard Tanh	$f(x) = 1\ , x > 1\ \ ; -1\ , x < -1$
ReLU	$f(x) = max(0, x)$
Leaky ReLU	$f(x) =\ x\ , x >= 0\ \ ;\ \ 0.01x\ , elsewhere$
Softmax	$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}\ \ for\ j = 1,2 \ldots K$
Parameterised ReLU	$f(x)\ =\ x, x >= 0\ ;\ = ax, elsewhere$
Exponential Linear Unit (ELU)	$f(x)\ =\ x,\ \ x >= 0\ ;\ = a(e^{x} - 1), elsewhere$
Softplus	$f(x)\ =\ ln[\ 1\ +\ exp(x)\ ]$

Received 20/April/2022; Accepted 15/May/2022

ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

Swish	$f(x) = x * sigmoid(x) = \frac{x}{(1 - e^{-x})}$

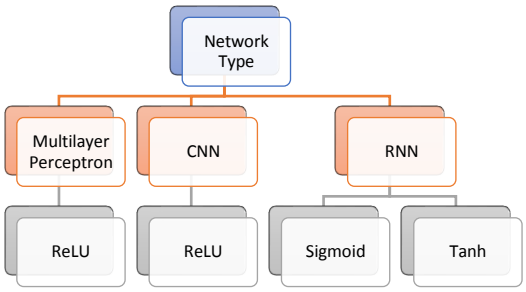

FIG. 4. HOW TO CHOOSE A HIDDEN LAYER ACTIVATION FUNCTION.

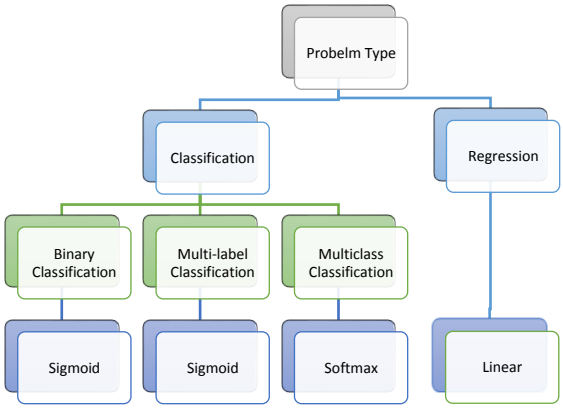

FIG. 5. HOW TO CHOOSE AN OUTPUT LAYER ACTIVATION FUNCTION.

On the other hand, and as feature of the enhancement algorithm, the blunder for the present status of the model should be assessed over and over. This requires the decision of an error function, conventionally called a loss function, that can be utilized to estimate the loss of the model so the weights can be refreshed to diminish the loss on the following evaluation. Loss function is a function for finding the optimal gradients/ coefficients of a machine learning model and it does so by minimizing the error (in other words, assesses reframe training NNs like an optimization issue). The classification of these functions is described in *Fig. 6* below. NN models gain a planning from inputs to outputs from examples and the decision of loss function must match the framing of the specific predictive modeling issue, like grouping or regression. Moreover, the setup of the output layer should likewise be appropriate for the picked loss function. Other names for loss function are cost function, objective function or criterion or error function (i.e., the function we want to minimize) [39].

DL networks are formed from smaller networks and have four major architectures which are [48] Convolutional Neural Networks (CNNs), Unsupervised Pretrained Networks (UPNs), Recurrent Neural Networks, Recursive Neural Networks.

A. Unsupervised Pretrained Networks (UPNs): First of all, UPNs have three particular architectures which are:

- Autoencoders (AEs): Autoencoders were first introduced by G. E. Hinton [44] for data compression and learning useful abstraction of data, is used to return input. Thus, it ought to have an output layer that is able to return input. This means that the activation function must be selected neatly. Besides, the normalization of the range of input values must be such that the shape of output remains the same as input. There are two significant distinct of autoencoders which are compression AEs, where the input of the network should progress through a bottleneck region of the organization earlier being extended once again into the result portrayal, and denoising AEs where the autoencoder is given a debased form (e.g., a few elements are taken out haphazardly) of the information and the organization is compelled to get familiar with the uncorrupted result. Like any DL architecture, AE (*Fig. 7* below) works in layers of neurons, and is trained using backpropagation. The layers are divided into different encoder and decoder layers [49].

Received 20/April/2022; Accepted 15/May/2022

ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

The input is connected to the first encoder layer. In each subsequent layer of the encoder, the number of neurons is reduced till we reach the last encoded layer, which has the least number of neurons, representing the bottleneck features. The input transformed until this layer represents the reduced dimension of the data that is capable of representing the maximal signals in the data. After this layer, the decoder layers are set, in which the number of neurons is increased in each layer, until the output layer which has the same number of neurons as the input. For continuous input, squared error loss could be one of the loss functions for AE, and is given as [50]:

$$L(x, x') = ||x - x'||^2 = ||x - W'(Wx + b) + b'||^2 \quad (13)$$

Where x is the input, W is the weight of the encoder, $W'$is the weight of the decoder, and $b$is the bias of both decoder and encoder. The mean square error for the patch of $n$samples would be as follows:

$$MSE = \frac{1}{n}\sum_1^n L(x, x') \quad (14)$$

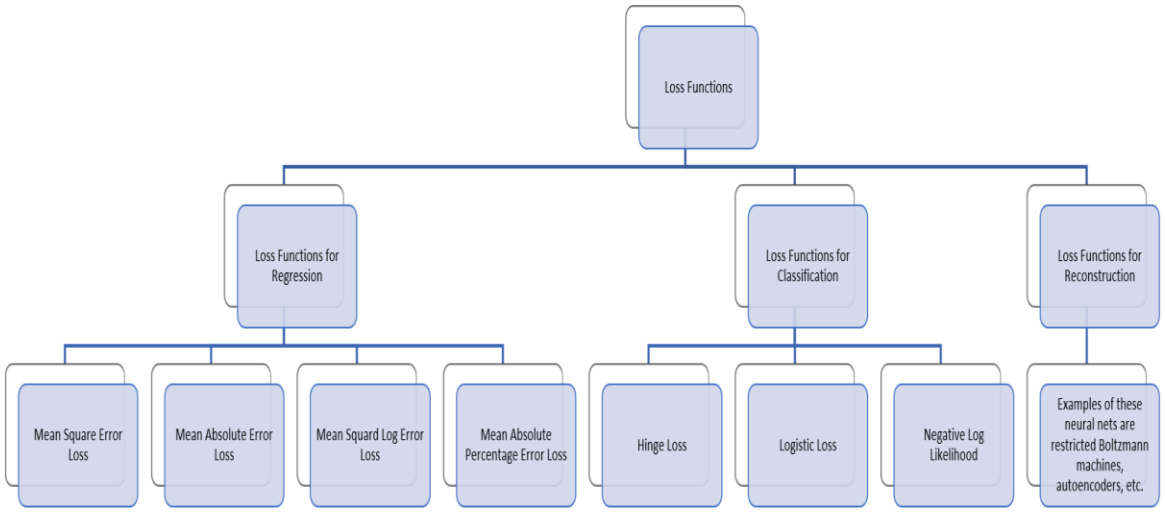

FIG. 6. LOSS FUNCTION'S CLASSIFICATION.

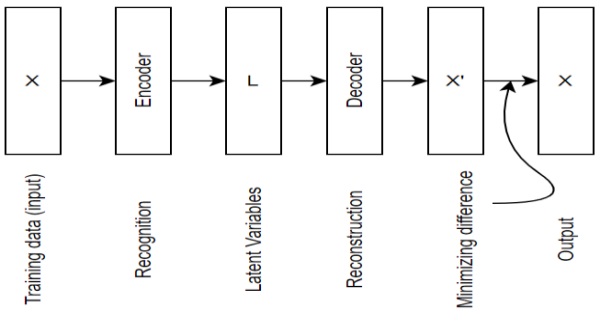

FIG. 7. ARCHITECTURE OF VAE.

- Deep Belief Networks (DBNs): Generally, there are two approaches to understand the multilayer models, directed and undirected views [51]. In the coordinated view, a multi-facet generative model that has endlessly many layers of inert factors is fitted, yet utilizes weight dividing between the higher layers to monitor the quantity of boundaries. In the undirected, "energy-based" view, we fit a somewhat straightforward sort of learning module that just has one layer of idle factors, however at that point we

Received 20/April/2022; Accepted 15/May/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

treat the exercises of the inactive factors as information and fit a similar kind of module again to this new "data". The straightforward learning module utilized in the undirected view is Restricted Boltzmann Machine (RBM) which is another unsupervised DL approximation. The preparation stage can be demonstrated utilizing a two-layer network called RBM [52]. RBM is an energy-based undirected generative model that utilizes a layer of stowed away factors for showing appropriation over apparent factors. One of the most popular DL approaches called DBN is proposed based on this approach. DBNs are mainly formed from layers of RBMs for the pretrain stage and afterward a feed-forward network for the tweak stage. The general architecture of DBN is described in the following *Fig. 8*.

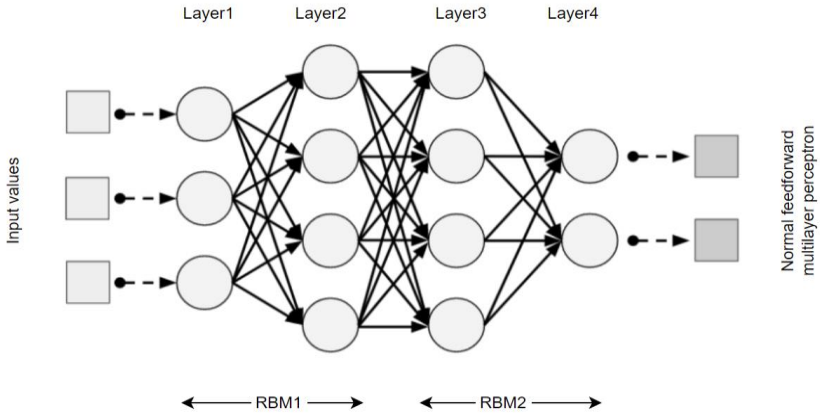

FIG. 8. ARCHITECTURE OF DBN.

In the undirected view, RBMs have a productive rough preparation technique called "contrastive divergence" which makes them proper as building blocks for learning DBNs where each weight w_{ij} in binary RBMs is frequently updated based on the difference between two measured correlations:

$$\Delta w_{ij} \propto (v_i h_j)_{data} - (v_i h_j)_{correlation} \quad (15)$$

The first term of the RHS is the deliberate recurrence of the apparent unit v_i and stowed away unit h_i are together when the noticeable vectors are tests from the preparation set and the conditions of the secret still up in the air by:

$$p(h_j = 1|v, \theta) = \sigma(a_j + \sum_{i=1}^{v}(w_{ij} v_i) \quad (16)$$

Where the conditional distribution $p(h|v, \theta)$is factor, $\theta = (w, b, a)$, b, a are bias terms and $\sigma(x) = (1 + e^{-x})^{-1}$. The subsequent term is the deliberated frequency where both i and j are activated whenever the noticeable vectors are "recreations" of the information vectors and the conditions of hidden objectives are set through stratifying the conditional distribution factor to the reconstructions. The energy of joint configuration is given by:

$$E(v|h, \theta) = -\sum_{i=1}^{V}\sum_{j=1}^{H} w_{ij} v_i h_j - \sum_{i=1}^{V} b_i v_i - \sum_{j=1}^{H} a_j h_j \quad (17)$$

Where $V, H$are quantities of noticeable and secret units, sequentially. Subsequent to learning the loads in a RBM module, when driven by real data, the conditions of the hidden units are utilized as the data for training one more module of a similar kind. This interaction can be rehashed to learn however many layers of highlights as we need.

In the coordinated view, the learning calculation is more convoluted to clarify, yet a lot simpler to justify [1]. Here, the sigmoid function comprises various layers of binary stochastic units where the upper "hidden" layers address binary features and the base, "visible", layer addresses a binary data vector. the

Received 20/April/2022; Accepted 15/May/2022

ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

issue of changing weights on the hierarchical associations with the goal that the model is bound to create the double preparation information on its apparent units. The issue of changing weights on the hierarchical associations so the model is bound to create the binary training information on its visible units can be discussed with two approaches:

1. Learning with tied weights: referring to the equation of getting unbiased samples of the hidden situations from their rearward allocation presented a sampled data vector:

$$\Delta w_{ij} \propto \langle h_j{}^{(k)}\left(h_i{}^{(k-1)} - p(h_i{}^{(k-1)} = 1|h^{(k)}, W^{(k)})\right)\rangle \quad (18)$$

where the angle brackets symbolize an expectancy on the training data. The tied weights imply that the most common way of inducing h^2 from h^1 is equivalent to the method involved with creating v from h^1. Subsequently, h^2 can be considered a noisy gauge of the probabilities for the apparent units anticipated by h^1. Correspondingly h^3 can be considered a boisterous gauge of the probabilities for the units in the main secret layer anticipated by h^2. We can utilize these two realities and the above condition to get a fair-minded gauge of the amount of the subordinates for the initial two layers of loads. This gives the accompanying learning rule which is known as "contrastive divergence":

$$\Delta w_{ij} \propto \langle h_j{}^{(1)}(v_i - h_i{}^{(2)}) + h_i{}^{(2)}(h_j{}^{(1)} - h_j{}^{(3)})\rangle \propto \langle v_i h_j{}^{(1)}\rangle - \langle h_i{}^{(2)} h_j{}^{(3)}\rangle \quad (19)$$

To make reasonable component finders, be that as it may, even a somewhat inadequately tuned generative model is adequate, and the learning rule in the above condition is adequate in any event, when the weights get very enormous. The surmised greatest probability subordinates delivered by this learning rule become profoundly one-sided, yet they are modest to register and they additionally have extremely low difference [53] which is significant for permitting a high gaining rate when the subsidiaries are assessed from little small-scale clumps of information.

2. Learning with various loads in each layer: The thought is to make the generative model all the more impressive by permitting various loads in various layers. The process is to freeze and untie the most minimal duplicate of the right now tied weight frameworks and it can be repeated as many times as needed. *K* layers of different features have been learned with different weight matrices. Then all layers above the K^{th} layer are eliminated and a last "softmax" layer is added of name units addressing HMM (Hidden Markov Model) states. For displaying genuine qualities in the apparent layer, the twofold unit is supplanted by direct units with Gaussian commotion that has a change of 1. The probability $p(l|h^{(K)})$ and mean μ_i are given by [51]:

$$p(l|h^{(K)}) = \frac{exp(p_l + \sum_i h_i{}^{(K)} w_{il})}{\sum_m exp(p_m \sum_i h_i{}^{(K)} w_{im})} \quad (20)$$

$$\mu_i = b_i{}^{(0)} + \sum_j w_{ij}{}^{(1)} h_j{}^{(1)} \quad (21)$$

This kind of generative pre-preparing followed by discriminative adjusting has been utilized effectively for written by hand character acknowledgment, dimensionality decrease 3-D object detection, extricating guides from jumbled flying pictures, data recovery and machine literal interpretation. As we will see, it is additionally awesome for telephone acknowledgment [51].

- Generative Adversarial Networks (GANs): GANs [3] are a DL tactic lately evolved using Goodfellow in 2014 with unsupervised learning to intern two patterns in parallel as described in *Fig. 9*. The Discriminator (D) and Generator (G) perform a min-max competition with a $V(G, D)$ by:

$$min_G max_D V(D, G) = E_{x \sim P_{data}(x)}[log(D(x))] + E_{z \sim P_{data}(z)}[log(1 - D((G(z)))] \quad (22)$$

Received 20/April/2022; Accepted 15/May/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

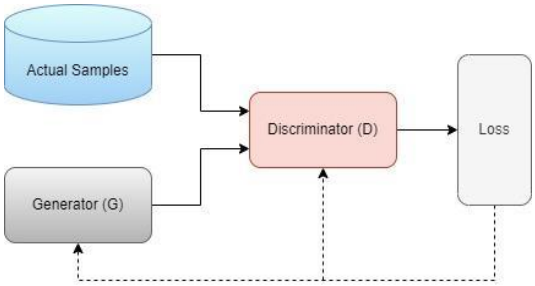

FIG. 9. BLOCK DIAGRAM OF GAN.

By and by, this condition may not give adequate gradient to learning G (what began from arbitrary Gaussian commotion) at the early stages. In those phases can dismiss tests since they are plainly unique contrasted with training samples. In this situation, $log(1 - D((G(z)))$ will be saturated. Rather than preparing G to limit $log(1 - D((G(z))), G$ can be trained to maximize $log(D(x))$ objective capacity which gives much better slopes in beginning phases during learning. Nonetheless, there were a few restrictions of the assembly cycle during preparing with the main rendition. In the first place express a GAN has a few constraints with respect to the accompanying issues: the absence of a heuristic expense work (as pixel-wise estimated implies square mistakes (MSE)) and unsteady to prepare (once in a while that can be reason for creating outlandish results).

- Convolutional Neural Networks (CNNs): These networks are more like the human visual handling framework, fit to protest acknowledgment with 2-D and three-dimensional pictures and reliably top picture grouping rivalries, also powerful at learning and extracting abstractions of 2D elements. The maximum pooling layer of CNNs is powerful in engrossing shape varieties. Additionally, made out of spare associations with tied loads, CNNs have altogether less boundaries than a completely connected network of comparative size. The aim is to learn higher-request highlights in the data through convolutions. They can recognize faces, individuals, street signs, platypuses, and various pieces of visual data. CNNs cross-over with text examination through optical person acknowledgment, yet they are likewise valuable while breaking down word 6, for model, as discrete literary units. They are likewise great at examining sound [48]. As described in *Fig. 10*, CNNs is comprised of two primary parts: feature extractors, where each layer of the network gets the output from its nearby past layer as its feedback, and passes its result as the contribution to the following layer, and a classifier in which one or more layers are completely associated with take the higher-request elements and produce class probabilities or scores. The CNN engineering comprised of a mix of three kinds of layers: max-pooling, convolution and request. Two sorts of layers are in the low and center level of the association: convolutional layers and max-pooling layers. Convolutions have the even numbered layers and max-pooling undertakings have the odd numbered layers. The result hubs of the convolution and max-pooling layers are collected into a 2D plane named "include planning" [3]. The output of the convolutional layers is given by:

$$x_j{}^l = f(\sum_{i \in M_j} x_i{}^{l-1} * k_{ij}{}^l + b_j{}^l) \quad (23)$$

Where $x_j{}^l$ is the output of the stream layer, $x_i{}^{l-1}$ is the output from the previous layer, $k_{ij}{}^l$ is the learnable kernel (the convolution of include maps from past layers) of the present layer, $b_j{}^l$ is the bias of the current layer and M_j is the selection of input maps.

Received 20/April/2022; Accepted 15/May/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

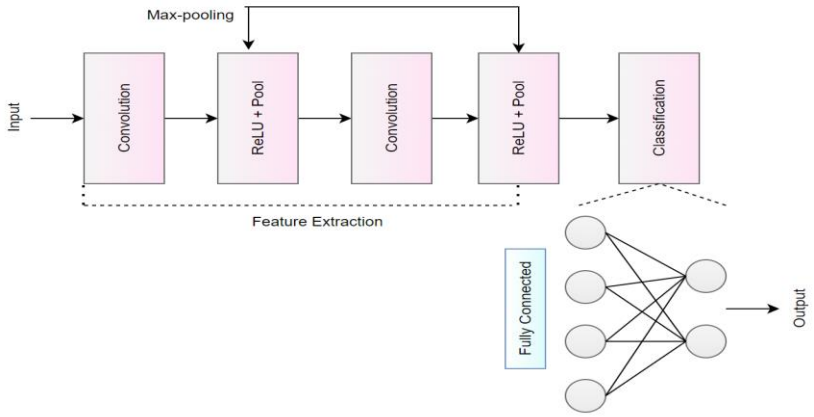

FIG. 10. THE ARCHITECTURE OF CNN.

- Recurrent Neural Networks (RNN): They fall in the group of feed-forward NNs. It centers around utilizing successive information - it processes data utilizing an input circle, permitting the networks to retain knowledge on what has previously been characterized; be that as it may, this data is just put away for a brief time frame before it is lost [54]. They are not quite the same as other feed-forward networks in their ability to send information over time-steps. These networks take every vector from a succession of information vectors and model them each in turn. This permits the network to hold state while demonstrating each information vector across the window of info vectors. Repetitive NNs appear differently in relation to other profound networks in what kind of information they can demonstrate (non-fixed input): non-Fixed computation steps, non-Fixed output size and it can work over groupings of vectors, like edges of video. The essential contusion of these networks is the way it is possible to work with info and result within an individual manner. RNNs are a superset of feed-forward NNs yet they add the idea of repetitive associations. These associations (or repetitive edges) range nearby time-steps (e.g., a past time-step), giving the model the idea of time. The conventional connections don't contain cycles in RNNs. Nonetheless, repetitive associations can frame cycles including associations back to the first neurons themselves at future time-steps. These organizations are recognized to have obstacles within the "evaporating inclination issue". This snag happens when the slopes become excessively enormous or excessively little and make it hard to show long-range conditions (ten time-steps or more) in the construction of the information dataset. The best method for getting around this issue is to utilize the LSTM variation of RNNs which DL4J upholds [48]. The architecture of this type of network is made up of feed forward multi-layers as shown in *Fig. 11* below.

- Recursive Neural Networks: similar to RNNs in dealing with input of different length. The main variation is that RNNs can display the various leveled structures in the preparation of dataset [48]. The engineering is made out of a common-weight matrix and a binary tree structure that permits the recursive network to get the hang of shifting successions of words or portions of an image. It is valuable as a sentence and scene parser. These Networks utilize a variety of backpropagation named Backpropagation Through Structure (BPTS). Varieties of these networks include:
    1. recursive autoencoders (learn how to reconstruct the input like text).
    2. semi-supervised recursive autoencoder (learns the likelihood of certain labels in each context).

Received 20/April/2022; Accepted 15/May/2022

ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

3. Recursive, supervised, NTN-Neural Tensor Network (processes a managed objective at every hub of the tree).

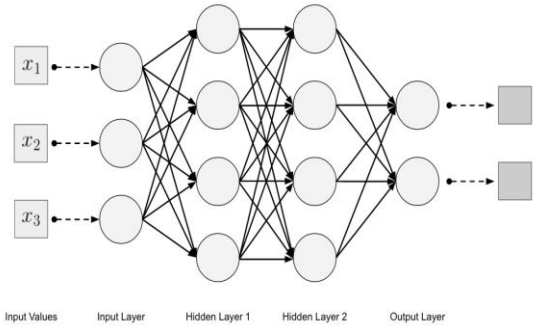

FIG. 11. THE ARCHITECTURE OF LSTM.

The tensor, which is part of three or more dimensions, ascertains the gradient somewhat better, figuring in more information at every hub by exploiting one more component of data utilizing a tensor. Applications for recursive NNs include Image scene decomposition, NLP, and Audio-to-text transcription.

III. THE APPLICATIONS OF DEEP LEARNING

Recently, DL [55] [56]has become an important part in application design. This section will summarize and discuss several applications that have been applied with Deep Learning algorithms.

- **Image Recognition**

Deep Learning concepts Convolutional Neural Networks use for image object recognition [57] because they are an exceptionally viable class of NNs that is profoundly viable at the assignment of picture ordering, object recognition and other PC vision issues [58][59]. Grouping of food pictures, for instance, is extremely difficult since the dataset of food pictures is profoundly non-straight. This system is necessary for dietary assessments. Image recognition innovation is likewise the example acknowledgment innovation of pictures. Its functioning mode is to display picture data, construct models, remove highlights, and afterward investigate and handle the picture, recognize and order the picture as indicated by the element data, lastly accomplish the ideal impact. Different examples on DL in image recognition includes [60][61]:

1. Face recognition: It is an innovation for ID by contrasting facial features. In late years, as the profound learning blast has progressed, numerous scientists have joined face acknowledgment with profound learning. Co-managed by Softmax misfortune and focus misfortune to prepare CNN to further develop acknowledgment precision; CNN-based face acknowledgment framework is totally assessed, and the effect of various designs of CNN on face acknowledgment execution is quantitatively assessed. The outcomes show that CNN Feature combination of various layers in the center can work on the exhibition of face acknowledgment.
2. Human action recognition: This activity is a very important point in PC vision examination, and it has additionally been applied to many fields, like augmented simulation, human-PC canny connection what's more, different fields [62]. Somebody proposed an acknowledgment technique that consolidates speed increase sensors and profound convolutional NNs, which can actually order the five sorts of activities of the human body: strolling, sitting, lying, running, and standing. This technique utilizes the sliding window collapsing technique to change the sensor information into a three-channel data design like RGB pictures. The model naturally extricates data elements to group

Received 20/April/2022; Accepted 15/May/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

each activity; the calculation accomplishes an acknowledgment pace of 91.26% on the Actitracker database. Others have proposed a technique for human activity acknowledgment in light of hereditary calculation and CNN, utilizing hereditary calculation to instate CNN loads to limit classification errors; utilizing genetic algorithm to assess CNN classifiers, utilizing the worldwide hereditary calculation The hunt capacity and the nearby inquiry capacity of the inclination plunge calculation track down the ideal arrangement and work on the presentation of the organizer. In the matter of big data, gathered convolutional NNs are more adaptable than convolutional NNs, with quick preparation speeds and significant acknowledgment rates. However, on account of comparable activities, the element data extricated by the profundity model is generally something similar, which is inclined to misinterpretation of activities.

3. Fall detection: The rate of falls is high in the existences of the old. The rate of falls is high in the existence of the old, so it is of extraordinary importance to consequently distinguish the unplanned fall of the old, which can obviously lessen the gamble of death. The article in [63] gives an extensive survey on cutting edge fall recognition advances thinking about the most remarkable DL philosophies. the latest and viable DL techniques are inspected for fall identification and ordered into three classes: CNN-based frameworks, LSTM-based frameworks, and AE-based frameworks, three layered (3D) CNN, CNN with 10-crease cross-approval, LSTM with CNN based frameworks played out the best regarding exactness, responsiveness, explicitness, and so forth. The surveyed frameworks were thought about in view of their functioning standards, utilized DL techniques, utilized datasets, execution measurements, and so forth. This survey is pointed toward introducing a synopsis and correlation of existing state-of-the-art DL based fall location frameworks to work with future advancement in this field.

- **Automatic Speech Recognition (ASR)**

ASR refers to a machine's or computer's ability to recognize the content of spoken words and phrases and convert them into a machine-understandable format. Speech recognition is useful in a variety of situations. Such uses can be found in dictating computers rather than typing, spaceships with occupied extremities, and many other places. The ASR technology, converting oral speech into text, allows users to operate digital devices by speaking rather than using traditional tools like keystrokes [64][65]. ASR been a concentrated examination region for a really long time. Many center advances, for example, Gaussian Mixture Models (GMMs), stowed away Markov models (HMMs), Mel-Frequency Cepstral Coefficients (MFCCs) and their subsidiaries, discriminative preparation, and different transformation strategies have been created with time, for the most part before the new hundred years. These strategies significantly progressed the best in class in ASR and in its connected fields. Contrasted with these previous accomplishments, the progression in the examination and use of ASR in the ten years before 2010 was generally sluggish and less energizing, albeit significant procedures, for example, GMM-HMM grouping discriminative preparation were made to function admirably in commonsense frameworks during this period [66].

- **Bioinformatics**

Various regions of attraction bother the Omics (investigation of the genome-genomics-and proteins-transcriptomics, proteomics, and metabolomics), bioimaging (investigation of natural tissues), clinical imaging (investigation of the human organs by making visual portrayals), BBMI (investigation of the mind and body machine interface) and Public and medical Health Management (PmHM) [67]. This field is segmented into several sub-fields:

Received 20/April/2022; Accepted 15/May/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

1. Omics: genomics and proteomics are the investigation of the DNA/RNA sequencing and protein associations and design expectation.
2. Bioimaging: the investigation of natural cells and tissue by dissecting histopathology or immunohistochemically pictures.
3. Clinical imaging: the investigation of the human organs by examining attractive reverberation imaging (X-ray), x-beam, and so on.
4. Cerebrum and body machine interface: the investigation of the mind and body interpreting machine interface by breaking down various bio signals like electroencephalogram (EEG), electroencephalogram (EEG), and so forth.
5. Public and clinical wellbeing the executives (PmHM): the investigation of enormous clinical information to create and improve general medical services choices for a mankind prosperity

- **COVID-19 Applications**

One of the most pressing issues of the twenty-first century is COVID-19 illness. Approximately 43 million people have tested positive as of this publication, with 1.2 million people dying as a result [68]. In 2021, Connor Shorten et al, examines how DL is utilized to battle the COVID-19 epidemic besides to giving references for aftertime COVID-19 exploration. DL applications in NLP, Computer Vision and Life Sciences are enveloped and discuss how large data availability affects all these applications, although how to build learning problems. They start by estimating the present level of DL and terminate with a argumentation of DL's main restrictions in COVID-19 implementations. Interpretability, Speculation Measurements, Gaining from Restricted Marked Information, and Information Protection are generally instances of these impediments. Digging Coronavirus research for Data Recovery and Question Responding to, as well as Falsehood Location and Public Feeling Examination, are examples of NLP applications. Medical Image Analysis and Ambient Intelligence are examples of computer vision applications. The aim of their paper is to help accelerate the use of DL for COVID-19 research [69].

Deep learning is used to evaluate the impacted region of the body like cancer, bone issues, accidents, besides to lung ailments using X-ray diagnostics [11,13]. Mishra[70] et al, employed CNN-based models ResNet-50, and Inception-ResNet-v2 to predict COVID-19 patients via chest X-ray images and ResNet-50 had the optimal detection percentage (98%), according to them. The feature extraction procedure with chest X-ray pictures also uses DL algorithm and a support vector machine (SVM) to identify the image as healthy or diseased. Different DL models were employed, including Inception-v3, AlexNet, VGG16, Inception-ResNet-v2, VGG19, ResNet-18, ResNet-50, GoogLeNet, ResNet-101, DenseNet201, and XceptionNet, with ResNet50 and SVM achieving a 95.38% percision. In 1972, Godfrey Hounsfield and Allan Cormack invented the CT scan. CT scan diagnosis using DL is a technique that employs X-ray technology to diagnose sensitive interior organs [11, 13]. In 2020, Harsh P. et al, presented the nCOVnet deep learning neural network-based technique as an alternative rapid screening method for identifying COVID-19 by evaluating patients' X-rays and looking for visual indications identified in COVID-19 patients' chest radiography images [71]. *Fig. 12* describes detection and analysis of COVID-19 in medical images using deep learning techniques.

Received 20/April/2022; Accepted 15/May/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

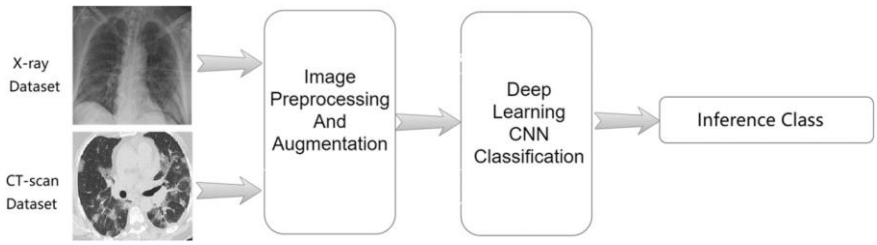

FIG. 12. DETECTING COVID-19 WITH DL TECHNIQUE.

- **Self-Driving Cars**

Deep learning is one possible solution for object identification and scene perception issues, allowing algorithm-driven and data-driven automobiles to be developed [72][73]. Self-driving vehicles [74] are important research topic in science and innovation, which has an extraordinary impact on friendly and monetary turn of events. Profound learning is one of the ongoing key regions in the scope of computerized reasoning examination. It was broadly applied in picture handling, normal language getting it, etc. Lately, increasingly more profound learning-based arrangements is introduced in the field of self-driving vehicles and accomplished extraordinary outcomes. quick headway of the Internet economy and Artificial Intelligence (AI) has progressed the headway of self-driving vehicles. The market interest and monetary worth of self-driving engines are progressively conspicuous. As of now, a steadily expanding number of endeavors besides to logical exploration foundations placed assets into this field. Google, Tesla, Apple, Nissan, Audi, General Motors, BMW, Toyota, Mercedes, Nvidia, and Volkswagen have partaken in the innovative work of self-driving engines [75].

Google is an Internet organization, that is one of the forerunners in self-driving vehicles, in light of its strong establishment in computerized reasoning [76]. In June 2015, two Google self-driving vehicles were tried on the street. Up until this point, Google vehicles have gathered more than 3.2 million km of tests, turning into the nearest to the genuine use. Another organization that has gained extraordinary headway in the field of self-driving vehicles is Tesla (see *Fig. 13*). Tesla was the main organization to dedicate self-driving innovation to creation. Followed by the Tesla models series, its "auto-pilot" innovation has made major forward leaps lately. Albeit the tesla's autopilot innovation is just viewed as Level2 stage by the public interstate traffic security organization (NHTSA), as one of the best organizations in autopilot framework application by a wide margin, Tesla offers that the vehicle has fundamentally understood programmed driving under specific circumstances.

FIG. 13.TESLA'S SELF-DRIVING CAR.

Received 20/April/2022; Accepted 15/May/2022

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

- **Color Based Object Sorting Systems**

Because industries have become fully automated as a result of various technological advancements, an automated sorting system is essentially required to replace the traditional manual sorting system, knowing that this process can be made completely autonomous by properly channeling the use of technology. Thus, Object sorting is a frequent industrial use, but it is also a laborious procedure since handling so many items is a menial effort that does not guarantee consistency, resulting in quality concerns [77][78]. The sorting scenes in current everyday existence, for example, family benefits, trash arranging and planned operations circulation, are many times unstructured scenes. Notwithstanding these conditions, the robot framework not just has to recognize and find the objective item yet additionally needs to grasp the entire climate. Many existing techniques for the most part just distinguish and find objects in the scene by target location or layout coordinating, which can't play a steady arranging impact notwithstanding an assortment of obscure article arranging situations with stacking. Due to the absence of comprehension of the spatial relationship of articles in the scene, the objective item is harmed during the time spent getting a handle on. Subsequently, it is of incredible importance to concentrate on the best way to involve robots for protected, steady and exact arranging of complex multi-object scenes [79].

- **Agriculture Domain**

Deep Learning has recently been used in the agriculture field [80], where it is being used to solve a variety of agricultural and food production problems as a tool that produces high-accuracy findings and, in many cases, outperforms standard image processing approaches [81]. Authors in [82] play out a review of 40 exploration endeavors that utilize DL procedures, applied to different horticultural and food creation challenges. Problem description of these researches' ranges from classifying leaves of different plant species [82], defining kinds of plant diseases out of healthy leaves [83], identifying different land-use classes containing a variety of spatial patterns, to other agricultural domain problems. Research [84] classified leaves of different plant species for Flavia dataset that contains 1,907 leaf images of 32 species with at least 50 images per species and at most 77 images. The uses DL model was Author-defined CNN and RF classifier. The evaluations yield state-of-the art performance with an average accuracy of 97.3%±0:6% compared to traditional features which obtain an average accuracy of 91.2%±1:6%. Authors in[82] identified 13 distinct sorts of plant illnesses out of solid leaves for creators made information base with 4,483 images and used CaffeNet CNN DL model. Their experimental results on the developed model achieved precision between 91% and 98%, for separate class tests, on average 96.3%. Researchers in [83] identified 21 land-use classes containing various spatial examples for UC Merced land-use dataset Aerial ortho-symbolism with a 0.3048-m pixel goal. Dataset incorporated from a choice of 100 pictures/class with Author-defined CNN and Multiview model averaging. The competitive performance is achieved for the UC Merced data set, where the 93.48% accuracy of Multiview deep learning outperforms the 85.37% accuracy of SIFT-based methods and the 90.26% accuracy of unsupervised feature learning.

IV. HARDWARE/SOFTWARE TOOLS USED WITH DEEP LEARNING

Many deep learning textboxes are widely available on the internet. The codes for several deep learning models, such as DBNs, LeNet-5, AlexNet [85] and VGGNet [86] are frequently given in each textbox. Under certain permissions, researchers may utilize the codes directly or construct new models based on the codes. Theano, fCaffe, gTensorFlow, and MXNet [87] are briefly introduced in the following sections. It's a Python library called Theano. It's strongly integrated with NumPy and allows users to quickly design, optimize, and evaluate multi-dimensional array mathematical expressions. Furthermore, it could conduct

Received 20/April/2022; Accepted 15/May/2022
 ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

data-intensive operations 140 times faster on GPUs than on CPUs. TensorFlow [88] is an open-source software framework that uses data flow graphs to do numerical computations. The network nodes represent mathematical operations, and the graph edges represent multidimensional data arrays (tensors) that are transferred between them. MXNet is a collaborative effort between a number of colleges and businesses. Multiple programming languages, including C, Python, Scala [89], Julia [90], Matlab, and Javascript, are supported, as well as imperative and symbolic programming. In general, the speed of MXNet codes is comparable to Caffe codes [91], and substantially quicker than Theano and TensorFlow codes. MXNet is currently supported by Azure and AWS, two major public cloud providers. k At Amazon Web Services, MXNet has been chosen as the deep learning framework of choice. Caffe is a deep learning toolbox written entirely in C/CUDA. It does, however, include command-line, Python, and MATLAB interfaces. Caffe codes are fast and can swap seamlessly between CPU and GPU [8].

V. CONCLUSIONS

In this review paper, a detailed explanation about DL and its architecture besides to different applications was introduced. Although, a historical background was illustrated. In the broad view, DL represents a solution to different tasks and issues in today's programming and advanced technology. The most important uses of DL lie in the field of the image processing classification and prediction techniques. We can also conclude that the most appropriate programming language for implementing such algorithms is Python in the open-source TensorFlow framework.

REFERENCES

[1] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput*, vol. 18, no. 7, 2006, doi: 10.1162/neco.2006.18.7.1527.

[2] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science (1979)*, vol. 313, no. 5786, 2006, doi: 10.1126/science.1127647.

[3] M. Z. Alom *et al.*, "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches," Mar. 2018.

[4] Y. Bengio, "Learning deep architectures for AI," *Foundations and Trends in Machine Learning*, vol. 2, no. 1, 2009, doi: 10.1561/2200000006.

[5] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, 2015, doi: 10.1038/nature14236.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition." [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[7] R. A. FISHER, "THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS," *Ann Eugen*, vol. 7, no. 2, 1936, doi: 10.1111/j.1469-1809.1936.tb02137.x.

[8] G. Zhong, L. N. Wang, X. Ling, and J. Dong, "An overview on data representation learning: From traditional feature learning to recent deep learning," *Journal of Finance and Data Science*, vol. 2, no. 4. 2016. doi: 10.1016/j.jfds.2017.05.001.

[9] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol Rev*, vol. 65, no. 6, 1958, doi: 10.1037/h0042519.

[10] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities.," *Proceedings of the National Academy of Sciences*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982, doi: 10.1073/pnas.79.8.2554.

[11] D. H. Ackley, G. E. Hinton, and T. J. Sejnowski, "A learning algorithm for boltzmann machines," *Cogn Sci*, vol. 9, no. 1, 1985, doi: 10.1016/S0364-0213(85)80012-4.

[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, 1986, doi: 10.1038/323533a0.

[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, 1998, doi: 10.1109/5.726791.

[14] F. A. Gers and J. Schmidhuber, "Recurrent nets that time and count," in *Proceedings of the International Joint Conference on Neural Networks*, 2000, vol. 3. doi: 10.1109/ijcnn.2000.861302.

[15] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of Machine Learning Research*, vol. 3, no. 1, 2003, doi: 10.1162/153244303768966139.

Received 20/April/2022; Accepted 15/May/2022
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.
[17] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, vol. 2017-January. doi: 10.1109/CVPR.2017.243.
[18] M. Ren, W. Zeng, B. Yang, and R. Urtasun, "Learning to reweight examples for robust deep learning," in *35th International Conference on Machine Learning, ICML 2018*, 2018, vol. 10.
[19] M. Amodio *et al.*, "Exploring single-cell data with deep multitasking neural networks," *Nat Methods*, vol. 16, no. 11, 2019, doi: 10.1038/s41592-019-0576-7.
[20] B. Bigdeli, H. Amini Amirkolaee, and P. Pahlavani, "Deep feature learning versus shallow feature learning systems for joint use of airborne thermal hyperspectral and visible remote sensing data," *Int J Remote Sens*, vol. 40, no. 18, 2019, doi: 10.1080/01431161.2019.1597310.
[21] J. Johnson *et al.*, "Accelerating 3D deep learning with PyTorch3D," 2020. doi: 10.1145/3415263.3419160.
[22] K. Yu, L. Tan, L. Lin, X. Cheng, Z. Yi, and T. Sato, "Deep-Learning-Empowered Breast Cancer Auxiliary Diagnosis for 5GB Remote E-Health," *IEEE Wirel Commun*, vol. 28, no. 3, 2021, doi: 10.1109/MWC.001.2000374.
[23] "What is the difference between deep learning and shallow learning?" https://ai.stackexchange.com/questions/21219/what-is-the-difference-between-deep-learning-and-shallow-learning (accessed Oct. 15, 2021).
[24] G. Zhong, X. Ling, and L. N. Wang, "From shallow feature learning to deep learning: Benefits from the width and depth of deep architectures," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 9, no. 1. 2019. doi: 10.1002/widm.1255.
[25] O. Sharma, "Deep Challenges Associated with Deep Learning," 2019. doi: 10.1109/COMITCon.2019.8862453.
[26] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *J Big Data*, vol. 2, no. 1, 2015, doi: 10.1186/s40537-014-0007-7.
[27] X. W. Chen and X. Lin, "Big data deep learning: Challenges and perspectives," *IEEE Access*, vol. 2. 2014. doi: 10.1109/ACCESS.2014.2325029.
[28] M. Svensen and C. M. Bishop, "PRML: Errata-2ed," *J Electron Imaging*, vol. 16, 2006.
[29] S. Russell and J. Bohannon, "Artificial intelligence. Fears of an AI pioneer.," *Science*, vol. 349, no. 6245, 2015.
[30] A. Sabharwal and B. Selman, "Book review; S. Russell, P. Norvig, Artificial Intelligence: A Modern Approach, Third Edition.," *Artif Intell*, vol. 175, no. 5–6, 2011.
[31] Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective (Adaptive Computation and Machine Learning series)*, Illustrated edition. The MIT Press; Illustrated edition , 2012.
[32] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction (Adaptive Computation and Machine Learning series)*, vol. 258, no. 6685. 1998.
[33] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019, vol. 2019-June. doi: 10.1109/CVPR.2019.00202.
[34] Tom M. Mitchell, *Machine Learning (McGraw-Hill International Editions Computer Science Series)*, Paperback. McGraw-Hill Education; 1st edition, 1997.
[35] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar, *Foundations of Machine Learning (Adaptive Computation and Machine Learning series)*. The MIT Press, 2012.
[36] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "A Brief Survey of Deep Reinforcement Learning," Aug. 2017, doi: 10.1109/MSP.2017.2743240.
[37] Ian H. Witten, Eibe Frank, and Mark A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques (Morgan Kaufmann Series in Data Management Systems) 4th Edition*, 4th ed. Morgan Kaufmann; 3rd edition, 2011.
[38] VAPNIK and V. N., "The Nature of Statistical Learning," *Theory*. 1995.
[39] I. Goodfellow, Y. Benigo, and Courville Aaron, "Deep Learning (Adaptive Computation and Machine Learning series)," *MIT Press*. 2016.
[40] T. Hastie, R. Tibshirani, and J. Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition (Springer Series in Statistics) (9780387848570): Trevor Hastie, Robert Tibshirani, Jerome Friedman: Books," in *The elements of statistical learning: dta mining, inference, and prediction*, 2011.
[41] James Stewart, *Calculus: Early Transcendentals*, 6th edition. Cengage Learning; 6th edition, 2007.
[42] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *30th International Conference on Machine Learning, ICML 2013*, 2013, no. PART 3.
[43] L. Bottou, "Stochastic gradient descent tricks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7700 LECTURE NO, 2012, doi: 10.1007/978-3-642-35289-8_25.
[44] D. Gupta, "Fundamentals of Deep Learning – Activation Functions and When to Use Them?," *Analytics Vidhya*, 2020.
[45] J. Brownlee, "How to Choose an Activation Function for Deep Learning," *Machine Learning Mastery*, 2021.
[46] H. Tatsat, S. Puri, and B. Lookabaugh, *Machine Learning and Data Science Blueprints for Finance*. 2020.
[47] F.rancois Chollet, *Deep Learning with Python 1st Edition*. Manning; 1st edition, 2017.
[48] A. J. Patterson and A. Gibson, *Patterson, J., Gibson, A. (2017). Deep Learning: A Practitioner's Approach.*, no. 1. 2017.

Received 20/April/2022; Accepted 15/May/2022
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

[49] D. Chicco, P. Sadowski, and P. Baldi, "Deep autoencoder neural networks for gene ontology annotation predictions," 2014. doi: 10.1145/2649387.2649442.
[50] M. Sewak, S. K. Sahay, and H. Rathore, "An overview of deep learning architecture of deep neural networks and autoencoders," *J Comput Theor Nanosci*, vol. 17, no. 1, 2020, doi: 10.1166/jctn.2020.8648.
[51] A. R. Mohamed, G. E. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *IEEE Trans Audio Speech Lang Process*, vol. 20, no. 1, 2012, doi: 10.1109/TASL.2011.2109382.
[52] Y. Freund and D. Haussler, "Unsupervised learning of distributions on binary vectors using two layer networks," *Igarss 2014*, no. 1, 2014.
[53] Christopher K. I. Williams and Felix V. Agakov, "An analysis of contrastive divergence learning in gaussian boltzmann machines," 2002. [Online]. Available: http://deeplearning.cs.cmu.edu/pdfs/Williams_Agakov.2002.pdf
[54] "Teknik och samhälle Datavetenskap och medieteknik Examensarbete A COMPARATIVE STUDY OF DEEP-LEARNING APPROACHES FOR ACTIVITY RECOGNITION USING SENSOR DATA IN SMART OFFICE ENVIRONMENTS."
[55] P. P. Shinde and S. Shah, "A Review of Machine Learning and Deep Learning Applications," 2018. doi: 10.1109/ICCUBEA.2018.8697857.
[56] "Top Applications of Deep Learning Across Industries." https://www.mygreatlearning.com/blog/deep-learning-applications/ (accessed Aug. 07, 2021).
[57] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun ACM*, vol. 60, no. 6, 2017, doi: 10.1145/3065386.
[58] K. Horak and R. Sablatnig, "Deep learning concepts and datasets for image recognition: overview 2019," 2019. doi: 10.1117/12.2539806.
[59] D. Keysers, T. Deselaers, C. Gollan, and H. Ney, "Deformation models for image recognition," *IEEE Trans Pattern Anal Mach Intell*, vol. 29, no. 8, 2007, doi: 10.1109/TPAMI.2007.1153.
[60] N. F. Hordri, S. S. Yuhaniz, and S. M. Shamsuddin, "Deep Learning and Its Applications: A Review," *Conference on Postgraduate Annual Research on Informatics*, no. October, 2016.
[61] M. Feng, "Research on the Application of Deep Learning in Computer Image Recognition," *J Phys Conf Ser*, vol. 1915, no. 4, p. 042034, May 2021, doi: 10.1088/1742-6596/1915/4/042034.
[62] N. Jaouedi, N. Boujnah, and M. S. Bouhlel, "A new hybrid deep learning model for human action recognition," *Journal of King Saud University - Computer and Information Sciences*, vol. 32, no. 4, 2020, doi: 10.1016/j.jksuci.2019.09.004.
[63] M. Islam *et al.*, "Deep Learning Based Systems Developed for Fall Detection: A Review," *IEEE Access*, vol. 8. 2020. doi: 10.1109/ACCESS.2020.3021943.
[64] S. Sharmadha, K. Shivani, K. Shruthi, B. Bharathi, and S. Kavitha, "Automatic Speech Recognition Using Deep Neural Network," in *Advances in Intelligent Systems and Computing*, 2020, vol. 1118. doi: 10.1007/978-981-15-2475-2_33.
[65] A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, and K. Shaalan, "Speech Recognition Using Deep Neural Networks: A Systematic Review," *IEEE Access*, vol. 7, 2019, doi: 10.1109/ACCESS.2019.2896880.
[66] Dong Yu and Li Deng, *Automatic Speech Recognition: A Deep Learning Approach (Signals and Communication Technology)*, Kindle edition. Springer; 2015th edition, 2015.
[67] R. Zemouri, N. Zerhouni, and D. Racoceanu, "Deep learning in the biomedical applications: Recent and future status," *Applied Sciences (Switzerland)*, vol. 9, no. 8, 2019, doi: 10.3390/app9081526.
[68] "COVID-19 CORONAVIRUS PANDEMIC." https://www.worldometers.info/coronavirus (accessed Jan. 07, 2021).
[69] C. Shorten, T. M. Khoshgoftaar, and B. Furht, "Deep Learning applications for COVID-19," *J Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-020-00392-9.
[70] P. Mishra, C. Kochgaven, and S. Shitole, "Medical Imaging and its Role in Detection of COVID-19," 2021. doi: 10.1109/CONIT51480.2021.9498413.
[71] H. Panwar, P. K. Gupta, M. K. Siddiqui, R. Morales-Menendez, and V. Singh, "Application of deep learning for fast detection of COVID-19 in X-Rays using nCOVnet," *Chaos Solitons Fractals*, vol. 138, 2020, doi: 10.1016/j.chaos.2020.109944.
[72] M. Zhou *et al.*, "SMARTS: Scalable Multi-Agent Reinforcement Learning Training School for Autonomous Driving," Oct. 2020.
[73] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," *Array*, vol. 10, 2021, doi: 10.1016/j.array.2021.100057.
[74] S. Lade, P. Shrivastav, S. Waghmare, S. Hon, S. Waghmode, and S. Teli, "Simulation of self driving car using deep learning," 2021. doi: 10.1109/ESCI50559.2021.9396941.
[75] J. Ni, Y. Chen, Y. Chen, J. Zhu, D. Ali, and W. Cao, "A survey on theories and applications for self-driving cars based on deep learning methods," *Applied Sciences (Switzerland)*, vol. 10, no. 8. 2020. doi: 10.3390/APP10082749.
[76] M. Birdsall, "Google and ITE : the road ahead for self-driving cars," *ITE Journal (Institute of Transportation Engineers)*, vol. 84, no. 5, 2014.
[77] Y. J. Heo, S. J. Kim, D. Kim, K. Lee, and W. K. Chung, "Super-high-purity seed sorter using low-latency image-recognition based on deep learning," *IEEE Robot Autom Lett*, vol. 3, no. 4, 2018, doi: 10.1109/LRA.2018.2849513.
[78] D. Díaz-Romero, W. Sterkens, S. van den Eynde, T. Goedemé, W. Dewulf, and J. Peeters, "Deep learning computer vision for the separation of Cast- and Wrought-Aluminum scrap," *Resour Conserv Recycl*, vol. 172, 2021, doi: 10.1016/j.resconrec.2021.105685.

Received 20/April/2022; Accepted 15/May/2022
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.23.1.4

[79] H. Zhang, H. Liang, T. Ni, L. Huang, and J. Yang, "Research on multi-object sorting system based on deep learning," *Sensors*, vol. 21, no. 18, 2021, doi: 10.3390/s21186238.
[80] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, vol. 147. 2018. doi: 10.1016/j.compag.2018.02.016.
[81] D. M. Bongulwar, "Identification of Fruits Using Deep Learning Approach," *IOP Conf Ser Mater Sci Eng*, vol. 1049, no. 1, 2021, doi: 10.1088/1757-899x/1049/1/012004.
[82] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification," *Comput Intell Neurosci*, vol. 2016, 2016, doi: 10.1155/2016/3289801.
[83] F. P. S. Luus, B. P. Salmon, F. van den Bergh, and B. T. J. Maharaj, "Multiview Deep Learning for Land-Use Classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 12, 2015, doi: 10.1109/LGRS.2015.2483680.
[84] D. Hall, C. McCool, F. Dayoub, N. Sünderhauf, and B. Upcroft, "Evaluation of features for leaf classification in challenging conditions," 2015. doi: 10.1109/WACV.2015.111.
[85] N. A. Muhammad, A. A. Nasir, Z. Ibrahim, and N. Sabri, "Evaluation of CNN, alexnet and GoogleNet for fruit recognition," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 2, 2018, doi: 10.11591/ijeecs.v12.i2.pp468-475.
[86] E. Prasetyo, N. Suciati, and C. Fatichah, "Multi-level residual network VGGNet for fish species classification," *Journal of King Saud University - Computer and Information Sciences*, 2021, doi: 10.1016/j.jksuci.2021.05.015.
[87] M. Li *et al.*, "Improving the Performance of Distributed MXNet with RDMA," *Int J Parallel Program*, vol. 47, no. 3, 2019, doi: 10.1007/s10766-018-00623-w.
[88] Tutorials Point, "TensorFlow Tutorial," *Tutorials Point (I) Pvt. Ltd.*, 2019.
[89] M. Guller, "Programming in Scala," in *Big Data Analytics with Spark*, 2015. doi: 10.1007/978-1-4842-0964-6_2.
[90] Kabir, "Julia – Programming Language," *Machine Learning +*, 2020.
[91] V. Turchenko, E. Chalmers, and A. Luczak, "A deep convolutional auto-encoder with pooling - unpooling layers in caffe," *International Journal of Computing*, vol. 18, no. 1, 2019, doi: 10.47839/ijc.18.1.1270.

Received 20/April/2022; Accepted 15/May/2022
 ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*