Review of Neural Network Controllers for Robot Manipulators

Maqbool Ahemad Shaikh
Mechanical Engineering Department
MIT – Art, Design and Technology University
Pune, India
maqbool.ahemad@gmail.com

Prof. M. N. Dhavalikar
Mechanical Engineering Department
MIT – Art, Design and Technology University
Pune, India
mangesh.dhavalikar@mituniversity.edu.in

***Abstract*—The rapid growth of factory automation has led to the utilisation of a significant number of industrial robot manipulators in production lines. This is done in order to improve both the efficiency of production and the quality of the end product. A position control technique known as computed-torque control, which is also known as inverse dynamic control, decouples each joint of the robot and linearizes it using projected robot dynamic models. However, this technology suffers from performance loss due to model uncertainties and must be used with caution. The incorporation of a neural network (NN) controller into the control system is one efficient method of mitigating the detrimental impacts that are brought on by the aforementioned uncertainties. Robot manipulators equipped with Neural Network controllers have potential applications in a variety of domains, including orientation control, force management, parallel-link mechanism control, and digital NN control. Understanding the neural network controller for industrial vision-based manipulators and algorithms is the objective of this review.**

Keywords— Neural Network, Convolutional Networks, Types of Layers, Robot vision and manipulators.

I. Introduction

Robots have become widely utilised in industries as a result of the need for industrial automation and the advancement of robot control theory. Robot systems in manufacturing are utilised to do repetitive activities in controlled environment. In an unstructured environment, framework stability and execution can't be guaranteed. Many robot controllers are now plagued with issues, and vision-based control is a viable solution. It would enable robots to be operated in more disorganized surroundings. The bulk of vision-based robot manipulator control research has practiced free movement control, which has limited implementations. At the point when the end-effector comes into contact with a limitation surface, it's basic to control the movement as well as the power.

" There are numerous laws suggested for the regulation of forces used by robots to manipulate objects. In despite the variety of ways, the vast majority of these methods may be split into two main groups: impedance management and hybrid position/force modulation." [9-11]. The majority of these force control approaches require the robots to be operated in nicely organized surrounding with well-defined surface constraint and also the kinematics and dynamics. "Constraint uncertainty is a significant study topic because it is often difficult to simulate the constraint surface precisely in practice"[34]. Also, an unclear constraint makes it difficult to determine the necessary location at restraint surface in Cartesian space, resulting in difficulty in finding the desired joint position. Furthermore, an object or tool is picked up by the robot with an unknown length, orientation, or gripping point, its kinematics become unpredictable. Thus, in situations where the working environmental factors and article position are difficult to manage, robots stay futile. Fig. 1 shows the conversion of image to grey scale and then to binary values of the pixels with colour. "Cheah et al. [27] projected a sensory feedback control law.

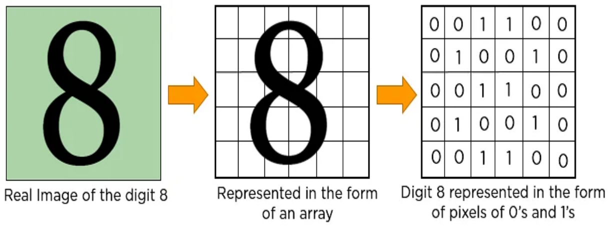

Fig. 1. Conversion of Real Image to Pixel Value [41].

Object recognition forms the reason for more complex and undeniable level visual errands, including division, scene understanding, target following, picture portrayal, occasion discovery, and action identification, as it is the foundation of picture comprehension and CV.

Advancement of AI, particularly profound learning, has been research focus in recent years that, how robot can bypass barriers through self-learning. "Deep learning is an end-to-end learning strategy that uses a deep learning network to map the link between input and output" [1]. "That is, by feeding a large amount of data into the algorithm, the system automatically learns the features of the data. End-to-end learning, as demonstrated by Levine et al. [12], outperforms the traditional approach of fixed vision layers". Yenn LeCun built the first CNN called LeNet in 1988. Convolutional networks recently gained considerable success within domains of object detection. Many Convolutional neural research have concentrated on challenges such as tiny item detection, face detection, crowd detection, traffic sign detection, and automobile detection. CNNs have far less associations and boundaries and are accordingly simpler to train.

Regardless of their engaging benefits and the overall productivity of their design, CNNs have remained restrictively costly to apply to high-resolution inputs for an expansive scope. Fig. 2 gives the general idea of layers in CNN and the sequence of the layer in CNN. Luckily, the present GPUs, when joined with a very much advanced execution of 2D convolution, are adequately strong to consider the training of large CNNs, while ongoing datasets like ImageNet give an adequate number of named guides to prepare models without critical overfitting. The Convolutional Neural Network incorporates the action of convolution, and serves as the basis for the network's concept. The process of convolution is a mathematical operation that, when performed on two capabilities, results in the creation of a third ability that conveys how the status of the first two capabilities is modified by the final one.. In CNN each picture is addressed as varieties of pixel values.

Types of Layers in Convolution Neural Network: -

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

1) Convolution Layer 2) Activation (ReLU) Layer

3) Pooling Layer 4) Fully Connected Layer

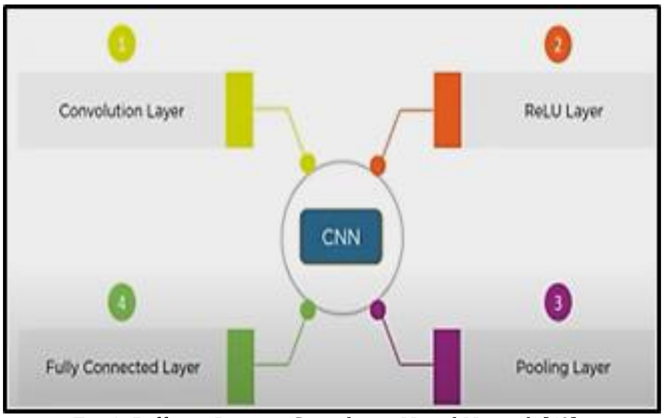

Fig. 2. Different Layer in Convolution Neural Network [40].

II. WORKING OF CNN

Convolutional neural networks are the main parts of NN. The name "convolutional neural network" alludes to the network's utilization of the convolutional numerical strategy. CNNs are a neural organization that involves convolution instead of generalizing matrix multiplication increase in no less than one layer. CNNs utilize little pre-handling contrasted with other picture arrangement algorithms. To identify objects, recognize faces, etc. CNNs use picture acknowledgment and arrangement. They are comprised of neurons with loads and inclinations that can be learned. Every neuron gets countless data sources and afterward registers a weighted aggregate, which it then, at that point, goes through an initiation work prior to reacting with a result. CNNs are chiefly used to bunch pictures, pack them by resemblances, and subsequently perform object acknowledgment. Various estimations using CNNs can recognize faces, street signs, animals, etc.

They use multi-directed pictures and incited by volume. CNNs, then again, cannot recognize level pictures with just width and tallness that people can see. Because of the red-blue-green (RGB) encoding in advanced shading pictures, CNNs mix those three tones to shape the shading range that people see. There are different hidden and fully connected layers in the NN such as I. Convolutional layer, II. Activation layer, III. Pooling layer which also have flattening process, IV. Fully connected layer which gives the final output. Fig. 3 shows the connected layers in CNN and the recognition of the bird.

A. *Convolutional Layer*

It is the principal layer of a CNN, and it is the fundamental structure block that handles most of the computational work. Channels or portions are utilized to convolve information or pictures. Channels are little units that is used in a sliding window to apply across the information. Fig. 4 gives the example of mapping.

The picture's profundity is equivalent to the contribution; for instance, in the event that the RGB worth of depth is 4, filter with a profundity of 4 is imposed on picture. Mentioned method involves getting the component savvy result of the picture's channels and afterward adding those precise qualities for each sliding development. A 2d grid would be the result of a convolution with a 3d shading channel. A convolution is necessary application of channel to information that results in activation. Repetitive application of a comparison channel to a data yields a feature map, which displays the areas and strength of a detected component in a data, similar to a picture. Convolutional layers apply a convolution movement to the data, passing the result to the accompanying layer. A convolution changes over all of the pixels in its responsive field into a singular worth

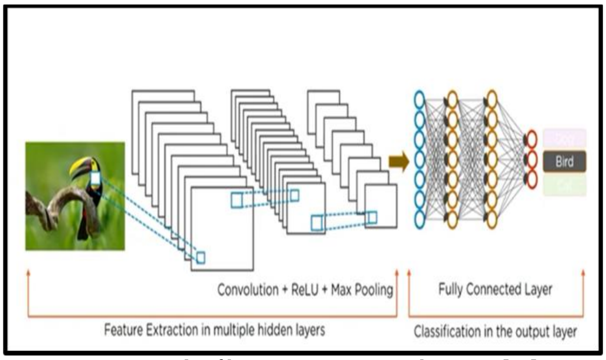

Fig. 3. Example of how CNN recognizes the image [41].

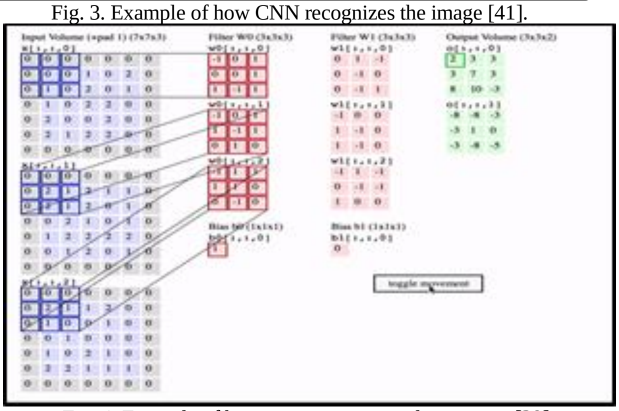

Fig. 4. Example of how inputs are mapped to outputs [39].

.Activation Layer

The Rectified Linear Activation Function (ReLU) for short is a piecewise linear capacity that yield the info straightforwardly assuming it positive, any other way yield zero. It has become the default initiation work for various types of NN since a model that uses it is less difficult to prepare and consistently achieves superior execution. The ACTIVATION LAYER uses different activation functions that helps in activating the nodes present in the layer. The ReLU function is the most commonly used function in the CNN nowadays. In this, the rectifier function is utilized to expand the CNN's non-linearity. Various things that are not direct to one another are utilized to make pictures. There are also different activation functions such as follows:

$$S(x) = \frac{e^x}{e^x + 1} = \frac{1}{1 + e^{-x}} \qquad (1)$$

Sigmoid - A numerical function with a trademark "S"-moulded curve, otherwise called a sigmoid curve, is known as a sigmoid capacity. The logistic function portrayed by the equation, is a continuous illustration of a sigmoid function.

Tanh - With a distinction in yield scope of - 1 to 1, the Tanh function is surprisingly like the sigmoid function, and even has a similar S-shape. The bigger the information the nearer the result to 1.0, and the more modest the info the nearer the result to - 1.0.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^x} \qquad (2)$$

ReLU - ReLU has derivative function and empowers for back propagation while remaining computationally effective, in spite of the way that it is by all accounts a direct capacity. Neurons will be deactivated assuming that result of direct change is under 0.

$$f(x) = \max(0, x) \qquad (3)$$

Leaky ReLU - Leaky Rectified Linear Unit, or Leaky ReLU, is a sort of enactment function in light of a ReLU, yet it has a little slant for negative qualities rather than a level incline.

$$f(x) = \max(0.1x, x) \qquad (4)$$

Maxout – This activation function is itself trained by our model. A single Maxout unit can be deciphered as making a piecewise direct guess to an arbitrary convex function.

$$h(x) = max(w_1^T x + b_1, w_2^T x + b_2) \qquad (5)$$

ELU - Exponential Linear Unit is also variant of ReLU which alters the incline of the negative piece of the function. ELU utilizes a log bend to characterize the negative qualities dissimilar to the Leaky ReLU and Parametric ReLU capacities with a straight line.

$$f'(x) = \begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases} \qquad (6)$$

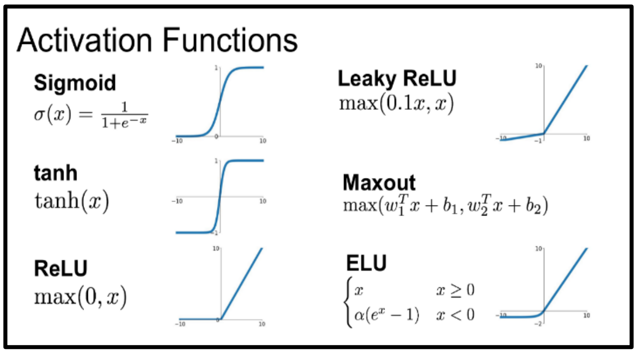

Fig. 5. Different types of Activation Function [42].

Fig. 5 is giving the mathematical function of each activation layer, that can be used in CNN.

B. Pooling Layer

The POOLING LAYER is the third layer, which involves feature down sampling. In order to reduce the components of the element maps, pooling layers are used. As a result, it reduces the number of boundaries to memorise and the amount of math required. The pooling layer sums up the highlights present in an area of the component map created by a convolution layer. In this way, further activities are performed on summed up highlights rather than exactly situated highlights created by the convolution layer. This makes the model more powerful to varieties in the place of the highlights in the information picture. Every layer in 3D-volume gets it. Within this layer, there are usually hyper parameters:

- The dimension of spatial extent: is worth of n which take N-cross & element portrayal and guide to solitary value.
- Stride: is number of highlights that sliding window skips along the width and height.

Typical pooling layer utilizes 2×2 max channel with step of '2', this is non-covering filter. Most filters would return the greatest value in components inside the area. In the wake of completing the past two stages, we should have a pooled include map at this point. As the name of this stage deduces, we are from a genuine perspective going to fix our pooled feature map into a fragment like in the image under. Fig. 6 is the example of 2 by 2 max pooling for the m

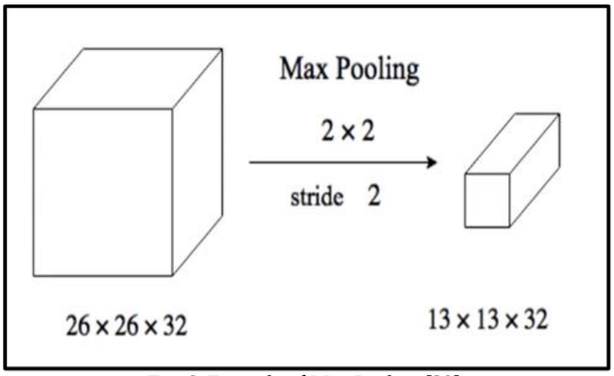

Fig. 6. Example of Max Pooling [39].

C. Fully Connected Layer

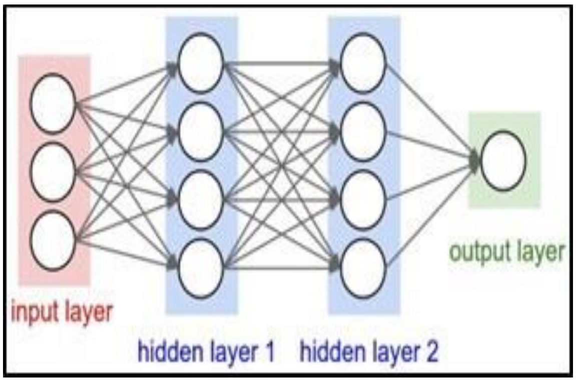

Fig. 7 Fully Connected layer [39].

This is the layer which incorporates Flattening, in last advance. The whole pooling featured map grid is changed into a single segment, which is then provided to NN for handling. Which assembles these characteristics to make a model utilizing the completely connected layers. At last, to order the result, we utilize an activation function like SoftMax or sigmoid. This Layers structure is the network's final couple of layers. Contribution to the fully associated layer is the outcome of the previous Pooling or Convolutional Layer, which is fixed and then dealt with into the completely associated layer. FC layers are utilized to recognize explicit worldwide designs of the elements identified by the lower layers in the net. Any movement of entirely linked layers connecting each neuron in single layer to each neuron in other is referred to as a totally related neural network (NN). Fig. 7

show the connection of I/O layers with the help fully connected layers.

III. LITERATURE REVIEW

Neural Networks are the interconnected network of neurons or nodes. A NN is development of algorithms that pursuits to detect fundamental connections in bunch of information through a cycle that impersonates the manner in which human mind works. These NNs can be used to control, to predict, also to recognize image. Emulating the elements of human cycles, these regulators find out with regards to the frameworks they are controlling on-line, and subsequently naturally work on their presentation. Up until this point, NN have transformed the areas of allocation and pattern identification: with this achievement they've turned into a significant apparatus in the rundown of the signal processor and PC researcher.

The field of neural networks has produced a large and diverse body of knowledge. Signal processing and classification, as well as control, are two types of applications for which neural networks can be used. Besides, open and closed loop control are two sorts of control applications. Signal processing and classification applications are similar in spirit to identification applications in that they regularly utilize similar calculations. However, because the NN is located within the control circle in shut circle applications, the engineer should take extra care to guarantee that the NN loads remain constrained during the control run. Because of these difficulties, designers have been not able to make solid and adaptable NN closed-loop control frameworks.

Canglong Liu et al. [1] talked about the way to deal with the versatile robot investigate climate and snag aversion utilizing techniques for start to finish learning-based CNN. The dataset was utilized to prepare a deep NN, which changes RGB pictures to guiding orders straightforwardly. In an established climate, we likewise covered how to increment precision by changing the climate mark. In a continuous test, our method had the option to accomplish phenomenal precision in deterrent evasion for portable robots. In the test led, the robot anticipated the directions of the relating activity by the prepared model and play out the activities of ROS. All through the activity, the robot tried not to hit the KT board and tried not to hit the square table in the focal point of the area.

Ming Xiao et al. [2] Faster R-CNN is used for object detection in this research, and a method that blends Faster R-CNN with the humanoid robot simulation environment is presented for object identification. By tweaking the initial convolutional network, certain performance improvements can be realised, and further experimental study can be carried out using the simulation platform, with relatively decent results. Meanwhile, additional research and testing are required to improve the overall dependability and practicability of the simulation platform so that our findings may be applied. We also need to undertake more research on a humanoid robot's motion trajectory planning to improve its flexibility.

Y. Zhao et al. [3] proposed the vision based NN regulator for robot controllers with dubious imperative surface, kinematics, and dynamics. The proposed regulator doesn't need the limitation surface's particular model and design. It is shown that the soundness can be accomplished with these vulnerabilities. Simulation is led, in which the point and wanted point are situated on the genuine imperative surface. The ideal point can be acquired in picture space from the camera.

Neville Hogan [4-6] Part-1: In this paper a bound together way to deal with control named impedance control is introduced. By assuming that no control technique can cause an actual framework to act like something besides an actual framework, security diagram network thoughts can be utilized to delineate how the regulator can change the controller's conduct. Following that, a couple of fundamental yet significant perceptions can be made: To direct powerful connection between frameworks, the regulator should order and control a connection between port factors as well as telling and controlling a vector like position or power. That connection should be an impedance, a capacity, perhaps nonlinear, dynamic, or even spasmodic, showing the power created because of a movement forced by the climate in the most well-known situation (e.g., a mass, potentially kinematically restricted).

Part-2: The procedures for carrying out an ideal impedance on a controller were considered in this paper. Procedures for end-guide impedance of the controller without plan of action toward input were additionally talked about. Different actuators and "extra" linkage levels of opportunity can be utilized to manage end-point impedance, and it's been conjectured that the primate outer muscle framework's evident redundancies may truly assume a significant part in regulating intelligent conduct. An element of impedance control is that different regulator activities might be superimposed.

Part-3: In this paper it is shown that as a general rule, the impedance suitable to a given assignment might be found from the undertaking objective, and a technique which utilizes improvement hypothesis. The technique for controlling a controller which might communicate powerfully with its current circumstance. Impedance control might give the premise to understanding device involving conduct in primates, re-establishing this capacity to a tragically handicapped person utilizing a fake appendage, a carrying out it on a modern robot.

Ajith Thomas et al. [7] designed the robot: Fume Bot which is constrained by deep neural-network for ecological observing. Information was broadcast remotely to a PC from an automated stage that was equipped with a variety of sensors and cameras. The robot's route was planned using the CNN regulator, which was produced using an administered learning technique using the video feed and directional commands provided to the robot. The prepared neural organization was placed through its speeds on an obstruction course to perceive how well it could perform. The CNN achieved a magnificent speculation of the gig, as it was as yet ready to finish the fundamental responsibilities when given circumstances that were not in the dataset. Consolidating memory capacities into the organization has been featured as the subsequent stage in conquering the framework's current requirements. It has additionally given the stage to test for new neural organization engineering for versatile robots.

Teck-Seng Low et al. [8] portrayed a system for training of NN for handling drives having nonlinearities, in light of a closed loop training technique. The nature of training data gave has a significant impact on how well a multi-layered feed-forward neural network learns a desired function. Inadequate and insufficient training data causes the neural network to be unable to learn desired function, resulting in performance decline. The conclusion of re-enactment in the paper shows that the closed - loop technique preparing procedure gives a precise way to deal with the preparation of NN. The utilization of trained network in a nonlinear control technique is displayed to furnish unrivalled execution in examination with NN got from an open-loop training plan.

Frank L. Lewis [9] repeatable NN regulator plan algorithms for inflexible robot arms, force control, and parallel connection robot arms were given by the author. The method produces a multi-loop intelligent control structure in each situation, using neural networks in some loops. The restricting presumptions of common versatile regulators, like linearity in the boundaries, are not needed by these regulators. Control frameworks that recreate the human fitness for figuring out how to convey the right info flags that outcome in an ideal reaction without broad comprehension of the framework elements are conceivable utilizing NN regulators.

Amer Mahdy Hamadi [10] developed a mathematical model for a quadrotor using Simulink. To manage the framework, four PID regulators were planned and added to the model, and the framework was actually settled. Two CNN structures were chosen to carry out two different jobs. To detect impediments in front of a quadrotor, the first one required a 15-layer network. The principal CNN structure created utilizes a 15-layer NN to distinguish obstructions before a quadrotor. The second assignment utilized a transfer deep learning method from a pre-prepared AlexNet to perceive an administrator's motion, which brought about an incredible achievement rate. Both the undertakings were carried out on quadrotor utilizing an Arduino microcontroller and a point of interaction NN to control quadrotor.

Marzieh Y. et al. [11] This study gives a uniform control solution for the following issue of a 5-DOF upper-limb exoskeleton robot. Within the context of potential flaws and limited external irritants, these flaws and irritants are taken into consideration. Combining a NN feed-forward term with the more recently developed RISE feedback control mechanism is the approach that has been suggested. Without the need for a full system model, the feed forward NN can approximate the nonlinear dynamics of the robot and account for uncertainty. At the same time, the RISE feedback term gets rid of the mistake caused by the NN approximation. The outputs of the suggested control scheme are compared to those of a PD-based controller as well as a NN-based controller. The work also included a discussion of the dynamics of a five degrees of freedom (DOF) upper-limb exoskeleton robot. The proposed methodology was successful in removing the NN reconstruction error that was caused by the NN approximation.

Yoshikawa T. et al. [12] the authors offer a technique for building regulators for dynamic position/force half and half control of robot controllers. This plan utilizes nonlinear state input to linearize the controller elements. Proposed plan comprises of two stages: initial step is the linearization of the controller elements by nonlinear state input. The subsequent advance is the plan of position and power regulators for the linearized model which assesses both the order reaction and the heartiness of the regulators to displaying mistakes and aggravations.

N. Harris McClamroch et al [13] utilizing a numerical models, conditions for adjustment of a shut circle obliged robot are created. A nonlinear regulator in light of a variety of the determined force strategy was utilized to lay out worldwide settling conditions. A direct regulator with a predetermined input structure was likewise used to lay out nearby balancing out conditions. The work additionally incorporates an examination of the qualities of closed - loop frameworks utilizing the recommended regulator structure. He has exhibited that under consistent power unsettling influences and vulnerability in the requirement work, 'high gain displacement feedback loops' lower the consistent state dislodging guideline blunder. Likewise 'high gain force feedback loop' lessen the consistent state imperative power guideline mistakes for steady power aggravations and vulnerability in the limitation work.

J Zhao-Hui et al. [14] a powerful direction following control strategy for modern robot controllers utilizing a linear feedback regulator and NN regulator have been proposed in the work. The manifold was created to specify the robot system's required trajectory tracking capabilities. The linear controller was then created based on the feedback in relation to the manifold. The NN is a three-layer feed-forward network that was acquainted with the framework in corresponding with the linear regulator to set up the control framework. A straightforward dynamic model was utilized to develop the learning law of loads for NN.

IV. CASE STUDY

Following are the case studies of the main CNN developed:

A. ResNet [22]

The ILSVRC 2015 winner, Microsoft Residual Network, was created by Kaiming He and others. The Networks, as GoogleNet eliminates completely associated layers at network's end. It includes more than 150 "Ultra Deep" layers and were trained for two to three weeks on eight GPU workstations. Strategy on “ImageNet-2012” characterization dataset that comprises 1k classes is assessed for study. The models were created on the 1.28 million planning pictures and evaluated on the 50k endorsement pictures. Eventual outcome was acquired on 100k test pictures, announced by test server. Below fig. show the example of ImageNet, VGG-19 model, and residual network with 34 parameter layers. Fig. 8 is the example of structure for ImageNet network.

Layer name	Output size	18xlayer	34-layer	50-layer	101-layer	152-layer
Conv1	112x112	7x7,64, stride 2				
Conv2_x	56x56	32x3 max pool, stride 2				
		$\begin{bmatrix}3\times3, & 64\\3\times3, & 64\end{bmatrix}\times 2$	$\begin{bmatrix}3\times3, & 64\\3\times3, & 64\end{bmatrix}\times 3$	$\begin{bmatrix}1\times1, & 64\\3\times3, & 64\\1\times1, & 256\end{bmatrix}\times 3$	$\begin{bmatrix}1\times1, & 64\\3\times3, & 64\\1\times1, & 256\end{bmatrix}\times 3$	$\begin{bmatrix}1\times1, & 64\\3\times3, & 64\\1\times1, & 256\end{bmatrix}\times 3$
Conv3_x	28x28	$\begin{bmatrix}3\times3, & 128\\3\times3, & 128\end{bmatrix}\times 2$	$\begin{bmatrix}3\times3, & 128\\3\times3, & 128\end{bmatrix}\times 4$	$\begin{bmatrix}1\times1, & 128\\3\times3, & 128\\1\times1, & 512\end{bmatrix}\times 4$	$\begin{bmatrix}1\times1, & 128\\3\times3, & 128\\1\times1, & 512\end{bmatrix}\times 4$	$\begin{bmatrix}1\times1, & 128\\3\times3, & 128\\1\times1, & 512\end{bmatrix}\times 4$
Conv4_x	14x14	$\begin{bmatrix}3\times3, & 256\\3\times3, & 256\end{bmatrix}\times 2$	$\begin{bmatrix}3\times3, & 256\\3\times3, & 256\end{bmatrix}\times 6$	$\begin{bmatrix}1\times1, & 256\\3\times3, & 256\\1\times1, & 1024\end{bmatrix}\times 6$	$\begin{bmatrix}1\times1, & 256\\3\times3, & 256\\1\times1, & 1024\end{bmatrix}\times 23$	$\begin{bmatrix}1\times1, & 256\\3\times3, & 256\\1\times1, & 1024\end{bmatrix}\times 36$
Conv5_x	7x7	$\begin{bmatrix}3\times3, & 512\\3\times3, & 512\end{bmatrix}\times 2$	$\begin{bmatrix}3\times3, & 512\\3\times3, & 512\end{bmatrix}\times 3$	$\begin{bmatrix}1\times1, & 512\\3\times3, & 512\\1\times1, & 2048\end{bmatrix}\times 3$	$\begin{bmatrix}1\times1, & 512\\3\times3, & 512\\1\times1, & 2048\end{bmatrix}\times 3$	$\begin{bmatrix}1\times1, & 512\\3\times3, & 512\\1\times1, & 2048\end{bmatrix}\times 3$
	1x1	Average pool, 1000-d fc, SoftMax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

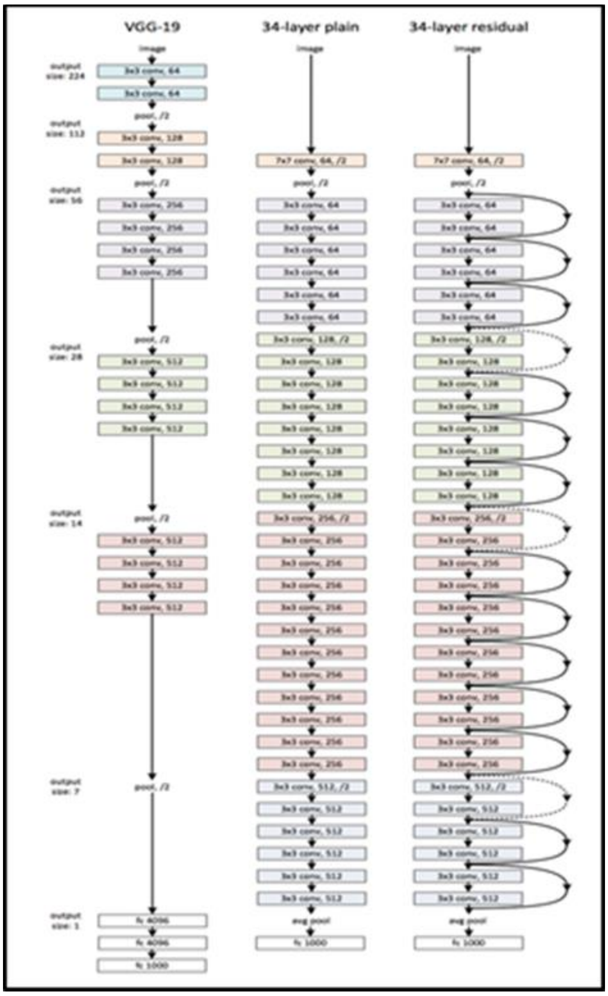

Fig. 8. Example of Network architecture for ImageNet [22].

The key contribution of this paper was demonstrating that network depth has an important impact in network performance. There are 16 CONV/FC layers and features in network. Rather than involving generally huge responsive fields in the first conv. layer, or 7×7 with stride 2, author utilized tiny 3×3 open fields all through the entire net, which are convolved with the contribution at each pixel.

B. GoogleNet[15]

This paper presented another module called the inception which utilizations normal pooling rather than FCL Net, which assists with diminishing an enormous number of boundaries. There are additionally a few subsequent forms of GoogleNet, most as of late Inception-v7. It was prepared on various top of the line GPUs for seven days.

TABLE I. GOOGLENET INCARNATION OF THE INCEPTION ARCHITECTURE [15]

type	Patch size/ stride	Output size	depth	#1 × 1	#3 × 3 reduce	#3 × 3	#5 × 5 reduce	#5 ×5	Pool Proj	params	ops
convolution	7 × 7 /2	112 × 112 × 64	1							2.7K	34M
max pool	3 × 3 /2	56 × 56 × 64	0								
convolution	3 × 3 /1	56 × 56 × 192	2		64	192				112K	360M
max pool	3 × 3 /2	28 × 28 × 192	0								
inception (3a)		28 × 28 × 256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28 × 28 × 480	2	128	128	192	32	96	64	380K	304M
max pool	3 × 3 /2	14 × 14 × 480	0								
inception (4a)		14 × 14 × 512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14 × 14 × 512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14 × 14 × 512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14 × 14 × 528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14 × 14 × 832	2	256	160	320	32	128	128	840K	170M
max pool	3 × 3 /2	7 × 7 × 832	0								
inception (5a)		7 × 7 × 832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7 × 7 × 1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7 × 7 /1	1 × 1 × 1024	0								
dropout (40%)		1 × 1 × 1024	0								
linear		1 × 1 × 1000	1							1000K	1M
SoftMax		1 × 1 × 1000	0								

V. RESULTS

The comparison between the CNN that are discussed in the case studies is given in this section. Number of layers present in each of the CNN is different and thus affecting the accuracy of the network. The architecture of Resnet consist of 152-layer deep CNN while yet having lower intricacy than VGGNet. Correspondingly VGGNet comprises of 19-layers and GoogleNet comprises of 22-layer profound CNN, yet the boundaries were diminished to 4 million from the earlier CNN (VGGNet) to 4 million. The FLOP (floating point operation) values of CNNs are given in the table to show how many operations are required to run a single instance of a given model.

TABLE II. COMPARISON BETWEEN RESNET (152), VGGNET (19), GOOGLENET.

Year	CNN	Developed by	Accuracy	No. of Para-meter	FLOP	Salient Feature
2015	Res Net	Kaiming He	96.4%	60.3 million	11.3 billion	Shortcut connections
2014	VGG Net	Simonyan, Zisserman	92.7%	138 million	19.6 billion	Fixed-size kernels
2014	Goog leNet	Google	93.33%	4 million	1.5 billion	Less Testing set

REFERENCES

[1] C. Liu, B. Zheng, C. Wang, Y. Zhao, S. Fu, and H. Li, “CNN-based vision model for obstacle avoidance of mobile robot,” MATEC Web Conf., vol. 139, pp. 4–7, 2017, doi: 10.1051/matecconf/201713900007.

[2] X. Ming, X. Nanfeng, Z. Mengjun, and Y. Qunyong, “Optimized Convolutional Neural Network-Based Object Recognition for Humanoid Robot,” J. Robot. Autom., vol. 4, no. 1, pp. 122–130, 2020, doi: 10.36959/673/363.

[3] Y. Zhao and C. C. Cheah, “Vision based neural network control for constrained robots with constraint uncertainty,” IET Control Theory App, vol. 02, no. 10, pp. 906–916, 2008, doi: 10.1049/iet-cta:20070316.

[4] N. Hogan, “Impedance Control : An Approach to Manipulation : Part I - Theory,” J. Dyn. Syst. Meas. Control, vol. 107, no. March 1985, pp. 1–7, 2017.

[5] N. Mogan, “Impedance Control : An Approach to Manipulation : Part II -Implementation,” J. Dyn. Syst. Meas. Control, no. MARCH 1985, 2017.

[6] N. Hogan, “Impedance Control : An Approach to Manipulation : Part III -Applications,” J. Dyn. Syst. Meas. Control, vol. 107, no. March 1985, pp. 17–24, 2017.

[7] A. Thomas and J. Hedley, “FumeBot: A deep convolutional neural network controlled robot,” Robotics, vol. 8, no. 3, 2019, doi: 10.3390/robotics8030062.

[8] T. S. Low, T. H. Lee, and H. K. Lim, “A Methodology for Neural Network Training for Control of Drives with Nonlinearities,” IEEE Trans. Ind. Electron., vol. 40, no. 2, pp. 243–249, 1993, doi: 10.1109/41.222646.

[9] F. L. Lewis and A. Yesildirek, “Neural Net Robot Controller with Guaranteed Tracking Performance,” IEEE Trans. Neural Networks, vol. 6, no. 3, pp. 703–715, 1995, doi: 10.1109/72.377975.

[10] A. M. Hamadi, “Autonomous Quadrotor Control Using Convolutional Neural Networks,” pp. 1–58, 2019.

[11] M. Yazdanzad, A. Khosravi, R. Ghaderi, and P. Sarhadi, “Combined Neural Network Feedforward and RISE Feedback Control Structure for a 5 DOF Upper-limb Exoskeleton Robot with Asymptotic Tracking Marzieh,” J. Adv. Comput. Res., vol. 6, no. 1, pp. 47–62, 2014.

[12] C. Finn, S. Levine, T. Darrell, and P. Abbeel, “End-to-End Training of Deep Visuomotor Policies,” J. Mach. Learn. Res., vol. 17, pp. 1–40, 2016.

[13] N. H. McClamroch and D. Wang, “Feedback Stabilization and Tracking of Constrained Robots,” IEEE Trans. Automat. Contr., vol. 33, no. 5, pp. 419–426, 1988.

[14] Z. H. Jiang and T. Ishita, “A neural network controller for trajectory control of industrial robot manipulators,” J. Comput., vol. 3, no. 8, pp. 1–8, 2008, doi: 10.4304/jcp.3.8.1-8.

[15] C. Szegedy, S. Reed, P. Sermanet, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” pp. 1–12, 2014.

[16] X. Li and C. C. Cheah, “Adaptive neural network control of robot based on a unified objective bound,” IEEE Trans. Control Syst. Technol., vol. 22, no. 3, pp. 1032–1043, 2014, doi: 10.1109/TCST.2013.2293498.

[17] L. Cheng, Z. G. Hou, and M. Tan, “Adaptive neural network tracking control for manipulators with uncertain kinematics, dynamics and actuator model,” Automatica, vol. 45, no. 10, pp. 2312–2318, 2009, doi: 10.1016/j.automatica.2009.06.007.

[18] Y. Zhang, J. Gao, and H. Zhou, “Breeds Classification with Deep Convolutional Neural Network,” PervasiveHealth Pervasive Comput. Technol. Healthc., pp. 145–151, 2020, doi: 10.1145/3383972.3383975.

[19] S. Lyu and C. C. Cheah, “Vision based Neural network control of robot manipulators with unknown sensory Jacobian matrix,” IEEE/ASME Int. Conf. Adv. Intell. Mechatronics, AIM, vol. 2018-July, pp. 1222–1227, 2018, doi: 10.1109/AIM.2018.8452467.

[20] C. Liang, S. Bhasin, K. Dupree, and W. E. Dixon, “A Force Limiting Adaptive Controller for a Robotic System Undergoing a Non-Contact to Contact Transition,” Proc. 46th IEEE Conf. Decis. Control, pp. 3555–3560, 2007.

[21] P. M. Patre, W. Mackunis, K. Kaiser, and W. E. Dixon, “Asymptotic Tracking for Uncertain Dynamic Systems via a Multilayer NN Feedforward and RISE Feedback Control Structure,” Proc. 2007 Am. Control Conf., pp. 5989–5994, 2007.

[22] J. S. Kaiming He, Xiangyu Zhang, Shaoqing Ren, “Deep Residual Learning for Image Recognition,” Indian J. Chem. - Sect. B Org. Med. Chem., pp. 770–778, 2006, doi: 10.1002/chin.200650130.

[23] J. C. Larsen and N. J. Ferrier, “A Case Study in Vision Based Neural Network Training for Control of a Planar , Large Deflection , Flexible Robot Manipulator,” Conf. Intell. Robot. Syst., pp. 2924–2929, 2004.

[24] R. Feraud and F. Clerot, “A methodology to explain neural network classification,” Neural Networks, vol. 15, pp. 237–246, 2002, doi: 10.1016/S0893-6080(01)00127-7.

[25] H. D. Patiño, R. Carelli, and B. R. Kuchen, “Neural networks for advanced control of robot manipulators,” IEEE Trans. Neural Networks, vol. 13, no. 2, pp. 343–354, 2002, doi: 10.1109/72.991420.

[26] Y. H. Kim and F. L. Lewis, “Neural Network Output Feedback Control of Robot Manipulators,” IEEE Trans. Robot. Autom., vol. 15, no. 2, pp. 301–309, 1999.

[27] C. C. Cheah, S. Kawamura, and S. Arimoto, “Feedback Control for Robotic Manipulator with Uncertain Kinematics and Dynamics,” Int. Conf. Robot. Autom., no. May, pp. 3607–3612, 1998.

[28] R. V Patel, K. Khorasani, and H.A. Talebi, “Inverse Dynamics Control of Flexible-Link Manipulators Using Neural Network,” Int. Conf. Robot. Autom., pp. 806–811, 1998.

[29] F. L. Lewis, “Neural Network Control of Robot Manipulators,” IEEE Expert. Syst. their Appl., pp. 64–76, 1996.

[30] S. Jung, “A Study On Neural Network Control of Robot Manipulators,” Robotica, pp. 1–21, 1996, doi: 10.1017/S0263574700018890.

[31] H. Ali, K. Aizat, K. Yerkhan, T. Zhandos, and O. Anuar, “Vision-based Robot Manipulator for Industrial Applications,” Procedia Comput. Sci., vol. 133, pp. 205–212, 2018, doi: 10.1016/j.procs.2018.07.025.

[32] H. Sekkat, S. Tigani, R. Saadane, and A. Chehri, “Vision-based robotic arm control algorithm using deep reinforcement learning for autonomous objects grasping,” Appl. Sci., vol. 11, no. 17, pp. 1–14, 2021, doi: 10.3390/app11177917.

[33] B. Espiau, F. Chaumette, and P. Rives, “A New Approach to Visual Servoing in Robotics,” IEEE Trans. Robot. Autom., vol. 8, no. 3, pp. 313–326, 1992, doi: 10.1109/70.143350.

[34] J. K. Mills and A. A. Goldenberg, “Force and Position Control of Manipulators During Constrained Motion Tasks,” IEEE Trans. Robot. Autom., vol. 5, no. 1, pp. 30–46, 1989.

[35] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–14, 2015.

[36] O. Khatib, “A Unified Approach for Motion and Force Control,” IEEE J. Robot. Autom., vol. 1, pp. 43–53, 1987.

[37] M. H. Raibert and J. J. Craig, “Hybrid position/force control of manipulators,” J. Dyn. Syst. Meas. Control. Trans. ASME, vol. 103, no. 2, pp. 126–133, 1981, doi: 10.1115/1.3139652.