A STUDY ON DEEP LEARNING MODELS AND ALGORITHMS USED IN COMPUTER VISION

Dr. Rashmi Dongre, Assistant Professor
Tilak Maharashtra Vidyapeeth, Mail: rashmibichkar15@gmail.com
Dr. Asmita Namjoshi , Assistant Professor
Tilak Maharashtra Vidyapeeth, Mail: asmita03@gmail.com

ABSTRACT :
One of the most interesting and popular subfields of artificial intelligence is computer vision. Without even realizing it, you have probably come across and utilized computer vision apps. Computer vision techniques are revolutionizing industries worldwide, whether it is for electronic deposit picture processing or crop quality control via image classification. The goal of computer vision is to mimic the intricate functioning of the human visual system so that a machine or computer can recognize and process various items in pictures and videos in a similar manner to a human. Computer vision algorithms are now able to process enormous amounts of visual data thanks to developments in deep learning, neural networks, artificial intelligence, and machine learning. In certain tasks, such as object detection and labeling, computer vision algorithms have outperformed humans in terms of speed and accuracy. Deep learning methods are now mostly applied to computer vision. This study investigates many applications of deep learning in computer vision.

Keywords: Computer vision, Object detection, Neural Network, Deep learning

IINTRODUCTION :

The field of computer vision is devoted to the interpretation and comprehension of images and videos. It is employed in the process of teaching computers to "see" and to use visual data to carry out visual tasks that humans are able to do. The purpose of computer vision models is to interpret visual data by using characteristics and contextual information that are discovered during training. This makes it possible for models to decipher pictures and videos and use their interpretations for tasks involving prediction or judgement [8]. Computer vision and image processing are not the same, despite their shared interest in visual data. Image processing is the process of improving or changing pictures to get a different outcome. It can involve cropping, boosting resolution, obscuring sensitive data, and adjusting brightness or contrast. Image processing and computer vision vary in that the former does not always involve content identification [2]. After analyzing specific parameters in pictures and videos, computer vision algorithms apply their interpretations to tasks involving prediction or decision-making. Deep learning methods are now mostly applied to computer vision. The various applications of deep learning for computer vision are examined in this article. You will discover the benefits of employing convolutional neural networks (CNNs), in particular, since they offer a multi-layered design that enables neural networks to concentrate on the most important characteristics in the image [3]. Computer vision (CV), natural language processing (NLP), speech and video recognition (V/SP), and finance and banking (F&B) are among the current applications of DL. The DL technique has challenges in the early stages of CV development because of computer memory, CPU, and GPU restrictions [5]. Thus, the majority of academics are investigating the use of ML in resumes. In the meantime, numerous CV techniques have been put out, including Support Vector Machine (SVM), Expectation-Maximization (EM), K-Nearest Neighbour (KNN), Decision Tree, Boosting, Random Forest, Haar Classifier, K-means, and Naive Bayes classifier.

The past few decades have seen a tremendous advancement in deep learning, which may be generally categorized into ten groups based on algorithm and architecture: Long Short-Term Memory Networks (LSTMs), Autoencoders, Radial Basis Function Networks (RBFNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Generative Adversarial Networks (GANs), Self-Organizing Maps (SOMs), Deep Belief Networks (DBNs), Autoencoders, and Multilayer Perceptrons (MLPs) [6]. This research papers explores different ways of using deep learning for computer vision.

NEED FOR STUDY :

The advancement of deep learning technology has made it possible to construct computer vision models that are more intricate and accurate. The integration of computer vision applications is becoming increasingly beneficial as these technologies advance. Speech recognition, language translation, and image categorization have all benefited from deep learning. Without the need for human assistance, it can be applied to resolve any pattern recognition issue. Deep learning is powered by artificial neural networks, which include multiple layers [9]. Hence it is beneficial for implementing deep learning models and algorithms for computer vision applications

REVIEW OF LITERATURE :

Over the past ten years, there has been an unparalleled surge in computer vision research, primarily attributed to the development of deep learning technologies, particularly convolutional neural networks (CNN) [12]. They have surpassed all prior state-of-the-art records in image processing, making them the undisputed finest technical option for these tasks.

Over time, the discipline of computer vision has seen tremendous change as new methods and algorithms are created to increase the precision and effectiveness of image and video analysis [14]. Several significant turning points in the development of computer vision algorithms include:

- **Early computer vision:** The earliest computer vision algorithms were created in the 1960s and 1970s, with a primary emphasis on image processing and pattern recognition methods. The capacity of these early algorithms to identify and comprehend complicated scenarios and images was constrained.
- **Machine learning:** Machine learning techniques were introduced to computer vision in the 1980s and 1990s, enabling algorithms to learn from examples and gradually enhance their performance. As a result, algorithms for tasks like object identification and image categorization were developed.
- **Deep learning:** Deep learning methods, which employ neural networks to identify patterns in pictures and videos, first appeared in the 2000s. The accuracy of computer vision tasks, such object detection and facial recognition, has increased dramatically because to these methods.
- **Real-time systems:** Recent developments in hardware and software have made it possible to run computer vision algorithms in real-time, which opens up a variety of applications for them, including robotics, drones, and self-driving cars [4].

A growing number of real-world applications, including those in retail, security, entertainment, and medical imaging, are being made possible by advances in computer vision. Computer vision algorithms are improving in their ability to comprehend an image's context and carry out increasingly difficult tasks including action detection, semantic segmentation, and scene interpretation [13]. Among the aforementioned, this research study expands on the use of deep learning techniques, particularly in computer vision [1]. When it comes to difficult computer vision issues like face recognition, object identification, and image classification, deep learning techniques can produce cutting-edge outcomes.

RESEARCH FINDINGS

4.1 Deep Learning Models

4.1.1 ResNet-50 for Image Classification

A breakthrough in deep learning for computer vision, ResNet-50 is a variation of the ResNet (Residual Network) model, which has shown promise in picture classification tasks in particular. The network's layer count is indicated by the "50" in ResNet-50; it has 50 layers deep, a considerable increase over earlier models.

ResNet-50 uses residual blocks as its central concept. These blocks give the model the ability to use "skip connections" or "shortcut connections" to bypass one or more layers.

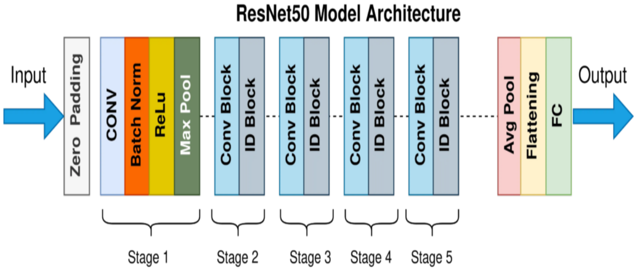


Fig: 2 Res Net 50 Model Architecture

The vanishing gradient problem, which occurs frequently in deep networks and makes it challenging to train very deep networks, is solved by this architecture. Gradients get smaller and smaller as they backpropagate through layers [15]. When it comes to computational resources, ResNet-50 is more efficient than other deep models. Because of its outstanding accuracy on a variety of image classification benchmarks, like as ImageNet, it is well-liked by both the scientific community and business.

The field of image categorization has evolved tremendously with ResNet-50. Many further advancements in computer vision and deep learning are based on its architecture. ResNet-50 made it possible to train deeper neural networks, which in turn increased the accuracy and complexity of tasks that computer vision systems can perform.

4.1.2: YOLO (You Only Look Once) Model

In computer vision, the YOLO (You Only Look Once) model is a ground-breaking method, especially for jobs involving object recognition. Because to YOLO's remarkable speed and effectiveness, real-time object identification is now possible.

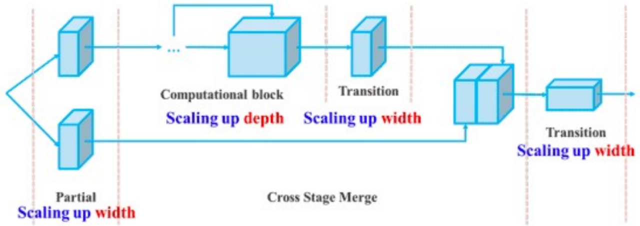


Fig: 3 YOLO Architecture for Object detection

YOLO does both at once using a single convolutional neural network (CNN). It is able to process photos instantly because of this unified methodology. Because of its architecture, YOLO can process images very quickly, which makes it appropriate for applications requiring real-time detection, such autonomous cars and video surveillance.

Yolo has made a substantial contribution to the field of deep learning for computer vision. Its real-time, accurate, and efficient object detection capabilities have created a plethora of opportunities for real-world applications that were previously constrained by slower detection speeds. Its change over time also represents the quick development and creativity in the computer vision area's deep learning field.

4.1.3: VISION TRANSFORMERS :

This model uses transformers, which were first created for natural language processing, to help in picture identification and classification. In order to do this, an image must be divided into fixed-size patches, which must then be embedded, given positional information, and fed into a transformer encoder. To process these image patches and carry out classification, the model's architecture combines Multi-head Attention Networks with Multi-Layer Perceptrons.

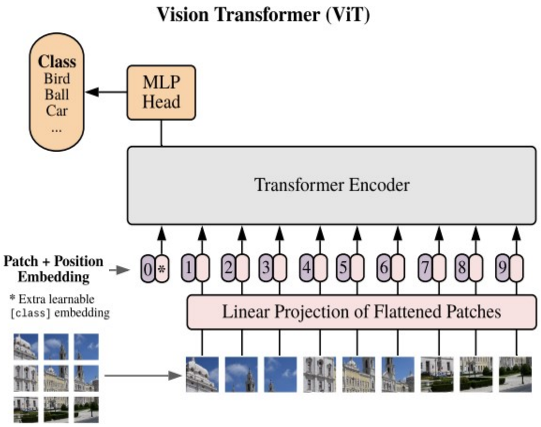

Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0*
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches

Fig 4: Vision Transformer (ViT)

ViT treats an image as a series of patches by partitioning it into patches and linearly embedding them. Positional embeddings are appended to the patch embeddings in order to preserve the spatial relationship of image portions. To grasp the relationships between various patches and concentrate on important areas within the image, it makes use of a multi-head attention network. ViT's learnable class embedding improves its picture classification performance. When it comes to image classification, ViT models have proven to be significantly more accurate and computationally efficient than conventional CNNs. A major development in computer vision, the Vision Transformer provides a potent substitute for traditional CNNs and opens the door for increasingly complex image processing methods. Because Vision Transformers (ViTs) are accurate and efficient at handling complicated picture data, they are finding growing use in a wide range of real-world applications across several fields.

4.1.4: STABLE DIFFUSION V2 MODEL

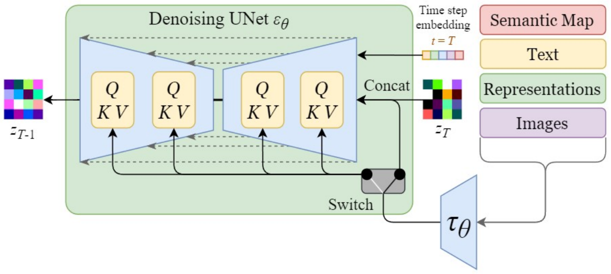


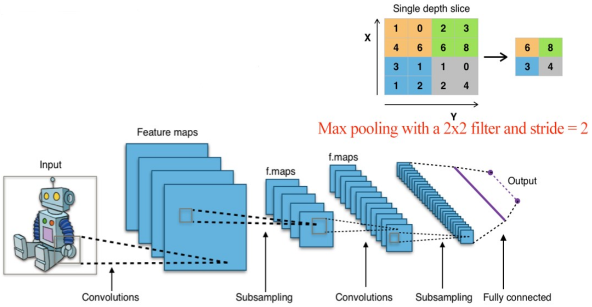
Fig 5: Stable Diffusion V2 Model Architecture

Strong text-to-image models are incorporated into Stable Diffusion V2, which uses a new text encoder (OpenCLIP) to improve the quality of generated images. These models offer huge gains over earlier iterations, producing images with resolutions as high as 768×768 pixels and 512×512 pixels.

The Upscaler Diffusion model, a noteworthy feature in V2, can boost image resolution by a factor of four. When paired with text-to-image models, this capability enables the conversion of low-resolution photos into much higher-resolution equivalents, up to 2048 x 2048 pixels or more. A new text-guided inpainting model in Stable Diffusion V2 makes it possible to quickly and intelligently alter specific areas of an image. This facilitates precise image editing and enhancement. The improved ability of Stable Diffusion V2 to produce crisp, high-resolution images from written descriptions marks a significant advancement in computer-generated imagery. This creates new opportunities in a number of industries, including content generation, graphic design, and digital art. More complex image modification and manipulation are possible because to Stable Diffusion V2's enhanced inpainting capabilities. This may be useful in industries like advertising, where it's frequently necessary to make quick, deft image adjustments. The accessibility and sophisticated capabilities of Stable Diffusion V2 could raise the bar for future developments and applications in the AI and computer vision communities.

4.1.5: PYTORCH AND KERAS

The open-source machine learning package PyTorch was created by Facebook's AI Research group. Due to its versatility, user-friendliness, and inbuilt support for dynamic computation graphs, it is especially well-suited for research and prototyping. Additionally, PyTorch has robust support for GPU acceleration, which is necessary for effectively training massive neural networks [7].Keras is an easy-to-use, high-level neural network API that is now integrated with Google's AI framework, TensorFlow. With its intuitive UI, Keras, which was first created as a standalone project, aims to facilitate quick experimentation and prototyping. While abstracting away many of the intricate intricacies, it supports all the fundamental elements required to build deep learning models, making it relatively user-friendly for novices. Both frameworks are widely utilized for a wide range of machine learning and artificial intelligence applications, from straightforward regression models to intricate deep neural networks, in both academic and commercial contexts.



1	0	2	3
4	6	6	8
3	1	1	0
1	2	2	4

6	8
3	4

Fig 6: PyTorch Model Architecture

The Python-written library Keras is available as open-source software. It is made to make deep neural network experimentation quick and easy. The deep learning framework Keras is versatile, modular, and easy to use. Furthermore, it is a high-level neural network API that can encapsulate a low-level API.

Several backend neural network computation engines are supported by the Keras. It cannot do the low-level calculation. It does this by utilizing a different library. Faster, more user-friendly, and modular in design are the goals of the Keras. Additionally, Keras assembles our model with optimizer and loss functions, training it with the appropriate function. The Keras is compatible with several backend engines. Tensor Flow serves as its main backend and Google is its main backer. We can modify our

$HOME/.Keras /Keras.json file and specify a different backend name, such Theano or CNTK, if we wish to switch the backend in Keras. In Keras, all low-level computation, including Tensor products and convolution, is done by the backend [10].

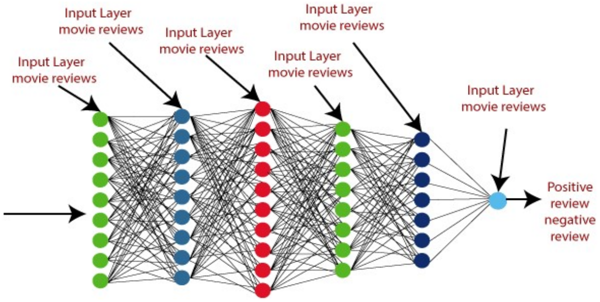

Input Layer movie reviews
Input Layer movie reviews
Input Layer movie reviews
Input Layer movie reviews
Input Layer movie reviews
Input Layer movie reviews
Positive review negative review

Fig 7: Keras Convolution Model

4.2 DEEP LEARNING METHODS AND DEVELOPMENTS:

4.2.1 Convolutional Neural Networks

In particular, the models of the visual system put out in provided inspiration for Convolutional Neural Networks (CNNs). Neocognitron contains the first computational models based on these local connectivities between neurons and on hierarchically organized image transformations. Neocognitron explains that a type of translational invariance is acquired when neurons with the same parameters are applied on patches of the previous layer at different locations [11]. Convolutional layers, pooling layers, and fully connected layers are the three primary categories of neural layers found in a CNN. Every kind of layer has a distinct function.CNN architecture for an object detection task in a picture is displayed in Figure 8. When a CNN reaches its final fully connected layers, each layer converts the input volume to an output volume of neuron activation, which produces a mapping from the input data to a 1D feature vector. In computer vision applications including face recognition, object identification, robotics vision, and self-driving cars, CNNs have shown to be incredibly successful.

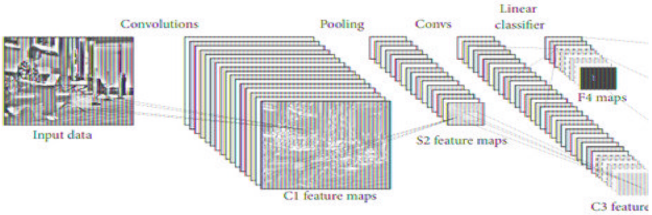


Fig 8: CNN

CONVOLUTIONAL LAYERS:

A CNN uses different kernels in the convolutional layers to convolve the entire image as well as the intermediate feature maps, producing a variety of feature maps.

Pooling Layers:

The purpose of pooling layers is to minimize the input volume's breadth and height so that the subsequent convolutional layer can work with it. The volume's depth dimension is unaffected by the pooling layer. Because the reduction in size results in a simultaneous loss of information, the operation carried out by this layer is also known as sub sampling or down sampling. Nonetheless, the network benefits from this kind of loss since it prevents over fitting and results in less computing overhead for the network's subsequent layers. Typical pooling and maximum pooling are the most widely employed techniques.

Fully Connected Layers:

Fully connected layers carry out the high-level reasoning in the neural network after a number of convolutional and pooling layers. As their name suggests, neurons in a completely linked layer are totally connected to every activation in the preceding layer. As a result, their activation can be calculated using matrix multiplication and bias offset. The 2D feature maps are subsequently transformed into a 1D feature vector by fully connected layers. The resultant vector may be used as a feature vector for additional processing or it may be fed into a specified number of categories for classification.

4.2.2 DEEP BELIEF NETWORKS AND DEEP BOLTZMANN MACHINES :

The Restricted Boltzmann Machine (RBM) is the learning module used by Deep Belief Networks and Deep Boltzmann Machines, two deep learning models that are members of the "Boltzmann family." Based on stochastic neural networks, the Restricted Boltzmann Machine (RBM) is generative. Directed connections connect the lower layers of DBNs to the directed connections at the top two layers, forming an RBM. Undirected connections exist between every network tier in DBMs. Figure 9 is a graphic representation of DBNs and DBMs. The fundamental properties of DBNs and DBMs will be discussed in the ensuing subsections, after the presentation of their fundamental building block, the RBM.

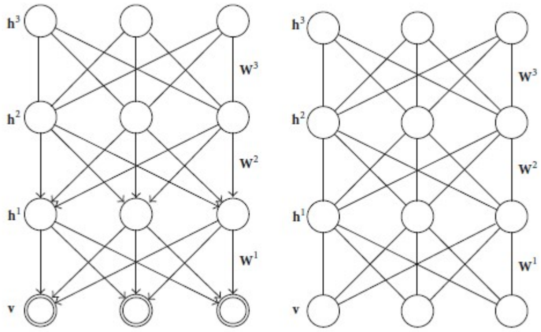


Fig 9: Deep Belief Networks and Deep Boltzmann Machines

Probabilistic generative models called Deep Belief Networks (DBNs) offer a combined probability distribution over observable data and labels. A DBN first initializes the deep network using an effective layer-by-layer greedy learning technique, and then in the follow-up, it jointly fine-tunes all weights with the desired outputs. Graph models known as DBNs are trained to derive a deep hierarchical representation from the training set. Using RBM as a building block, another kind of deep model is called a Deep Boltzmann Machine (DBM). The top two layers of the DBN constitute an undirected graphical model, while the lower layers form a directed generative model. In contrast, all of the connections in the DBM are undirected. This is the difference in architecture between the two models. Units in odd-numbered layers are conditionally independent on even-numbered layers, and vice versa, in DBMs, which feature many levels of hidden units.

Consequently, inference within the DBM is typically unmanageable. Nevertheless, more manageable iterations of the model may result from carefully choosing the interactions between the visible and hidden units. During network training, a DBM jointly trains all layers of a particular unsupervised model. The DBM then utilizes a stochastic maximum likelihood (SML) based technique to maximize the lower bound on the likelihood, rather than maximizing the likelihood directly.

4.2.3 STACKED (DENOISING) AUTOENCODERS :

Similar to how restricted Boltzmann machines are a component of deep belief networks, autoencoders serve as the foundation for stacked autoencoders. Therefore, before discussing the deep learning architecture of Stacked (Denoising) Autoencoders, it is crucial to quickly review the fundamentals of the autoencoder and its denoising variant. In order to enable input reconstruction from r(x), an autoencoder is trained to encode the input x into a representation r(x).Therefore, the autoencoder's goal output is the autoencoder input itself. As a result, the dimensionality of the output vectors and the input vectors are equal. The learned feature is the appropriate code, and during this process, the reconstruction error is minimized. When a single linear hidden layer is employed in conjunction with the mean squared error criterion for network training, the input is projected by the k hidden units within the range of the data's initial k principal components. The autoencoder responds differently from PCA in the event that the hidden layer is nonlinear, and it can capture multimodal characteristics of the input distribution. In order to minimize the average reconstruction error, the model's parameters are optimized. The denoising autoencoder is a stochastic variant of the autoencoder in which the uncorrupted input is still the reconstruction target despite the input being stochastically corrupted. By supplying the latent representation (output code) of the denoising autoencoder of the layer below as input to the current layer, denoising autoencoders can be stacked to create a deep network. Such architecture is pre-trained unsupervised, layer by layer. By reducing the error in reconstructing its input, which is the output code of the preceding layer, each layer is trained as a denoising autoencoder. One advantage of autoencoders as the fundamental unsupervised part of a deep architecture is that, in contrast to RBMs, they permit nearly any parameterization of the layers—as long as the parameters satisfy the training criterion.

5 CONCLUSION

The advancements in computer vision that deep learning has made possible have played a major role in the field's recent boom. Significant success rates in a range of visual understanding tasks have been attained by using the three main types of deep learning for computer vision that have been examined in this paper: CNNs, the "Boltzmann family," which includes DBNs and DBMs, and SdAs. CNNs possess a special capacity called feature learning, which allows them to automatically identify features based on the provided dataset. For some computer vision applications, CNNs' invariance to transformations is a great advantage. Furthermore Developments in deep learning (DL) have advanced at a rapid pace, while advancements in device capabilities—such as processing speed, memory size, power consumption, image sensor resolution, and optics—have enhanced the functionality and economic viability of vision-based applications. Additionally, convolutional neural networks may significantly cut down on computation time by utilizing GPU for processing, something that many other networks are unable to do.

REFERENCES

[1] Adamopoulou, E., & Moussiades, L. (2020). Chatbots: History, technology, and applications. *Machine Learning with Applications*, *2*, Article 100006. http://dx.doi.org/10.1016/j.mlwa.2020.100006

[2] Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep learning for visual understanding: A review. *Neurocomputing*, *187*, 27–48. http://dx.doi.org/10.1016/j.neucom.2015.09.116.

[3] Gando, G., Yamada, T., Sato, H., Oyama, S., & Kurihara, M. (2016). Fine-tuning deep convolutional neural networks for distinguishing illustrations from photographs. *Expert Systems with Applications*, *66*, 295–301. http://dx.doi.org/10.1016/j.eswa. 2016.08.057.

[4] Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. In Proceedings of the 30th international conference on neural information processing systems (nips), Spain, (pp. 379–387).

[5] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the 2017 ieee conference on computer vision and pattern recognition (*cvpr)* (pp. 1251–1258). USA: http://dx.doi.org/10.1109/CVPR.2017.195.

[6] Chai, J., & Li, A. (2019). Deep learning in natural language processing: A state-of-the-art survey. In Proceedings of the 2019 international conference on machine learning and cybernetics (icmlc) (pp. 1–6). Japan: http://dx.doi.org/10.1109/ICMLC48188.2019.8949185.

[7] Bochkovskiy, A., Wang, C. Y., & Liao, H.-Y. M. (2020). YOLOV4: Optimal speed and accuracy of object detection. arxiv preprint arXiv:2004.10934 [Cs, Eess]. http://arxiv.org/abs/2004.10934.

[8] Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8(1), 53. http://dx.doi.org/10.1186/s40537-021-00444-8.

[9] Athanasios Voulodimos ,Nikolaos Doulamis, (2018), Deep Learning for Computer Vision: A Brief Review, Computational Intelligence and Neuroscience Volume 2018, Article ID 7068349, 13 pages https://doi.org/10.1155/2018/7068349

[10] S. Cao and R.Nevatia, "Exploring deep learning based solutions in fine grained activity recognition in the wild," in Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 384–389, Cancun, December 2016.

[11] Badrinarayanan, V., Kendall, A., Cipolla, R., 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 39, 2481–2495.

[12] Diba, A., Sharma, V., Gool, L.V., 2017. Deep temporal linear encoding networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 1541–1550.

[13] Noh, H., Hong, S., Han, B., 2015. Learning deconvolution network for semantic segmentation. 2015 IEEE International Conference on Computer Vision (ICCV), 1520–1528.

[14] Sharma, V., Mir, R.N., 2020. A comprehensive and systematic look up into deep learning based object detection techniques: A review. Computer Science Review 38, 100301. URL: https://www.sciencedirect.com/science/
article/pii/S1574013720304019, doi:https://doi.org/10.1016/j.cosrev.2020.100301.

[15] Zhao, Z.Q., Zheng, P., Xu, S.T., Wu, X., 2019. Object detection with deep learning: A review. IEEE Transactions on Neural Networks and Learning Systems 30, 3212–3232. doi:10.1109/TNNLS.2018.2876865