Generally, a collection of training images is needed to test all the characteristics and grades of each characteristic of the neural network. More the number of images presented for each category, more finely the neural network learns to evaluate these categories. Companies such as Cvedia construct the synthetic datasets that are annotated and optimised for the neural network training. In the absence of other choices, self-produced and labelled images need to be made. Switching a single image to multiple images by rotating, resizing, stretching, and brightening or darkening saves time. Several developers in the deep learning industry open source their image labelling solutions and distribute them free of charge. Labelling, which is especially useful for unmarked datasets, offers a graphical image annotation tool that helps label the artefacts in the bounded boxes inside photos. Alternatively, third parties can be interested in the labelling process. Preparing the training data can become even more important in the light of particular hardware limitations or preferences, as some deep learning tools support only a limited range of hardware [41].

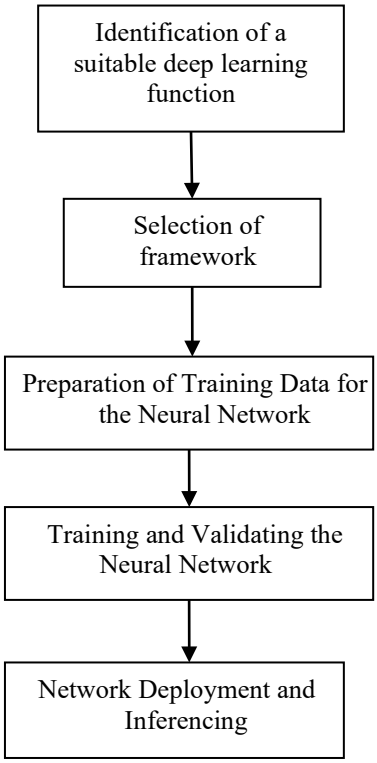


Figure 2. Steps in Training and Deployment of a Neural Network

Step 4: Train and validate the neural network to ensure accuracy

This stage involves configuring and running the scripts on a device until the training process offers an appropriate degree of accuracy for a particular use. Separating training and test data means that the neural network does not mistakenly train the data later used for evaluation. By taking advantage of the transfer of learning or the use of a pre-trained network and re-use it for another mission, this process can be accelerated. For example, a neural network already trained for feature extraction can only need a new collection of images to recognise a new feature. Frameworks like Caffe2 and TensorFlow provide open, pre-trained networks [42]. Neural networks learn the features directly from the data from which they are trained, so that the specialists do not need to bring out the features manually. There is a range of graphical user interface-based software options for the neural network training, such as Matrox Imaging Library (MIL) X, which works with the different frameworks and hence make the training and deployment process very intuitive even for less time [36].

Step 5: Deploy the neural network and run inference on new data

This last step includes deploying a qualified neural network on the selected hardware to assess the output and collect the field data. The first few phases of inference, preferably used in the field to gather the additional test data, can provide the training data for the future iterations. Cloud deployment provides the substantial savings on hardware costs and the opportunity to easily scale and install and distribute the changes at different locations. However, internet connectivity problems can cause the critical errors, and cloud deployment has a high latency compared to the edge deployment. In order to successfully deploy a trained model, there are two initial jobs.

1) First job is to provide our model an input that it expects.

2) Second job is to provide our end user an output that is useful.

The input that the network expects is determined both by its architecture and by how it was trained. The deployment requires writing code to convert the input we have to the input the network expects. The output that our network generates is determined by its architecture and what it learned. Deployment also requires writing code to convert the output that is generated to the output our end user expects.

5. Performance Analysis and Results

A dataset is a set of related examples used to train and test a model. A dataset is a collection of examples related to a specific topic or domain, and it generally aims to serve one or more applications. Because a dataset can be labelled, it is ideal for training and testing the supervised models. Unlabelled datasets, on the other hand, are used to train unsupervised models.

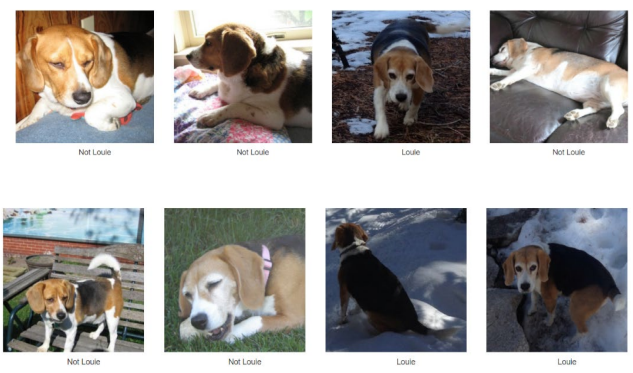
Image-Net is a well-known computer vision dataset that aided in the rise of deep learning. Image-net is commonly used for object recognition and classification. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) is a large-scale evaluation of algorithms for object detection and image classification. One high-level motivation is to enable the researchers to compare progress in detection across a

broader range of objects, making use of the costly labelling effort. Another reason is to track the advancement of computer vision for large-scale image indexing for retrieval and annotation. The most popular challenge is the image classification and localization task. It is available on Kaggle.

Deep learning presents additional challenges, despite the expert knowledge required in this multidisciplinary field of performance analysis. Deep learning models are diverse, encompassing a wide range of problem domains, models, and datasets. The pace of this field is quick, and cutting-edge models evolve every few months. Deep learning software and hardware systems are diverse and rapidly evolving, as are performance metrics such as accuracy, time to train, and inference latency. Metrics must be carefully chosen. To avoid the bias and extract deep insights, performance analysis must be carried out with care. A comprehensive set of benchmarks representing the state-of-the-art and emerging workloads and datasets should be included in a proper performance analysis methodology. The researchers must have a solid understanding of the resource requirements and bottlenecks associated with the cross-stack workloads. Meanwhile, researchers must be aware of the new workloads.

Image classification and training model is created by choosing (a) a dataset of images as shown in figure 3 (b) a neural network, and (c) how many training epochs to run. This highlights two concepts. Deep Neural Networks are the adaptable techniques motivated by the human mind that allow the specialists to use the training strategies motivated by the human learning. Deep Neural Networks fall very short of the human brain. After one epoch, the proposed model is predicting no better than chance. Image Classification is the process of recognizing and categorising the images in either of the various pre-decided separate classes. As a result, image classification applications can define what is shown in a picture and what can differentiate one object from the other.

In image classification with position, an image is placed in any class and then a bounding box is drawn in the vicinity of an object to indicate where it is placed in that image. The detection of objects categorizes the several objects in an image and displays the position of every object with those bounding boxes. So, it is basically a modification of the image recognition with the localization for a number of objects. The data collection "Beagles Images" satisfies this criterion. After preparation, the data set Model is obtained.



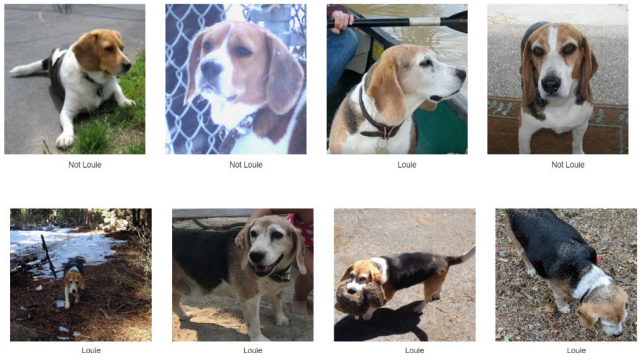


Figure 3. Dataset of Images

Each time the image is displayed, the output or prediction is produced as to whether the image contains a specific puppy or not. The model then produces a forecast that doesn't look any better than a chance. This time, the preparation takes a little longer (between 3 and 4 minutes).

While it trains, looking at the two graphs (figure 4 and figure 5), the two main concepts are illustrated - Learning Rate and Loss Rate. Learning rate is the rate at which each weight changes during training. Each weight is moving in the direction that reduces the loss at a value multiplied by the learning rate. The learning rate decreases throughout the training session because the network is getting closer to its ideal solution. At first, the network being trained knows nothing about what it may see. After a few epochs, the weights are getting better and hence it is important for the network to be less reactive to each image it sees. A training job is set up from this pre-trained model starting at the learning rate of 0.0001 as shown in figure 4. Training loss is calculated in the course of every epoch as shown in figure 5. While validation loss is computed after each epoch, training loss is regularly outlined during the whole epoch. But the validation metrics are calculated over the validation set only after the completion of the present training epoch. Sum () is a NumPy function that returns the sum of all array elements in the NumPy array. This sum () function can be used to count the number of pixels based on the criteria specified which is shown in figure 6 .The mean of all the images is taken (Mean is chosen for the prediction mainly because it can utilize all of the possible values present in the large dataset) as shown in figure 7 to identify the correct prediction about Louie image which is shown in figure 8. In figure 7, mean of image is calculated by dividing the sum of pixel values by the total count and subtracting the number of pixels in the dataset computed. In order to predict the Louie image as shown in figure 8, load the data into the memory. Just like training and testing the data, pre-process the data to meet the input requirements of our model and feed it to the network after pre-processing. The model will generate a prediction.

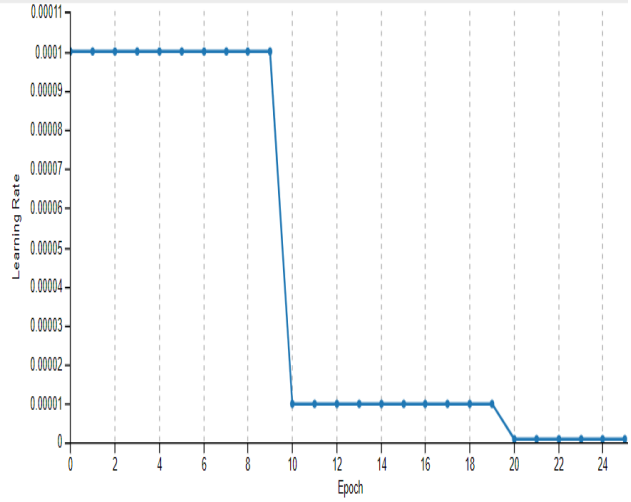


Figure 4. Performance Analysis of Learning Rate during Training

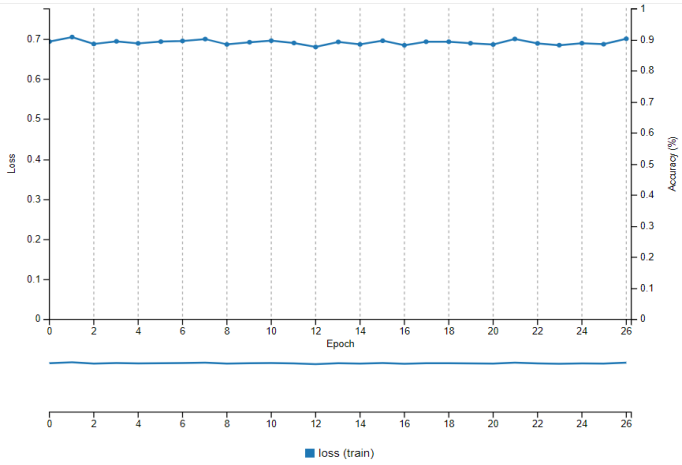


Figure 5. Performance Analysis of Loss rate during Training

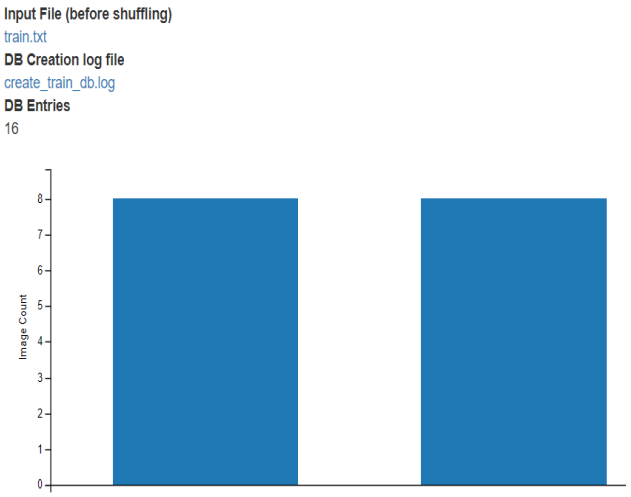


Figure 6. Creation of Image Count

Figure 7. Image Mean

Figure 8. Predicted Image of Louie

After that, a neural network model is trained by exposing the neural network to data. When exposed to data, the neural network changes to create an accurate map between inputs and outputs. An input of 256 x 256 color images to an output specifies the likelihood that each image belongs to a *class*, in this case case *dog* or *cat.* There is training of a neural network model to classify dogs from cats. After that, the trained model is deployed into a simulator. With image classification, the proposed network takes an *image* and generates *classification.* More specifically, the input in this case is a 256 x 256 x 3 tensor of pixel values and output is a 2-unit vector (because of 2 classes) of probabilities. A trained network consists of two components - A description of the architecture of the untrained network and the weights that were learnt while the network was trained. Deployment is the work of taking a trained model and putting it to work as a part of an application. For example, edge devices can be deployed such as robots or autonomous vehicles. There may also be the deployment to a server in order to play a role in any piece of software. Using this model, one can identify whether cats or dogs are existing in a particular image.

The goal is to avoid a cat from entering into the house through a doggy door. For the purpose of this simulation, a program is created that does one thing when the network

detects a dog and something different when the network detects a cat. Essentially, there is a need to combine deep learning with the traditional programming. An application could also have been created to see without deep learning but what deep learning can create to build an application has been used here. This step is called as d*eployment. The* trained neural network model can classify dogs and cats from DIGITS to create an application around it. This image classification model is shown in figure 9 that shows the predictions between cats and dogs. In the predication there is more probability of dogs.

The learning curve obtained from the training dataset provides a notion whether the model is learning properly. The learning curve on the basis of the hold-out validation dataset that provides an idea of how well the model generalizes is called Validation Learning Curve. The

Figure 9. Image Classification Model

Learning curve measured on the metric from which the various variables of the model are being optimized is called Optimization Learning Curve [43] for example - loss. The learning curve measured on the metric by which the model is accessed and chosen is called Performance Learning Curve, for example - accuracy. From the plot of loss as shown in figure 10, it might be observed that the proposed model has commensurate behaviour on training as well as validation datasets. If these similar plots begin to take-off harmoniously, it is an indication to terminate the training promptly. From the graph of accuracy, it may be seen that the model could possibly be upskilled somewhat more as the inclination for accuracy is increasing on both of the datasets since past few epochs. It can also be observed that the model has not learnt more so far in view of the training dataset that shows that the model is equivalently skilled on both the datasets.

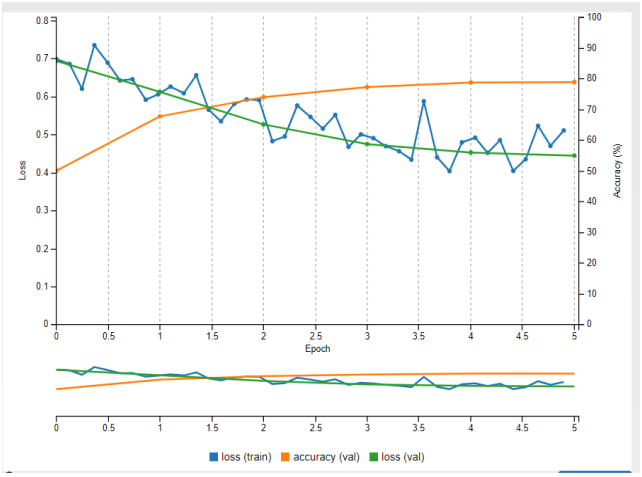


Figure 10. Performance Analysis of Loss with respect to Epoch for Training, Validation and Accuracy

6. Conclusion and Recommendations

Deep learning networks are emerging as a result of rising computing capacity on mobile hardware & the benefits of allowing excellent user experience. The behaviour and characteristics of deep-learning model training devices are largely non-transverse but realizing the performance characterization is crucial to the designing as well as execution of deep-learning models. The primary goal of this work is to investigate the performance of cutting-edge deep convolutional networks. In this work, a benchmark suite has been proposed for the training and deployment of Deep Neural Network. This encompasses a large variety of computer applications, from image recognition and machine translation to enhanced learning. DNN models have been implemented in three main configurations of deep learning: CNTK, MXNet and TensorFlow. Extensive research has been conducted using the various deep architectures such as AlexNet, InceptionNet, etc. To the best of authors' knowledge, this is the first work that presents a quantitative analysis of the deep architectures mentioned above. These models have been used to conduct the comprehensive behavioural survey and outlined to provide the effectiveness of Deep Neural Network training for the various configurations of hardware. The effectiveness of these methods has been evaluated using metrics such as learning rate, accuracy and epoch. The future scope of this research includes layer-by-layer fine-tuning of the ResNet50 architecture with larger datasets to improve the accuracy even further.

References

[1] Sindhwani, N.; Verma, S.; Bajaj, T.; and Anand, R.; "Comparative Analysis of Intelligent Driving and Safety Assistance Systems Using YOLO and SSD Model of Deep Learning." *International Journal of Information System Modeling and Design* 12, no. 1 (2021): 131-146.

[2] Kamalraj, R.; Neelakandan, S.; Kumar, M. R.; Rao, V. C. S.; Anand, R.; and Singh, H.; "Interpretable filter based convolutional neural network (IF-CNN) for glucose prediction and classification using PD-SS algorithm." *Measurement* 183: 109804.
[3] Ren, A.; Li, Z.; Ding, C.; Qiu, Q.; Wang, Y.; Li, J.; ... and Yuan, B.; "Sc-dcnn: Highly-scalable deep convolutional neural network using stochastic computing." *ACM SIGPLAN Notices* 52, no. 4 (2017): 405-418.
[4] Gupta, A.; Yadav, V.; "Hybrid Intelligence Model on the Second Generation Neural Network", *International Journal of Advanced Intelligence Paradigms*, Vol. 18, No. 3, pp. 398-416, February 2021.
[5] Rhu, M.; Gimelshein, N.; Clemons, J.; Zulfiqar, A. and Keckler, S. W.; "vDNN: Virtualized deep neural networks for scalable, memory-efficient neural network design." In *2016 49th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO),* pp. 1-13. IEEE, 2016.
[6] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; .... and Fei-Fei, L.; "Imagenet large scale visual recognition challenge." *International journal of computer vision* 115, no. 3 (2015): 211-252.
[7] Anand, R.; Singh, B.; and Sindhwani, N.; "Speech Perception & Analysis of Fluent Digits' Strings using Level-By-Level Time Alignment." *International Journal of Information Technology and Knowledge Management* 2, no. 1 (2009): 65-68.
[8] Juneja, S.; and Anand, R.; "Contrast Enhancement of an Image by DWT-SVD and DCT-SVD." In *Data Engineering and Intelligent Computing,* pp. 595-603. Springer, Singapore, 2018.
[9] Anand, R.; Shrivastava, G.; Gupta, S.; Peng, S. L.; and Sindhwani, N.; "Audio Watermarking With Reduced Number of Random Samples." In *Handbook of Research on Network Forensics and Analysis Techniques,* pp. 372-394. IGI Global, 2018.
[10] Umar, M.; Sabir, Z.; Raja, M. A. Z.; Baskonus, H. M.; Yao, S. W.; and Ilhan, E.; "A novel study of Morlet neural networks to solve the nonlinear HIV infection system of latently infected cells." *Results in Physics* 25 (2021): 104235..
[11] Parveen, H,; Yadav, V; "A Particle Swarm Optimization Strategy using QSAR modeling on the Second generation Neural Network", *IEEE International Conference on Advances in Computing, Communication Control & Networking (ICACCCN-2018)*, pp. 1188-1199, 2018.
[12] Umar, M.; Sabir, Z.; and Raja, M. A. Z.; "Intelligent computing for numerical treatment of nonlinear prey–predator models." *Applied Soft Computing* 80 (2019): 506-524.
[13] Sabir, Z.; Raja, M. A. Z.; Umar, M.; and Shoaib, M.; "Neuro-swarm intelligent computing to solve the second-order singular functional differential model." *The European Physical Journal Plus* 135 no. 6 (2020): 1-19.
[14] Schmidhuber, J.; "Deep learning in neural networks: An overview." *Neural networks 61*(2015): 85-117.
[15] Sabir, Z.; Guirao, J. L.; and Saeed, T.; "Solving a novel designed second order nonlinear Lane–Emden delay differential model using the heuristic techniques." *Applied Soft Computing* 102 (2021): 107105.
[16] Abdelkawy, M. A.; Sabir, Z.; Guirao, J. L.; and Saeed, T.; "Numerical investigations of a new singular second-order nonlinear coupled functional Lane–Emden model." *Open Physics*, 18, no. 1 (2020): 770-778.
[17] Sabir, Z.; Wahab, H. A.; Umar, M.; and Erdoğan, F.; "Stochastic numerical approach for solving second order nonlinear singular functional differential equation." *Applied Mathematics and Computation*, 363 (2019): 124605.
[18] Hordri, N. F.; Yuhaniz, S. S.; and Shamsuddin, S. M.; "Deep learning and its applications: a review." In *Conference on Postgraduate Annual Research on Informatics Seminar,* 2016.
[19] Shrestha, A.; and Mahmood, A.; "Review of deep learning algorithms and architectures." *IEEE Access* 7 (2019): 53040-53065.
[20] Dey, A.; "Machine learning algorithms: a review." *International Journal of Computer Science and Information Technologies* 7 no. 3 (2016): 1174-1179.
[21] Muhammad, I.; and Yan, Z.; "Supervised machine learning approaches: A survey." *ICTACT Journal on Soft Computing*, 5, no. 3 (2015): 946-952.
[22] Voulodimos, A.; Doulamis, N.; Doulamis, A.; and Protopapadakis, E.; "Deep learning for computer vision: A brief review." *Computational intelligence and neuroscience*, 2018. https://doi.org/10.1155/2018/7068349
[23] Ren, Y.; Yoo, S.; and Hoisie, A.; "Performance analysis of deep learning workloads on leading-edge systems." In *2019 IEEE/ACM Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS),* pp. 103-113. IEEE, 2019.
[24] Zhao, Z. Q.; Zheng, P.; Xu, S. T.; and Wu, X.; "Object detection with deep learning: A review." *IEEE transactions on neural networks and learning systems* 30, no. 11 (2019): 3212-3232.
[25] Liu, J.; Liu, J.; Du, W.; and Li, D.; "Performance analysis and characterization of training deep learning models on nvidia tx2." *arXiv preprint arXiv* 1906 (2019): 04278.
[26] Pouyanfar, S.; Sadiq, S.; Yan, Y.; Tian, H.; Tao, Y.; Reyes, M. P.; ... and Iyengar, S. S.; "A survey on deep learning: Algorithms, techniques, and applications." *ACM Computing Surveys (CSUR)* 51, no. 5 (2018): 1-36.
[27] Schwartzman, A.; Kagan, M.; Mackey, L.; Nachman, B.; and De Oliveira, L.; "Image Processing, Computer Vision, and Deep Learning: new approaches to the analysis and physics interpretation of LHC events." *Journal of Physics: Conference Series* 762, no. 1 (2016): 012035.
[28] Mouhssine,R.; Otman, A.; and Haimoudi, E.K; "Performance Analysis of Machine Learning Techniques for Smart Agriculture: Comparison of Supervised Classification Approaches." *International Journal of Advanced Computer Science and Applications,* 11, no. 3 (2020): 610-619.
[29] Kim, H.; Nam, H.; Jung, W.; and Lee, J.; "Performance analysis of CNN frameworks for GPUs." In *2017 IEEE*

International Symposium on Performance Analysis of Systems and Software (ISPASS), pp. 55-64. IEEE, 2017.
[30] Rahul, M,; Yadav, V,; "Zernike Moments based Facial Expression Recognition using Two staged Hidden Markov Model", *Advances in Computer Communication & Computational Sciences*, Vol. 924, pp. 661-670, 2019.
[31] Khan, A. I.; and Al-Habsi, S.; "Machine learning in computer vision." *Procedia Computer Science* 167 (2020): 1444-1451.
[32] O'Mahony, N.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Hernandez, G. V.; Krpalkova, L.; ... and Walsh, J.; "Deep learning vs. traditional computer vision." In *Science and Information Conference,* pp. 128-144. Springer, Cham, 2019.
[33] Suhail, A.; Jayabalan, M.; and Thiruchelvam, V.; "Convolutional neural network based object detection: A review" *Journal of Critical Reviews* 7*,* no. 11 (2020): 786-792.
[34] Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; and Alsaadi, F. E.; "A survey of deep neural network architectures and their applications." *Neurocomputing* 234 (2017): 11-26.
[35] Sze, V.; Chen, Y. H.; Yang, T. J.; and Emer, J. S.; "Efficient processing of deep neural networks: A tutorial and survey." *Proceedings of the IEEE* 105, no. 12 (2017): 2295-2329.
[36] Bengio, Y.; "*Learning deep architectures for AI*." Now Publishers Inc.(2009).
[37] Bakshi, G.; Shukla, R.; Yadav, V.; Dahiya, A.; Anand, R.; Sindhwani, N.; and Singh, H.; "An optimized approach for feature extraction in multi-relational statistical learning." *Journal of Scientific & Industrial Research* 80 (2021): 537-542.
[38] Wang, Z.; Liu, K.; Li, J.; Zhu, Y.; and Zhang, Y.; "Various frameworks and libraries of machine learning and deep learning: A survey." *Archives of computational methods in engineering*, (2019): 1-24. https://doi.org/10.1007/s11831-018-09312-w
[39] Sharma, D.; and Kumar, N.; "A review on machine learning algorithms, tasks and applications." *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)* 6, no. 10 (2017)): 2278-1323.
[40] LeCun, Y.; Bengio, Y.; and Hinton, G.; "Deep learning." N*ature* 521 (2015): 436-444.
[41] Abu, M. A.; Indra, N. H.; Abd Rahman, A. H.; Sapiee, N. A.; and Ahmad, I.; "A study on Image Classification based on Deep Learning and Tensorflow." *International Journal of Engineering Research and Technology* 12, no. 4 (2019): 563-569.
[42] Zhu, S.C.; and Mumford, D.; "A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision* 2, no.4 (2006): 259-362.
[43] Sindhwani, N.; "Performance Analysis of Optimal Scheduling Based Firefly algorithm in MIMO system." O*ptimization* 2, no. 12 (2017): 19-26.