Apriori Analysis of Deep Learning Models on a Multi-class Data Set

Sai Sathwik Kosuru⋆[1,2][0009−0008−8057−8799],
Padmanabha Reddy Y C A⋆⋆[1,3][0000−0002−9697−7072], Perala Venkata Akanksha[1,4], Kotagiri Chaitanya[1,5], Snehaja Parsi[1,6], and Madhu Babu Chunduri[1,7]

[1] Department of Computer Science and Engineering, B V Raju Institute of Technology, Narsapur, Medak, Telangana, India-502313
[2] 21211a05r6@bvrit.ac.in
[3] ananthayc.reddy@gmail.com
[4] akankshaperala2003@gmail.com
[5] chaithanyakotagiri06@gmail.com
[6] snehajaparsi@gmail.com
[7] madhubabu6585@gmail.com

Abstract. Deep Learning techniques have large scopes in terms of applications and resource requirements, hence calling for a deep analysis of their behavior. Many studies focused on the application part of these models and there is a relatively lesser discussion about the unit-wise behavior. A multi-class data set was taken and three different models were trained where their performances of training processes are described in terms of runtime characteristics like learning rate, times, etc. An apriori analysis was conducted where the runtime characteristics are investigated for their similarity with pre-defined knowledge of their architectures, as opposed to the posterior analysis which particularly focuses on observations. The work also serves as a framework for defining the logistics of the model training processes and fine-tunes the parameters devised in the future for the maximum accuracy of the model. The results were further converted into graphical representations for enhanced delivery of insights. The study is also novel where, as opposed to previous works, in which performances are only tested, the model is evaluated in a design-specific manner dealing with only architectures and the parameters defining them.

Keywords: Deep Learning Models · Apriori Analysis · Neural Networks · Hyperparameter tuning · Hardware and Software

1 Introduction

Deep Learning models are the convolutions of Feature analysis and Machine Intelligence Augmentations. While Machine Learning models perform similar

⋆ Primary Author
⋆⋆ Corresponding Author

tasks, that tasks lack Feature analysis feature and the extent of knowledge is less so far. Layer-based approach is implemented in the case of deep learning thereby mimicing the staged learning process of humans. The neural networks are built on the basis of human brain simulations differing from it's architecture only in biological aspects. Alternatively, these networks can be termed as virtual brains but yet significantly differ from the biological organ in terms of both structural and dynamic aspects. These models are analyzed just in terms of their performance in all the existing works, thus underscoring the importance of efficiency and precision in their working. Layers classify these models as- single-layer or multi-layer. Each layer is comprised of filters, collection of neurons, and interconnections. It is to be noted here that latent representations are made through each iteration and these representations are converged/ merged to get a perfect output. While these latent representations are created, it is important to observe that the features of the input image is stored and analyzed on an iteration basis. It should even be observed that the creation of a complete picture by the model, even with clarity and perfection, may not satisfy the desired use cases of the model. This is technically termed as **Overfitting**. In other words, particular instances cannot define the performance of the model, thus underscoring the need for generalizing in all aspects.

Conventional Neural Networks are considered as a part of this work which is critical to understanding the fundamentals of the parameter requirements for computer intelligence. Introspection of fundamental aspects may help in design of efficient models subject to effective learning. Current work stresses on run-time conditions of models' functioning and emphasize on parameters affecting models' efficiency. Hard-core solutions can be framed through the proposed findings and diverse range of new architectures can be built on the chassis developed here. The solution also serves as a basis for new models in the future in terms of enhanced efficiencies and performances. Existing systems least focused on the general aspects of these models. It should be admitted that the performance is not the same everywhere.Still, it is not risky to have a general intuition about these models' functions and individual effects. Therefore, the current work deals with a particular case of performance in a general manner which is presented in a graphical representation in this paper.

2 Literature Survey

2.1 Applications

Deep Learning Algorithms are often used in the medical diagnoses[4][8][9][11], CCTV footage analysis, Visual forensic sciences, etc. They deal with images and videos(videos may be considered as a collection of a large number of images called frames which are rolled continuously and steadily throughout the duration. Hence, the term videos may not be explicitly considered.). We may also see sparse applications in the scope of machine learning where they deal with mathematical computations and data tables. These models are adapted to multiple scales of images where radiographic images are even analyzed in case of MRI scan analysis in Medical applications. Vehicle counting and tracking is another sound system relevant to Transportation engineering. Agricultural[14] issues like plant disease detection, flower detection, sociological issues like anarchy detection, etc., are already being implemented across the globe with increasing efficiencies along the years. The YOLO models made the field of object detection highly efficient and fast, owing to state of the art frameworks proposed for many years. Surprisingly, fixed architecture have been defied and the novel concepts of dynamic architectures[6][12], suitable for the current requirements, can be sought out by advanced algorithms like Neural Architecture Search, which is the same that is integrated with YOLO model, resulting in the YOLO NAS model, the latest YOLO model that is able to provide high accuracy as compared to other versions of YOLO. Object detection in turn facilitates the use of mathematical models and structures to segment distinct objects in images, annotating them with bounding boxes. Besides object detection algorithms, relevance of deep learning pre-trained models like GoogleNet, AlexNet, etc. While applications just stress on accuracy[3], there is a need for broad study of internal studies thus underscoring the importance of current work.

2.2 Existing Works

Deep learning models are perceived on their performance in many of the existing systems where their performances are seen increased considerably over the previous ones. While various models are used in a single application to assess the suitable one, there is no general intuition as to what the apriori[5] factors state about them. Deep Learning technology is said to constitute the Fourth Industrial Revolution, thus forming an integral part of the **Technical Sustainability**. Deep Learning is absolutely higher in efficient on comparison to standard machine learning models. This is true in the cases where we consider advanced machine learning models and even overpowers the best ones. Surveys were also conducted in this context to make models smaller, faster, and efficient[2][10] thus highlighting the importance of the functioning of these technologies. These surveys are very much relevant to and played a critical role in the current work as it focuses on the same in a different and innovative way. Call for Efficient Deep Learning[7] has been increasing steadily in the recent days with outcome- and accuracy- based approaches. Colloquially, it is imminent that accuracy are solely considered while evaluating models in all the existing works. As a result, existing systems work on improving accuracy rather than exploring the fundamental backend structures[1], thus making the process unnecessarily complex and lengthy.

3 The Apriori Analysis

3.1 Dataset used for the study

The apriori analyses require multi-conditional considerations which calls for multiple constraints in the chosen problem. Data and its collection is a critical starting stage for deep learning models and hence, every deep learning model has its roots in the data collected. In this work, we have surveyed on multi-class data sets with minimum allowable number greater than six. There is a high-dimensional contrast amongst diverse classes in many data sets which makes each class significantly differentiable. Therefore, every model will have a sound performance on such data sets hence, defeating the original purpose of the current research. While there has been a greater number of innovations and researches in the direction of continuous curation of all relevant data in various domains, the problem of consistency in terms of stored features and similarities among various classes is gradually decreasing and the boundaries are rapidly subsidising. While there has relatively been a lull in simple data sets and fundamental models, such problems seldom find applications, though sparsely. Thus far, data sets are multi-class and multi-labelled, thus widening the scope of the work beyond certain boundaries.

Figure 1 illustrates all the dataset-specific details in detail and the description is as follows. Human and plant diseases, food images, Remote sensing image classifications, etc., often contain coarse features, some of which are impossible to discriminate based on visual data. This calls for extremely rugged images for generalizability and increased scalability. Hence, Plant disease detection data sets, having ten different class have been considered for the current work and the same were analyzed on a model-comparison basis where around three models are considered for evaluation-based examination. Several components of the collected data set have some similarities which cannot be differentiated purely based on visual information collected. Similarly, The Misclassification in images and semantics cannot be resolved in totality with just a single model. Hence, multiple models are considered for performance-based elucidations and the results are scrutinized for either making an analysis or perform testing of models.

3.2 Platform Preferences during analysis

Conventional systems, particularly household PCs, have less computational power where it is not possible to train every model. Moreover, Hardware inconsistencies may dominate the apriori analysis performed here. It is critical to be observed that the apriori analysis involves no hardware considerations and solely rely on functional implementations. Thus, the involvement of a study on platform dependencies may defeat the original purpose of the current research while underscoring the importance of pure-functional analyses in model comparisons. It is worth to be noted that an apriori analysis of deep learning models must include the concept of Hardware considerations since there are non-zero dependencies between function-specific software of models and hardware constraints. Though exploring the web of dependencies involved intricately in models' functions is not a primary focus, it has been necessitated to ensure proper channel of relations between them for efficient analyses, as evident in the current work. In this context, it is necessary to ensure a systematic evaluation of performance metrics across a wide range of platforms thereby calling for the running of same model on different platforms. Since this is an apriori analysis, platform parameters are never considered but model parameters varying as a function of platform-dependent parameters may be tested for a generalized conclusion.

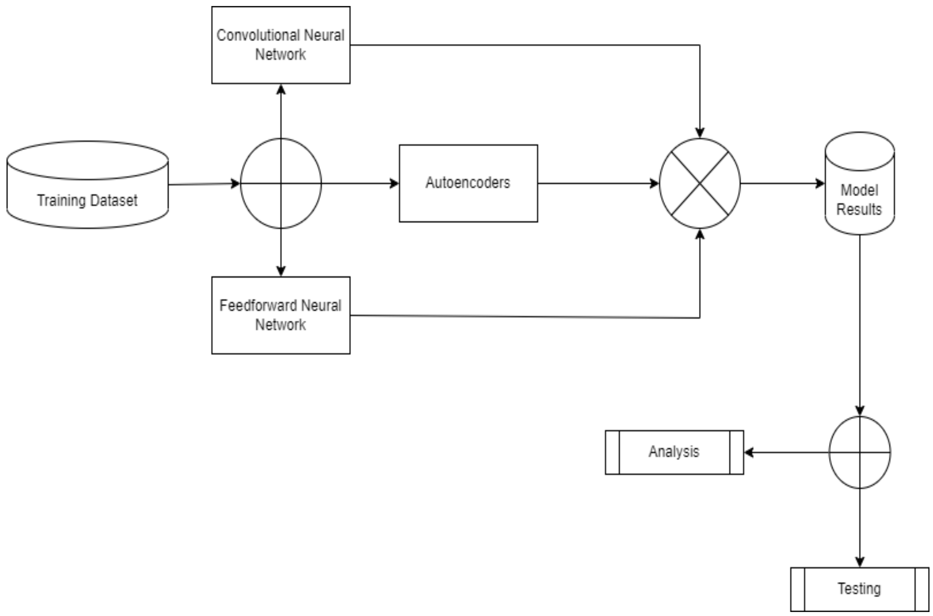


Fig. 1: Overview of the Contextual Dataset-Model interactions

Google Colab Google Colaboratory offers free GPU and deep learning-specific virtual infrastructure with a Jupyter-like interface. This service is available as a web-based IDE but with some limitations like not being able to access the physical resources of the computer. Intuitively, some or all of computer vision and related deep learning applications include the use of webcam, speakers, etc., which are generally not supported directly in this IDE. Additionally, it requires no explicit configurations and has many libraries or packages imported and ready to use at hand. Generally, Deep Learning frameworks and heavy algorithms are run on this IDE thereby making it increasingly usable for research purposes. Hence, all the models' trainings' performed as a part of the current work are run in this ide as a primary option or a primary choice.

Jupyter Jupyter is also a web-based ide but it runs on a kernel which is initiated by an Anaconda prompt. It is not directly web-based but utilizes hardware components of the system like Graphical Processing Units and needs installation of certain libraries/packages. Moreover, running deep leaning models on this ide is often tedious as it levies heavy burden on the systems resources where tight couplings can often cause failures in training processes and jamming of systems. It is even time consuming to run such models because system resources are often disbursed for various other processes and background tasks, hence significantly draining the efficacy of models' training processes when considering the time spent. As mentioned above, to accommodate use of system physical resources flexibly, this has been considered.

Kaggle Kaggle offers ide but to a limited extent. The amount of gpu offered too is limited but as it is data-based application, the ide offers faster training times and low failure rates or ratios. Hence, it has been consulted for few models even though was not initially preferred.

3.3 Analysis and working of Deep Learning Models

32	32	32	32	32	32
32	64	64	64	64	64
32	64	64	128	128	32
32	64	128	64	64	32
32	64	64	128	64	32

Table 1: Considered Hyperparameter Sets: Filter Settings

The following models are sought out to find probable relations as demonstrated in Figure 2.

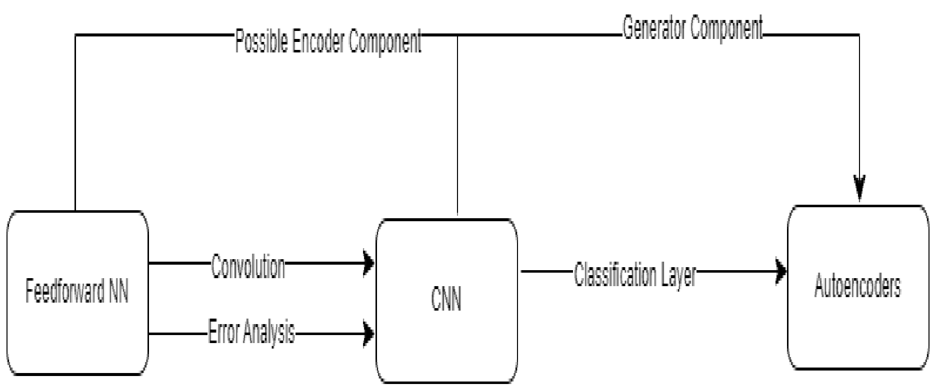


Fig. 2: Soughted Model Relations

Autoencoders Autoencoders are generally used for Generative AI applications and are broadly supervised. They may seem irrelevant in the current context but with appropriate modifications, they may be included. The layered structure supports equal neurons in both input and output layers with hidden layers performing critical functions. Latent representations are initially synthesized reproducing the features thereafter. In this reconstruction phase, errors are examined, accuracy is calculated and finally training is performed. It is worth to be noted that these iterations do not occur in minutes but take several hours for a good accuracy. There are two inbuilt models one of which is used to generate the latent forms using training algorithms and then computing errors to rectify them iteration-wise through the use of discriminator model. While these autoencoders often focus on reconstructions and latent representations, there has been an increasing need for classification-orientation of such models for model-comparison purposes. The only difference from the original version would be that a classifying layer is added in the initial parts of the model architecture, thus modifying it into a classification-based model. Alternatively the model can be used for reconstructing images for enhancing the data set content in addition to data augmentation and synthetic data techniques.

Feedforward Neural Network This is an elementary model which has three elementary layers: In Data Stream, A single Hidden Layer, and an Out Data Stream layer. There is no role of backtracking or backpropogation in this model and the flow of data is simply unidirectional. With appropriate modifications and optional addition of layers, we can continue improving the accuracy on a trial-and-error basis. Upon selecting appropriate hyperparameters, the model can be redefined to produce improved accuracy free of any overfitting. The concept of overfitting is highly dominant here since the number of layers is less and feature extraction may be performed at a higher pace. This leads to increased chances of exploding gradients thereby reducing efficiency gradually.

Convolutional Neural Network This algorithmic implementation is the most common and is treated as the most elementary deep learning neural network. Convolutions are typically found in biological organs like Brain, blood capillaries, and Kidneys where networks of fluid interchanges are mostly found. The same concept is applied on computers where the data/information takes the place of fluids and convolutions provide a more flexible approach for flexible and generalized applicability of the learnt model. Since CNN is found as the most common and preliminary model, the same was investigated for relevance rather than model comparison evaluations in the current work. We used Equation 1 to numerically extract features from different classes on a priority basis. Since the images are similar, we used correlation equations among pixels using three models for efficient performance.

$$G[i,j] = \sum_{n=-k}^{k} \sum_{n=-k}^{k} H[u,v]F[i+u,j+v] \tag{1}$$

H and F are filters and pixels respectively in 1

4 Results and Discussions

Thus far, there is no fixed benchmark for evaluating the sole responses and individual behaviours of these models. In the current paper, there is a focus on analyzing the models on an ideal basis with just time and performance applications. Model-to-model comparisons are made throughout the work but each model was initialized independent of the another. The results obtained are as shown in Figure 3 and the parameters of training are selected from 1.

4.1 Timestamp Analysis

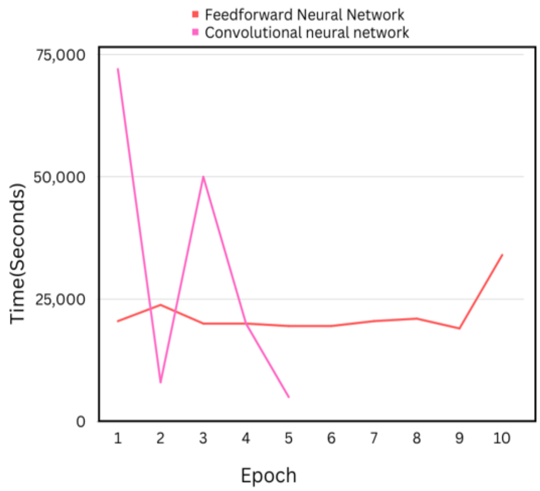
Feedforward Neural Network The fastest of the considered models is the Feedforward Neural Network with 48 hours for training on a data set of 1,21,000 images, divided into 10 classes. Hence, multi-class training is not taking a significant amount of time in this model when compared to the others. In this model,There has been a huge emphasis on learning rather than feature extraction as seen in Equation 2.

$$net = \sum w_i x_i + b_i \tag{2}$$

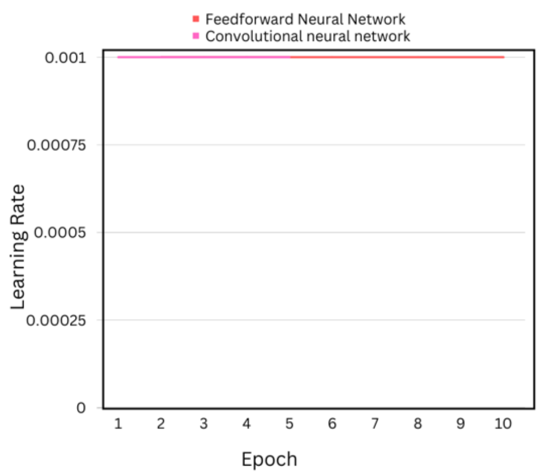
w_i stands for weights and, b_i stands for bias with x_i as input in Equation 2. Hence, it has taken less time where it just learns a latent version of the original image without delving deep into its features. This is evident when corn leaves are mistaken for pepper leaves owing to poor feature extraction.

Convolutional Neural Network Hyperparameter tuning played an inflecting part in the case of this model's training implementation where 8 sets of parameters are considered for timing analysis. CNN took 2 days for training where only 5 class are considered. For a large number of classes, the model is failing to execute completely wherein it is overfitting easily. The model is not complex in terms of timing considerations but is greatly complex in terms of hyperparameters where hyperparameters are relevant to the concepts of accuracy and had no role to play in time considerations.

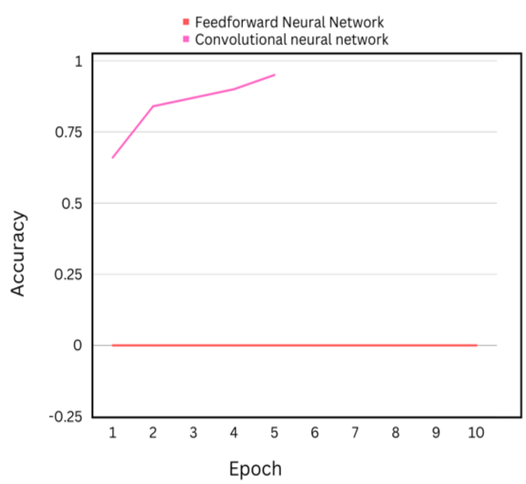
Autoencoders Autoencoders are quite complex where conventional IDEs are not supporting longer epochs and larger batches. It is worth noting that this model took multiple attempts for a successful model with repeated failures at approaching results. Autoencoders took around 3 days for initial training where the last epochs had a fatal failure when 10 diverse classes are considered. Hence, 5 classes are only considered where the epochs ran successfully for twelve hours relatively faster than the other models but the space is too high to accommodate the process of model saving. Hence, testing data is not viable even though time is taken at par with other even if partial training is performed.



(a) Timing Plots



(b) Learning Rate Plots



(c) Accuracy Demonstration(Apriori)

Fig. 3: Model Behavioral Plots

4.2 Accuracy Computation and Learning Rate determination

Accuracy is variable among the taken models where some models are overfitting too early and very easily. Surprisingly, the learning rate is too fast for dropout methods and regularization techniques to not be effective.

Feedforward Neural Network Epoch wise analysis gives a relatively faster learning rate where the accuracy curve has a very steep slope. Moreover, explicit definition of Learning rate seems unnecessary where the accuracy values are sharply increasing with batches. Hyperparameters are not even found to be efficient as the accuracy value begins at 0.9 for almost every set of hyperparameters. First and second epochs are relatively stable in terms of learning rates and the accuracy values are fixed at 0.9 consistently with one-hundredth gradient observed through all the batches, thus indicating a gradual start of accuracy values. While the third epoch starts with overfitting of the model, the model training still progresses even dropout and regularization parameters. Upon testing results, the model is being trained on pixelated values rather than features in this network where the model is not able to differentiate similar images. It is quite intuitive that feedforward neural networks has no backpropogation component and has few layers, hence the neurons are relatively less and they tend to receive all the information at an instance thereby crippling the systematic procedure that may occur in opposite conditions.

Convolutional Neural Network This is an elementary network which supports flexible number of layers, as per user definition.The Hyperparameter tuning is critical to improving the accuracy and determining the learning rate of the model. We proceed with defining the numerical amount of layer-wise filters, followed by the consideration of the layers' count. Multiple iterations, aimed at stabilizing accuracy rates from overfitting and improving learning rate were implemented but with marginal difference of results. As opposed to other networks, this network has a moderate learning rate and a gradually increasing accuracy slope. The accuracy, though it starts at 0.823 and progresses slowly in initial epochs, is on a steep rise in the later epochs with accuracy near to 1 in the last epochs. It is interesting to observe that the CNN is not overfitting easily even with increased number of filters. In extreme cases, number of filters used was nearly 8000 but still the model did not overfit. The only problem with this network is that it is overfitting with less data, thus underscoring the potential efficiency of the model.

Autoencoders Autoencoders found approximately stable accuracy(Equation 3) and learning rate variations while consuming a lot of resources for training. Autoencoders training involved a lot of integration issues and configuration problems where layers are not defining the accuracy of the model. Irrespective of the number of layers and defined hyperparameters, Autoencoders are found to be stable and has a very less learning rate. Since autoencoders are just investigated for generating new data sets, in addition to data augmentation and synthetic data techniques, they were not considered for accuracy and learning rate in the current work.

$$L_R = L(x, g(f(x)) + R(h) \tag{3}$$

Eq 3 is a relation in learning rates where x,g,h are dimensions.

4.3 Discussion of Results

The current apriori analysis results can be summarized as in Table 2.

Parameter	**Convolutional NN**	**Feedforward NN**	**Autoencoder**
Timing	48 Hours	18-44 Hours	72 Hours
Accuracy	0.33-0.98	0.9-1	0.53
Behaviour	Steady	Steep	Constant

Table 2: : Apriori Analysis Results

From the results, it is important to observe that timing, accuracy, and learning rate examinations may be considered worthwhile in case of simple models with latest models still requiring a deeper level of understanding. In ideal conditions, these parameters may be taken as sole determiners of model performances, but it is important to note the discrepancies in case of real life conditions where platform inconsistencies and system problems may arise. From the results, it can be stated clearly that model-to-model comparisons can be performed solely based on accuracy computations while their complexities may be examined using learning rates and timing calculations. The final equation of dependency may be framed as Equation 4.

$$f_{tr}(x) = O(T_{tr}(x)) + h(y_{hw}) * a \tag{4}$$

$$T_{tr}(x) \propto h(y_{hw}) \tag{5}$$

$\mathrm{f_{tr}}$(x) is a function that determines the functional dependencies of all the parameters considered in this analysis where $\mathrm{T_{tr}}$ is assumed to be timing calculations, h($\mathrm{y_{hw}}$) is a platform function, and a is the calculated accuracy. This is just a theoretical consideration and mathematical values are never sought in previous works. Key observations include the unexpected non-proportionality between time and learning rate(from 5) and the linear dependency between the functions linking all the sought observations(from 4). Though accuracy is taken as a non-linear function on a whole, it's behaviour is generally observed to be sequential or a step function where it increases gradually with epochs. In time and learning rates' data, they are purely linear functions with significant variations in slopes.

5 Conclusion and Future Scope

Preliminary analyses concluded with certain assertions about the computational requirements and output performances of various conventional learning models wherein timestamp, accuracy computation, and Learning rate determination was performed. Adding to this, Platform dependencies and IDE constraints were also considered since deep learning models and computational resources are deeply interlinked during the training period. While there has been a wide difference among the behaviours exhibited by various models, It is important to note the broad results produced by various models on the same dataset. It is even worth noting that irrespective of hyperparameters, the models possess characteristic behaviours, thus highlighting the importance of architectural designs in future models. Hence, the current apriori analysis provides an exemplary overview into the behaviours of the models and model-to-model comparisons on the same dataset on behavioral grounds.

The current research is just an abstracted apriori analysis of models' functioning, which doesn't include the concept of inherent structures and constituent neurons. With latest models and networks being introduced at peak frequencies into the market, it is important to observe the evolutionary behaviours of models and examining their performances. Since, there is no fixed benchmark to determine the performance of these models, an approximate mathematical notation may be devised to employ such measures. Hence, the future work emphasizes on a flexible framework that can adapt to any kind of models in the future, thus indicating a need for theoretical findings in the direction of the same.

References

1. Shrestha, A., Mahmood, A. (2019). Review of Deep Learning Algorithms and Architectures. IEEE Access, 7, 53040–53065. https://doi.org/10.1109/access.2019.2912200
2. Kiran, T. T. J. (2020, July). COMPUTER VISION ACCURACY ANALYSIS WITH DEEP LEARNING MODEL USING TENSORFLOW. International Journal of Innovative Research in Computer Science Technology, 8(4). https://doi.org/10.21276/ijircst.2020.8.4.13
3. Goswami, S., Singh, S. K. (2020, October). A simple deep learning based image illumination correction method for paintings. Pattern Recognition Letters, 138, 392–396. https://doi.org/10.1016/j.patrec.2020.08.013
4. Zhang, J., Xie, Y., Wu, Q., Xia, Y. (2019, May). Medical image classification using synergic deep learning. Medical Image Analysis, 54, 10–19. https://doi.org/10.1016/j.media.2019.02.010

5. Jie, Z. (2019). Intelligence Data Mining Based on Improved Apriori Algorithm. Journal of Computers, 52–62. https://doi.org/10.17706/jcp.14.1.52-62
6. Gilman, G., Walls, R. J. (2021, November). Characterizing concurrency mechanisms for NVIDIA GPUs under deep learning workloads. Performance Evaluation, 151, 102234. https://doi.org/10.1016/j.peva.2021.102234
7. Borisov, V., Leemann, T., Sessler, K., Haug, J., Pawelczyk, M., Kasneci, G. (2022). Deep Neural Networks and Tabular Data: A Survey. IEEE Transactions on Neural Networks and Learning Systems, 1–21. https://doi.org/10.1109/tnnls.2022.3229161
8. Syed, L., Jabeen, S., Manimala, S., Elsayed, H. a. G. (2019). Data science algorithms and techniques for smart healthcare using IoT and big data analytics. In Studies in fuzziness and soft computing (pp. 211–241). https://doi.org/10.1007/978-3-030-03131-211
9. Agravat, R., Raval, M. S. (2018). Deep learning for automated brain tumor segmentation in MRI images. In Elsevier eBooks (pp. 183–201). https://doi.org/10.1016/b978-0-12-813087-2.00010-5
10. Ramchandran, A., Sangaiah, A. K. (2018). Unsupervised anomaly detection for high dimensional data—an exploratory analysis. In Elsevier eBooks (pp. 233–251). https://doi.org/10.1016/b978-0-12-813314-9.00011-6
11. Kermany, D., Goldbaum, M. H., Cai, W., Valentim, C. C. S., Liang, H., Baxter, S. L., McKeown, A., Yang, G., Wu, X., Yan, F., Dong, J., Prasadha, M. K., Pei, J., Ting, M. Y. L., Zhu, J., Li, C. W.,Hewett, S., Dong, J., Ziyar, I.,Zhang, K. (2018). Identifying medical diagnoses and treatable diseases by Image-Based Deep Learning. Cell, 172(5), 1122-1131.e9. https://doi.org/10.1016/j.cell.2018.02.010
12. O' Mahony, N., Campbell, S., Carvalho, A., Harapanahalli, S., Hernandez, G. V., Krpalkova, L., Riordan, D., Walsh, J. T. (2019). Deep Learning vs. Traditional Computer Vision. In Advances in intelligent systems and computing (pp. 128–144). https://doi.org/10.1007/978-3-030-17795-910
13. Razzak, M. I., Naz, S., Zaib, A. (2017). Deep learning for medical Image Processing: Overview, challenges and the future. In Lecture notes in computational vision and biomechanics (pp. 323–350). https://doi.org/10.1007/978-3-319-65981-712
14. Guex, N., Peitsch, M. C., Schwede, T. (2009). Automated comparative protein structure modeling with SWISS-MODEL and Swiss-PdbViewer: A historical perspective. Electrophoresis, 30(S1), S162–S173. https://doi.org/10.1002/elps.200900140
15. A. Atanassov and D. Pilev, "Pre-trained Deep Learning Models for Facial Emotions Recognition," 2020 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 2020, pp. 1-6, doi: 10.1109/ICAI50593.2020.9311334.