iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** I **Month of publication:** January 2022

DOI: https://doi.org/10.22214/ijraset.2022.39873

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue I Jan 2022- Available at www.ijraset.com

Review on Different Software Tools for Deep Learning

Anshuja Anand Meshram[1], Kunal Bhojraj Kohale[2], Ankita Tote[3], Aniket Dnyandeo Agham[4], Prof. Sudesh A. Bachwani[5]

[1, 2, 3, 4, 5]*Department of Computer Engineering*

[1, 2, 3, 4]*Government College of Engineering Yavatmal , Maharashtra, India ,*[5]*Assistant Professor , Government College Of Engineering Yavatmal,Maharashtra, India*

[1, 2, 3, 4]*Dr. Babasaheb Ambedkar Technological University, Lonere*

Abstract: ***Deep Learning Applications are being applied in various domains in recent years. Training a deep learning model is a very time consuming task. But, many open source frameworks are available to simplify this task. In this review paper we have discussed the features of some popular open source software tools available for deep learning along with their advantages and disadvantages. Software tools discussed in this paper are Tensorflow, Keras, Pytorch, Microsoft Cognitive Toolkit (CNTK).***

Keywords: Deep Learning, Frameworks, Open Source, Tensorflow, Pytorch, Keras, CNTK.

I. INTRODUCTION

In the past decade, deep learning has been successfully applied in diverse application domains including computer vision, image classification, speech recognition, natural language processing, etc. The success of deep learning is attributed to its high representational ability of input data, by using various layers of artificial neurons. In order to improve the efficiency in developing new deep neural networks, many open-source deep learning toolkits have been recently developed. All these tools support multi-core CPUs and many core GPUs for high-performance. One of the main tasks of deep learning is to learn a huge number of weights, which can be implemented by vector or matrix operations.[1]

A deep-learning framework is responsible for dealing with the high number of system and model parameters associated with the learning process of the network. These parameters are in the millions, making their management the primary focus of any optimization problem for any DL framework. With the advancement of a number of deep-learning frameworks, the choice of framework for any problem plays an important role in efficiency.[2] In this paper ,we have discussed various software tools used for deep learning. Software tools discussed in this paper are Keras, Pytorch , Tensorflow and Microsoft cognitive toolkit (CNTK) .

II. LITERATURE REVIEW

A. *Tensorflow*

1) *Introduction:* TensorFlow is a software library or framework, designed by the Google team to implement machine learning and deep learning concepts in the easiest manner. It integrates the computational algebra of optimization techniques for simple computation of many mathematical expressions. Data preprocessing, developing the model, and finally training and evaluating the model are the three aspects of the Tensorflow structure. The name tensorflow is derived from the fact that tenserflow takes input in the form of a multi-dimensional array, which are commonly known as tensors. You can create a flowchart of the operations that you want to run on that input. The input enters at one end, travels through this system of various processes, and emerges as output at the other.

2) *Features*

a) Tensorflow includes a feature of that defines, optimizes and calculates mathematical expressions easily with the help of multi-dimensional arrays called tensors.

b) Tensorflow includes a programming support of deep neural networks and machine learning techniques.

c) It includes a high scalable feature of computation with various data sets.

d) TensorFlow uses GPU computing, automating management. It also has a unique feature of optimization of same memory and the data used. It supports many networks such as CNNs and RNNs with different settings. TensorFlow is designed for remarkable flexibility, portability, and high efficiency of equipped hardware.

3) *The Architecture of Tensorflow Consists Following Components*

1) *Tensorflow Servable:* These are the central uncompleted units in TensorFlow serving. Servables are the objects that the clients use to perform the computation. The size of a servable is flexible. One servable may include anything from a lookup table to a unique model in a tuple of interface models. Servable should be of any type and interface, allowing for flexibility and future enhancements. The TensorFlow server can handle one or more servable versions, during the lifecycle of any single server instance. It opens the door for new a algorithm, weights, and other data can be loaded over time. They also can allow more than one version of a servable to be charged at a time. They also allow more than one servable versions of to be loaded simultaneously, supporting roll-out and experimentation gradually.
2) *Servable Streams:* Servable streams are sequence of versions of any servable sorted by increasing version of numbers.
3) *Tensorflow Loaders:* Loaders are responsible for managing servable's life cycle. The Loader API enables the standard infrastructure independent of the specific learning algorithm, data, or product usage conditions involved.
4) *Sources in Tensorflow:* sources are modules that find and provide servable Each reference provides an additional zero or useful stream at a time. For each servable stream, for every servable a source provides only one loader instance.
5) *Tensorflow Managers:* Tensorflow managers handle the full lifecycle of a Servables. They are responsible for loading, unloading and serving servables. Manager observes sources and tracks all versions. The Manager tries to fulfill causes, but it can refuse to load an Aspired version.

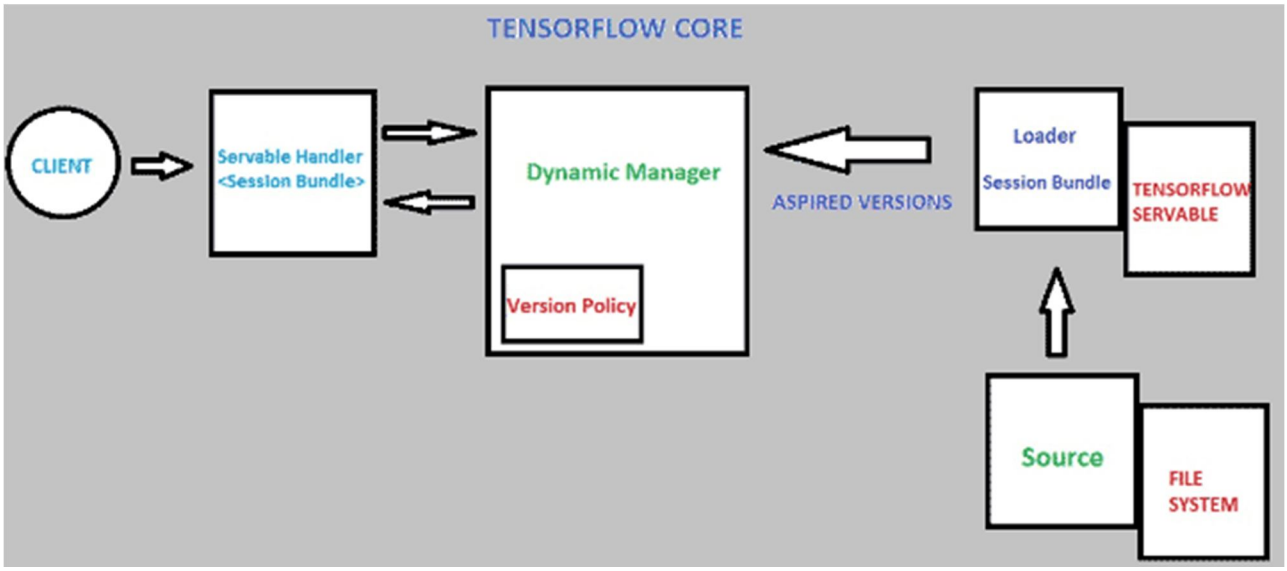


Fig. 1 Architecture of tensorflow

4) *Applications*

a) Image recognition
b) Voice recognition
c) Video detection
d) Text-based Application
e) Self driving cars
f) Sentiment analysis

B. *KERAS*

1) *Introduction:* Keras is an open source deep learning framework for python. It has been developed by Francois Chollet who is an artificial intelligence researcher at Google. Keras provides a complete framework to create any type of neural networks. Keras is innovative and is very easy to learn. It supports a simple neural network and also a very large and complex neural network model. Using Keras in deep learning allows for easy and fast prototyping as well as running seamlessly on CPU and GPU. This framework is written in Python code which makes it is easy to debug and allows ease for extensibility. It enables the user to complete the task in minimum lines of code. Keras is one of the most popular libraries in deep learning domain.

2) Features

a) Modularity: Keras is modular. It assumes a model in the graphical or a sequential form. You can save the model which you are working on in Keras. save() method in Keras can be used to save the current model. You can even use the model in the future.

b) Evaluation and Prediction: Keras includes evaluate() and predict() methods. Dataset of NumPy can be used by these methods. The evaluation of the result is done, after testing the data. These methods are used to evaluate our models.

c) Pre-trained Models: Keras contains a number of pre-trained models. These models can be imported from keras applications. These models can be used for feature extraction and fine-tuning.

- Consistent, simple and extensible API.
- Minimal structure – easy to achieve the result without any frills.
- It supports many platforms and backends.
- It is user friendly framework which runs on both CPU as well as GPU.
- Highly scalability of computation.

3) Keras Can be Devided into Three Main Categories

a) Keras Models: Keras model represents the actual neural network model. Keras models mainly are of two types which are Keras Sequential Model and Keras Functional API. Sequential model is the linear arrangement of Keras Layers. Sequential model is easy, minimal as well as has the ability to represent nearly all available neural networks. Functional API is used for creating complex models. Functional API lets us create models that have multiple input or output.

b) Keras Layers: Keras layers are the primary building block of Keras models. Each keras layer has input layer, hidden layer and output layer. Each layer receives input information, performes calculation and finally output the converted information. The output of one layer enters into the next layer as its input.

c) Keras Modules: Keras modules contains pre-defined classes, functions and variables which are useful for deep learning algorithm.

Every ANN is represented by Keras Models, in Keras. In fact, every Keras Model is a part of Keras Layers and represents ANN layers like input, hidden layer, output layers, convolution layer, pooling layer, etc., Keras model and keras layer access Keras modules for activation function, loss function, regularization function, etc., Any ANN algorithm (CNN, RNN, etc.,) can be represented in a simple and efficient manner by using Keras model, Keras Layer, and Keras modules,

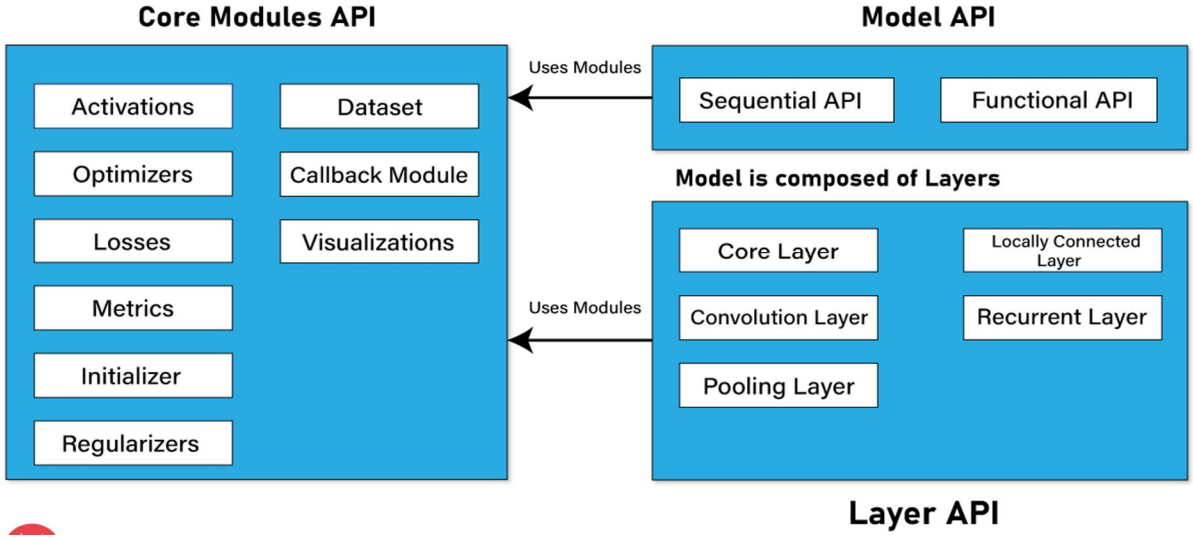


Fig 2. Architecture of keras

4) Applications

a) Keras is used for creating deep learning models which can be used on smartphones.

b) Keras is also used for distributed training of deep learning models.

c) Keras is used by companies such as Netflix, Yelp, Uber, etc.

C. PyTorch

1) Introduction: PyTorch is defined as an open source machine learning library for Python. It is used for applications such as natural language processing. Initially pytorch is developed by Facebook artificial-intelligence research group, and Uber's Pyro software for probabilistic programming which is built on it. Originally, it was developed as a Python wrapper for the LusJIT by Hugh Perkins based on Torch framework. There are two PyTorch variations. PyTorch redesigns and implements Torch framework in Python while sharing the same core C libraries for the backend code. The developers of pytorch modified this back-end code to run Python efficiently. They also kept the GPU based hardware acceleration as well as the extensibility features that made Lua-based Torch.

2) Features

a) Easy Usage: PyTorch offers easy to use API; hence it is considered to be very simple to operate and runs on Python. The execution of code is quite easy in this framework.

b) Python Interface: This library is considered to be Pythonic which smoothly integrates with the Python data science stack. Therefore, it can use all the services and functionality provided by Python environment..

c) Computational Graphs: PyTorch offers an excellent platform which providss dynamic computational graphs. Thus the changes in these graphs can be made by user during runtime. This is highly useful when a developer has does not know how much memory is needed for creating a neural network model.

d) Easy to Debug: It can use debugging tools like pdb and ipdb tools of python. PyTorch develops a computational graph at runtime so programmer can use Pythons IDE PyCharm for debugging.

e) Data Parallelism: It can distribute the computational tasks among multiple CPUs or GPU. This is possible because of the data parallelism (torch.nn.DataParallel) feature of pytorch, which wraps any module and helps us do parallel processing.

f) Useful Libraries: Pytorch has huge community of developers and researchers who built tools and libraries to extend it. These libraries helps in developing computer vision, reinforcement learning, NLP for research and production purposes. some of the popular libraries are GPyTorch, BoTorch, and Allen NLP. The rich set of powerful APIs helps to extend the PyTorch framework.

3) Components of PyTorch

Following are the major components of pytorch:

a) Tensors: Tensors are the Multi-Dimensional array similar to the Numpy array, and Tensors are available in Torch as a torch. IntTensor, torch.FloatTensor, torch.CharTen etc.

b) Variable: A variable is a wrapper around tensors to hold the gradient. The variable is available under torch.autograd engine as a torch.autograd.Variable.

c) Parameters: Parameters are basically a wrapper around the variable. It is used when we want tensor as a parameter of some module which is not possible using a variable as it is not a parameter or Tensors it does not have a gradient, so we can use parameters under the torch.nn as a torch.nn.Parameter.

d) Functions: Functions are responsible for performing the transform operations, and they do not have a memory to store any state. Like log functions will give output as log value, and linear layer can not function as it stores weight and biases value. Examples of functions are torch.log, torch.sum, etc., and functions are implemented under the torch.nn.functional.

e) Modules: Module used as under Torch as a torch.nn.Module is the base class for all neural networks. A module can contain other modules, parameters, and functions. It can store state and learnable weights. Module are types of transformation which can be implemented as torch.nn.Conv2d, torch.nn.Linear etc.

4) Applications

a) Computer Vision: It uses a convolution neural network to develop image classification, object detection, and generative application. Using PyTorch, a programmer can process images and videos to develop a highly accurate and precise computer vision model.

b) Natural Language Processing: The language translator, language modeling, and developing a chatbot can be done by natural language processing. It uses RNN, LSTM, etc. Architecture to develop natural language, processing models.

c) Reinforcement Learning: Developing Robotics for automation, Business strategy planning or robot motion control, etc can be done using reinforcement learning. Pytorch uses Deep Q learning architecture to build a model.

D. Microsoft Cognitive Toolkit (CNTK)

1) Introduction: Microsoft Cognitive Toolkit (CNTK), formerly known as Computational Network Toolkit, is a free, easy-to-use, open-source, commercial-grade toolkit that allows us to train deep learning algorithms to learn like the human brain. It allows us to create some popular deep learning systems such as time series prediction systems, feed forward neural network and Convolutional neural network (CNN) image classifiers. Its framework functions are written in C++, for optimal performance. Although we can call its function using C++, but the most commonly used approach for the same is to use a Python program.

2) Features

a) Built-in Components

CNTK has highly optimised built-in components that can handle multi-dimensional dense or sparse data from Python, C++ or BrainScript.
We can implement CNN, FNN, RNN, Batch Normalisation and Sequence-to-Sequence with attention using CNTK.
It provides us the feature to add new user-defined core-components on the GPU from Python.
It also offers automatic hyperparameter tuning.
Reinforcement learning, Generative Adversarial Networks (GANs), Supervised as well as Unsupervised learning can be implemented using CNTK.
CNTK contains built-in optimised readers, for massive datasets.

b) Usage of Resources Efficiently

Parallelism is provided by CNTK with high accuracy on numerous GPUs/machines via 1-bit SGD.
It provides memory sharing and other built-in methods, to fit the largest models in GPU memory.
For defining your own network, learners, readers, training and evaluation from Python, C++, and BrainScript, CNTK has full APIs.
We can easily evaluate models with Python, C++, C# or BrainScript, using CNTK.
It offers high-level and low-level APIs.
CNTK can automatically shape the inference, based on our data.
CNTK contains fully optimised symbolic Recurrent Neural Network (RNN) loops.

c) Measuring Model Performance: CNTK provides several components for measuring the performance of neural networks you built. It Generates log data from your model and the associated optimiser, which we can be used to monitor the training process.

3) General Architecture of CNTK

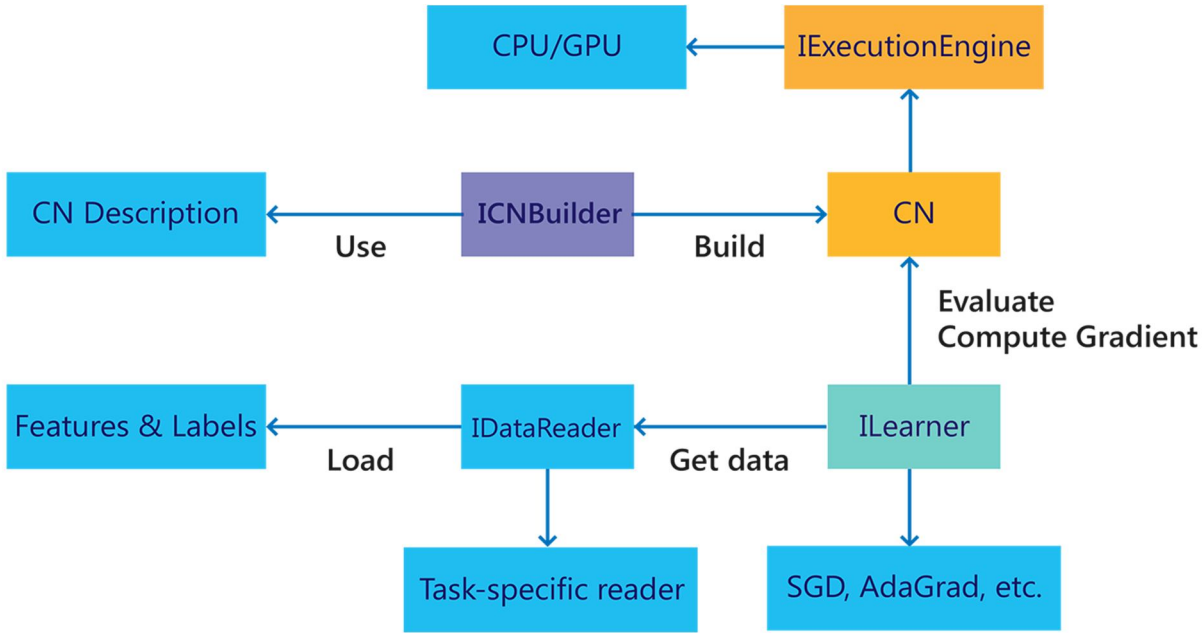


Fig. 3 Architecture of CNTK

Training a neural network with CNTK involves three components that must be configured:

a) *Network:* The neural network, including its structure/formula, model parameters. Training criteria and evaluation metric are also included here.
b) *Reader:* How training data is read
c) *SGD:* The hyper-parameters of the stochastic-gradient process

4) *Applications*

a) Speech, image, text & combination of data types deep learning applications.
b) CNTK is used for object recognition in the fragmented reality application.
c) Face detection application.
d) To train the models based on Feed Forward, Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) and Sequence-to-Sequence algorithms.

III. STUDY COMPARISON

Table I
Comparative Study of different software tools

Framework	Platform	Advantages	Disadvantages
Tensorflow	Linux, MacOS, Windows, Android	Tensorflow has a lot of documentation and guidelines. It offers monitoring for training processes of the models and visualization. It's backed by a large community of developers and tech companies. Tensorflow Lite empowers on-device inference with low latency for mobile devices.	TensorFlow is a bit slow compared to frameworks like CNTK. Debugging can be challenging in tensorflow.
Keras	Linux, MacOS, Windows	It's lightweight in terms of building DL models with a lot of layers, It features fully-configurable modules. It has a simplistic and intuitive interface suitable for beginners. It has built-in support for training on many GPUs. It supports NVIDIA GPUs, Google TPUs, and Open-CL-enabled GPUs such as AMD.	Keras can be too high-level and not always easy to customize. It is restricted to Tensorflow, CNTK, and Theano backends. It also doesn't provide as many functionalities as TensorFlow and ensures less control over the network.
Pytorch	Linux, MacOS, Windows, Android	It is user-friendly design and structure that makes constructing deep learning models transparent. It has useful debugging tools like PyCharm debugger. It contains several pre-trained models and supports distributed training.	It does not have interfaces for monitoring and visualization like TensorFlow. Comparatively, PyTorch is a very new deep learning framework and currently has less community support.
Microsoft Cognitive Toolkit (CNTK)	Windows, Linux, MacOS	It offers reliable and excellent performance. CNTK is a popular choice in many enterprises because of its scalability. It has numerous optimized components. It is easy to integrate with Apache Spark, which is an analytics engine for data processing. It works well with Azure Cloud, both being backed by Microsoft.	It has minimal community support compared to Tensorflow.

IV. CONCLUSION

In this review paper, we have explored top deep learning software tools . Selection of the tool depends on our requirement for the algorithm, our expertise level, and the price of the tool. Machine learning library should be easy to use. TensorFlow is very popular in machine learning, but it has a slight steep learning curve. PyTorch is also a popular tool for machine learning and support Python programming language. Keras.io and TensorFlow are good for neural networks. Microsoft cognitive toolkit provide exceptional scaling capabilities along with speed and accuracy and enterprise-level quality.

We estimate the running time performance of a set of modern deep learning software tools and observe how they perform on different types of neural networks and different hardware platforms. Our experimental results show that all tested tools can make good use of GPUs to achieve significant speedup over their CPU counterparts.

REFERENCES

[1] Shi S, Wang Q, Xu P, Chu X. Benchmarking state-of-the-art deep learning software tools. In2016 7th International Conference on Cloud Computing and Big Data (CCBD) 2016 Nov 16 (pp. 99-104). IEEE.

[2] Nara M, Mukesh BR, Padala P, Kinnal B. Performance evaluation of deep learning frameworks on computer vision problems. In2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) 2019 Apr 23 (pp. 670-674). IEEE.

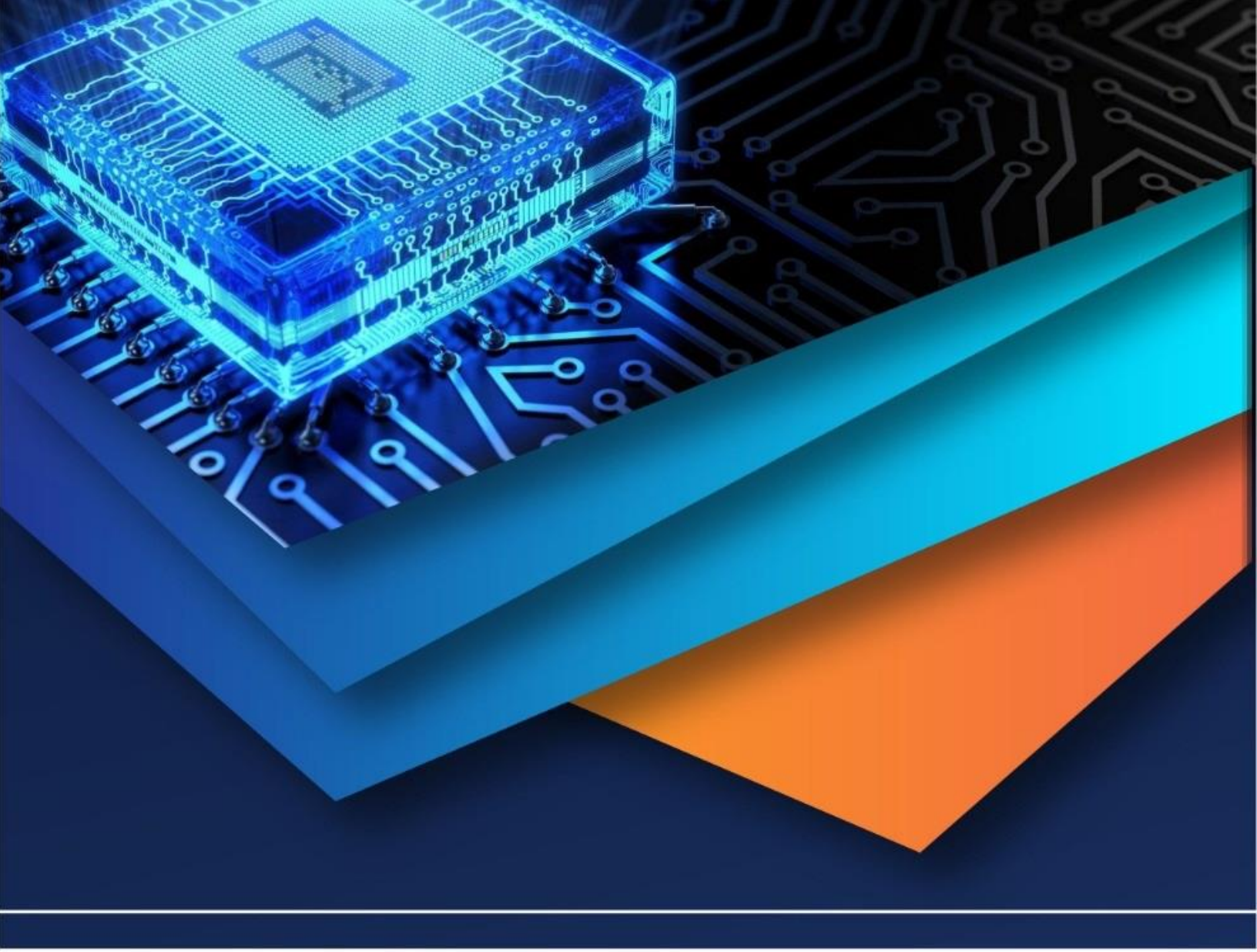
[3] Mihai Cristian Chirodea, Ovidiu Constantin Novac, Cornelia Mihaela Novac, Nicu Bizon, Mihai Oproescu, Cornelia Emilia Gordan. Comparison of Tensorflow and PyTorch in Convolutional Neural Network - based Applications. In2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI). IEEE.

[4] S. Bahrampour, N. Ramakrishnan, L. Schott, and M. Shah, “Comparative study of caffe, neon, theano, and torch for deep learning,” CoRR, vol. abs/1511.06435, 2015. IEEE.

[5] H. Kim, H. Nam, W. Jung, and J. Lee, “Performance analysis of CNN frameworks for GPUs,” in 2017 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), April 2017, pp. 55–64.

[6] Nguyen Quang-Hung, Hieu Doan, Nam Thoai, Performance Evaluation of Distributed Training in Tensorflow 2. In2020 International Conference on Advanced Computing and Applications (ACOMP). IEEE.

[7] Zejun Zhang,Yanming Yang,Xin Xia,Davi, Lo,Xiaoxue Ren,John Grundy. Unveiling the Mystery of API Evolution in Deep Learning Frameworks: A Case Study of Tensorflow 2. In2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP). IEEE



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY