International Journal of Religion
2024
Volume: 5| Number 11 | pp. 1203 – 1213
ISSN: 2633-352X (Print) | ISSN: 2633-3538 (Online)
ijor.co.uk
DOI: https://doi.org/10.61707/gkh1m822

Comprehensive Review and Comparative Analysis of Keras for Deep Learning Applications: A Survey on Face Detection Using Convolutional Neural Networks

Noor Abdalkareem Lafta[1],[2], Zainab Ali Abbood[3]

Abstract

This paper aims to provide a literature review and comparative analysis focused on the importance of using Python as the primary language for machine learning (ML), deep learning (DL): The incorporation of libraries adapted to different areas is highlighted. One of the most depressing things about using Python in the programming sector is its easy and extensive libraries. Of these few, Keras shines with the majority of its design decisions being made based on the core concepts. Keras allows a number of possibilities for model deployment in a production mode, it supports multiple GPUs properly, and it also supports distributed model training. The design of Keras is very simple to use and can be easily learned due to its usage of Python programming language, making it a good open-source tool to be used in building and testing deep learning (DL) models. This paper focuses on Keras, an open-source deep learning application program interface that runs on Python and is based on TensorFlow but compatible with others such as PyTorch, TensorFlow, CODEEPNEATM, and Pygame. The review goes deeper into the details regarding Keras, including its goals, issues it sought to address, achievements it has made and some lessons drawn from its use.

Keywords: *CNNs, DL, Face detection, Image Recognition, ML, Python, TensorFlow*

INTRODUCTION

The great advancement of machine learning and deep learning in the recent past has enabled the creation of multipurpose use in many sectors. This growth is due to the established processing power hence, availability and the big data and new applications that support the deep learning [1]. Currently, it is Python when it comes to designing artificial neural networks (ANNs), thereby leaving other programming languages that also offer this kind of feature [2]. It possesses various scientific libraries which can be used for various aims and objectives for instance, complex simulation, regression analysis, solution of ordinary differential equations among many others [3]. Both these libraries are both powerful and easy to use, and can be incorporated in other programs or projects in the way in which modules are used, which is a very high level of flexibility for the library, because it contains no specific contextual restraints inherent in the way as complexes as C++ or C libraries do.

Out of these libraries, Keras has been seen as a preferred library and is mostly used among the practitioners. Keras is a high-level neural networks API and it works with TensorFlow in the behind-the-scenes and provides a much simpler way to implement, train and test the models [4]. It makes the utilization of deep networks with TensorFlow easier due to the availability of only some complex features though it allows customizations and has a highly user-friendly nature [5]. Perhaps the most compelling aspect that differentiates Keras from the rest is the fact that it is highly recommended for deep learning applications [6].

This survey aims to existing a comprehensive review of the Keras library as a deep learning framework based on state-of-art contributions reported in diverse domains of research in the recent past three years, analyzed from different angles. This paper aims to reflect on all the important issues, such as objectives, methodologies, datasets, difficulties and major discoveries, as well as measures of accuracy. Furthermore, there is an emphasis on the most popular methods used within the Keras environment in the same case [7].

Face recognition has remained a favorite topic in the field of computer vision and systems. Face recognition is cool as it is a wide classification of subject where has a vast range of engineering and science subfields such as machine vision [2], biometrics and security [3], multimedia processing, psychology, neuroscience, etc. This

[1] Computer and Communication Engineering dept., Faculty of Engineering, Islamic University of Lebanon (IUL), E-mail: nn4304616@gmail.com

[2],[3]Computer Technology Engineering Dept, Al-Esraa University, Baghdad, Iraq. E-mail: zainababbood@esraa.edu.iq, (Corresponding Author)

rather extensive research area has constituted active work toward attaining more precise outcomes in the long run. Face recognition has been found to have certain advantages over the other biometric modalities as it is more acceptable; people are well conversant with, and easier to explain to the public [2]. As such, it has many uses such as criminal identification, access to locked mobile devices and laptops, home access, finding lost individuals, aid the blind, social media recognition, disease identification, and real-time monitoring including systems management.

KERAS

Keras is a Python deep learning API started as part of Tera-ML with the express purpose of sitting on top of low-level frameworks such as TensorFlow. One of its main objectives is to allow for a high degree of experimentation and quick prototyping the system, while mostly focusing on its usability, capability for modularity, and extensibility [8]. This enables Keras to have the following covered as its neural network components; Density layers, Convolutional layers, Recurrent layers, Dropout layers, Amongst these layers, the following is an illustration of how they can be combined; It also has simple procedures for various requirements like CPU and GPU resources, and ways of classifying training sessions. Furthermore, it provides several activations functions implementations, optimizers, and metrics implementations [9].

It has become clear that Keras provides native support to enable and utilize the tools of deep learning based on TensorFlow in the Python environment. This integration helps in easy deployment of sophisticated machine learning models and algorithms in their development, without having to write lines of low-level code. Moreover, Keras is equipped with many other ready models and utility for image and writings modes tasks that make development easier for researchers or developers. It being easy to use and flexible, has contributed to Keras being the most widely used tool in the deep learning field [5-3].

Refining the mentioned tasks, Keras is applicable to a broad range of problems, from classical image recognition and classification to the sophisticated Natural Language Processing, generative models, and reinforcement learning. This, coupled with Keras's self-organizing community and enormous documentation sources, speaks volumes to Keras's popularity and continual improvement. Keras can give a proper setting for carrying out studies associated with deep learning and real-life applications because of its versatility [7].

Keras Library with Deep Learning (DL)

Keras is seen as a deep learning approach that is specifically intended to assist in model creation and training. It is highly modular in architecture which can be compared with Lego; they are simply put together and the structure is quite easy to understand. The authors found that the implementation of these models can be undertaken with few parameters, fewer training epochs and most basic monitoring procedures as shown in 10. The ease of using deep learning models with Keras can well require the line of codes reduced by a considerably large margin. This can turn into a terrific opportunity for application developers to improve the efficiency of their operations and direct their efforts to the most pertinent issues like creating complex deep learning algorithms.

Sequential Model API in Keras is useful when implementing the typical models with less necessary coding. In fact, all seems quite simple with Keras which is the main reason why its API and together with Model and the Layer classes are easily modifiable in order to create more complex models. The Functional API is used for construction of models which work just like any Python function and also for graph-like designs plus layer stacking [6]. It provides a flexible way to use other experimental or innovative deep learning frameworks and layers learned with the Concept and Layer part of this program.

Keras is another great leap forward because it allows users to define and build models of neural networks spontaneously spare code. This can be quite advantageous to researchers as well as developers due to the fact that it enables them to focus on the technology aspect in addition to the development part without having to bother with complex coding. Further, Keras is well-documented and has a strong user base which is instrumental when it comes to fostering actual use within the deep learning community especially for new users.

The procedures for constructing the deep learning model using Keras are as follows: Figure 1 presents a simple and easy-to-follow manner to illustrate the process.

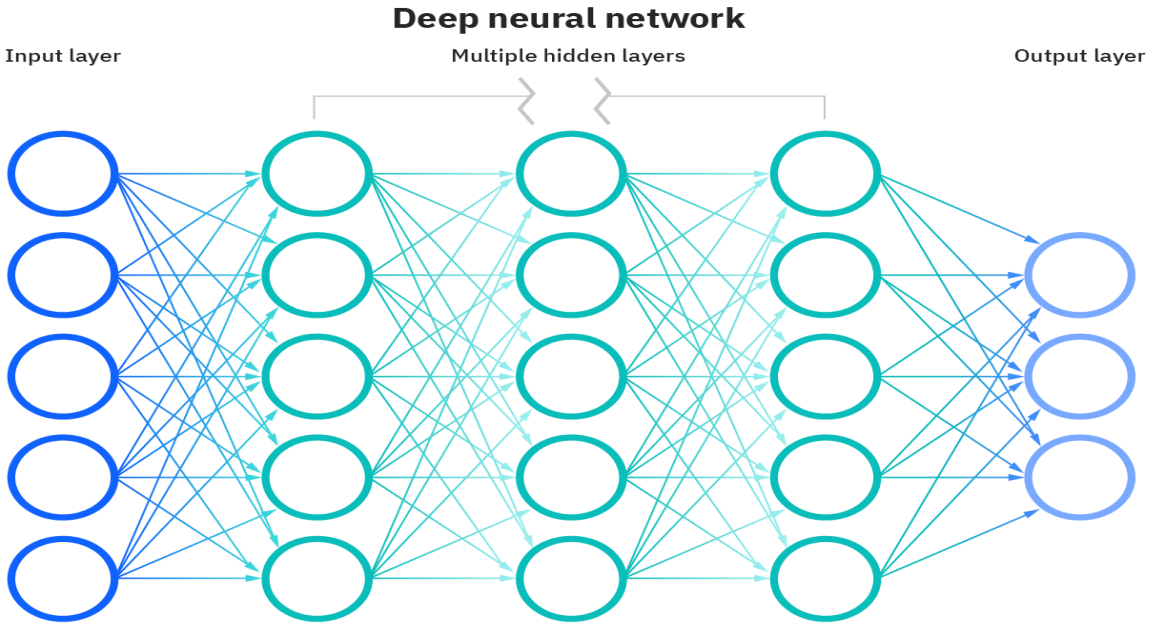

Figure 1. Deep Learning in Keras-Creating a DL Model [11].

To clarify, Keras is not an independent DL library. You may see it in action in Figure 2 atop another deep learning backend or library. Theano from MILA, TensorFlow from Google, or CNTK from Microsoft could be involved. Kernas can be run by a central processing unit (CPU), graphics processing unit (GPU), as well as Google's TPU [12].

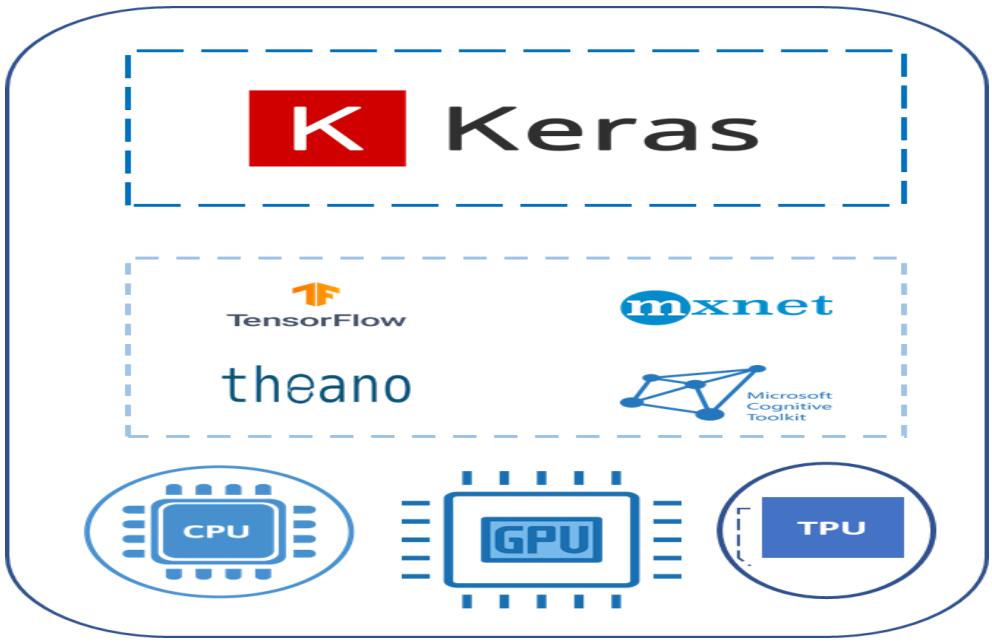

Figure 2. Keras is a sophisticated data repository that many DL models use as a foundation[13].

Advantages of Keras over Other Deep Learning Frameworks

Keras offers several advantages over other deep learning frameworks:Keras offers several advantages over other deep learning frameworks:

1. **User-Friendly Interface**: That is why Keras is the most user-friendly framework since it allows for engaging in deep learning with zero prior experience in the field.
2. **Versatile AP**: It has a rather simple API, which should be versatile for newbies and advanced users, which makes it suitable for all type of users.
3. **Customizability**: The blocks of Keras can be built individually and it makes it quite flexible which this makes it easy to expand in cases where it could be needed.
4. **Popularity**: To mention some, there are TensorFlow, Torch, and Keras, where Keras is one of the most popular deep learning frameworks after TensorFlow.
5. **Integration with TensorFlow**: TensorFlow integration to Keras implementation The TensorFlow module improves Keras implementation, allowing the direct use of TensorFlow capabilities practically in one click.
6. **Comprehensive Features**: Linking layer to all Keras features TensorFlow provides developers with versatile opportunities to create complex applications.
7. **Support for Various Architectures:** It has the following kinds of deep nets, autoencoder, CNNs, RNNs, LSTMs, deep belief nets DBNs, deep Boltzmann machines DBMs.

Furthermore, Keras makes it possible to integrate the autoencoders to help in noise reduction as depicted in Figure 3 below illustrates the model of autoencoders with Keras, TensorFlow, and deep learning. All of these make Keras a go-to framework when developing and deploying deep learning models – the flexibility tangibly improves its effectiveness, making it even more popular among machine learners.

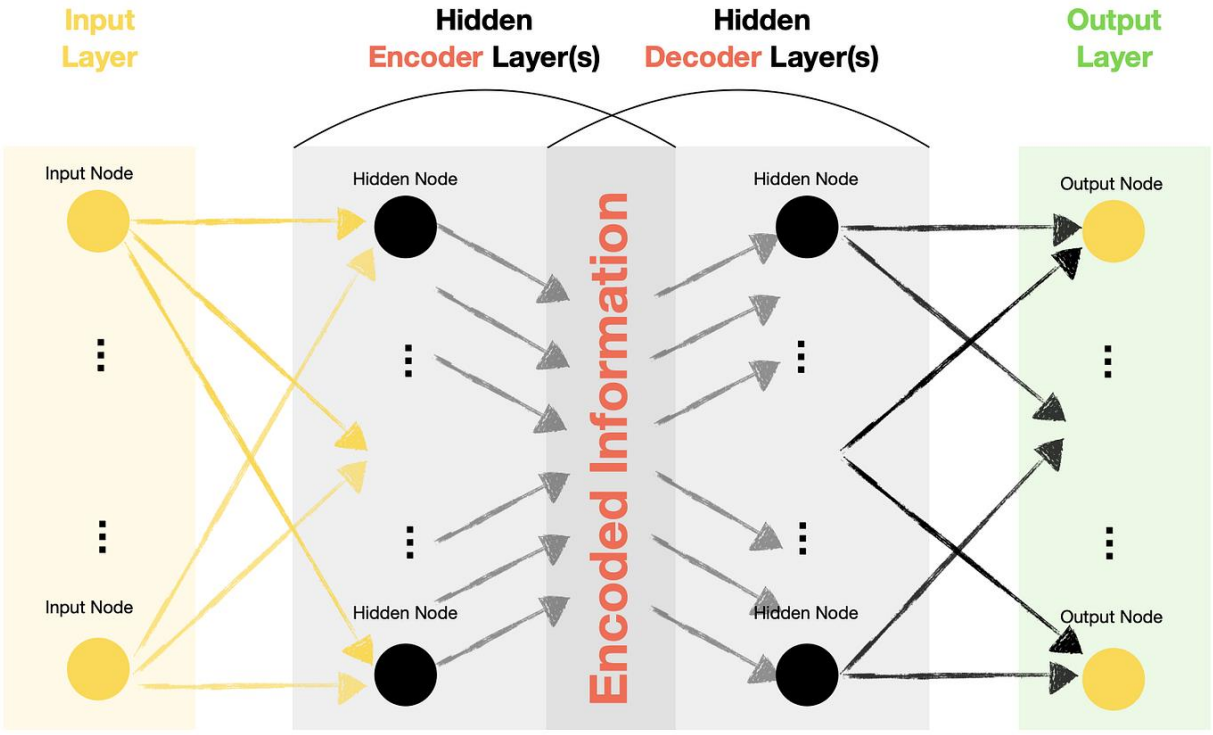

Figure 3. Keras, TensorFlow, and DL-based automatic noise-reduction encoders [14]

Python Keras Pros and Cons

Keras is a rather effective framework for those who want to build their professional experience in the context of neural networks. It provides an enhanced mind-map of execution procedures which makes it easier to develop neural network with the help of the software [16].

Pros Python Keras

a. The remedy can be quickly prepared and does not require prior preparation, as opposed to many procedures.
b. Due to extensive documentation and reception from the community, it is evident that the proposal is valid and reliable.
c. Migration for Outlook / Supported backends.
d. Availability of pre-trained models.
e. Support for multiple GPUs.

Cons Python Keras

a. Limitations with low-level API.
b. Certain features need enhancement.
c. In terms of performance, a reaction in the face of backend frameworks is considered to be slower.

Keras does a great job at hiding some of the intricate computation behind the scenes; thus, when coding NNs, one is able to create very clear and uncomplicated models. It is for this very reason that it is considered to be extremely useful irrespective of the stage of the career. It offers extensive documentation and a constructive community of users to help resolve any problem and provide a way forward to achieving more projects. However, there are certain disadvantages of Keras which includes: i) Some shortcomings are observed with the native low-level API of keras, ii) The features of keras are low and hence they need to be enhanced. However, it is worth noting that Keras still stands as one of the best frameworks because of the following reasons; it is a modular framework, it supports multiple backends and it can be used alongside GPUs. Further, the use of pre-trained models reduces the time needed for building models, thereby allowing the users to use complex models effortlessly for trying.

Kinds of Keras Models

Users can develop models using either the Sequential API or the more advanced Functional API, both of which are offered by Keras.

Sequential Model in Keras

Sequential Model enables a step-by-step approach to model building, where layers are added one at a time. However, it is not effective for constructing models with multiple inputs and/or outputs. Certainly, this model is best suited to simple stacks of layers starting from a single input tensor and emerging in a single output tensor. If any layer among these involves multiple inputs or outputs, the Sequential Model is ineffective in handling the situation. Nonetheless, it has been found out that this model may not be particularly suitable for non-linear topological conditions [17]. Figure 4 Sample example of Sequential models.

```
from keras.models import Sequential
from keras.layers import Dense

# Initialize Sequential model
model = Sequential()

# Add layers to the model
model.add(Dense(units=64, activation='relu', input_shape=(100,)))
model.add(Dense(units=32, activation='relu'))
model.add(Dense(units=10, activation='softmax'))

# Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy',
metrics=['accuracy'])

# Display the model architecture
model.summary()
```

Figure 4. Script for a Sequential Model Example.

In the above example: A sequential model is a feedforward neural network with three layers: a first hidden layer with 64 units of input data, a second fully connected layer with 32 units of input data, and an output layer with 10 densely connected nodes with softmax activation functions for multi-class classification. This model follows a sequential sequence of layers, with no feedback from the next layer back to previous layers, allowing for efficient multi-class classification.

Functional Application Programming Interface in Keras

Keras library, the Functional Application Programming Interface (API) to define models and layer connections offers more option than the Sequential Model. By including this API, it is easier to create models with multiple datasets to be fed as input or getting with multiple results coming out as output and it also allows for the integration of layers at will. As you can imagine, the Functional API in Keras allows the creation of layer graphs, in other words. Functional API focuses on a versatile software bundle and can be easily saved in one file; it is becoming increasingly convenient to replicate the same model without the code mentioned above. Besides, it makes the process of visualizing the graph's nodes and analyzing their features less complex [17]. Figure 5 depicts an example of Functional API, now let me explain it in detail.

```
from keras.models import Model
from keras.layers import Input, Dense

# Define input layer
input_layer = Input(shape=(784,))

# Define hidden layers
hidden_layer1 = Dense(128, activation='relu')(input_layer)
hidden_layer2 = Dense(64, activation='relu')(hidden_layer1)

# Define output layer
output_layer = Dense(10, activation='softmax')(hidden_layer2)

# Create the model
model = Model(inputs=input_layer, outputs=output_layer)

# Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Display the model architecture
model.summary()
```

Figure 5. Script for a Functional API Model Example.

In this example, the Functional API of Keras is used to build a simple neural network with multiple layers and shows the definite features of this API.

The example consists of an input layer with a shape similar to the input data shape, two more dense hidden layers derived from the previous layer, and an output layer with a density of 10 units and a softmax activation function. The model is defined using the Architecture of Model, where the input and output layers are specified. The architecture can be depicted using summary () after creating the model.

Distinction Between Tensorflow and Keras

The two widely used deep learning software tools, especially by experts within the subject area, are TensorFlow and Keras. These frameworks are not very dissimilar and are quite popular within the deep learning domain in general, and both serve as a massive portion of the deep learning utilization. However, there are some significant differences between TensorFlow and Keras even if they operate in a similar manner [18], as summarized in the following Table 1.

Table 1. Differences between TensorFlow & Keras.

TensorFlow	Keras
System with high-level & low-level APIs.	TensorFlow, CNTK, & Theano-backed API.
Preferred for analyzing complex neural networks.	Suitable for rapid deployment of models.
Developed by Google's Brain team.	First started via Francois Chollet & developed with a community.
Mainly coded in C++, CUDA, and Python	Python-based interface for Theano, TensorFlow, & CNTK.
Typically used for large datasets and high-performance models.	Often employed for handling small datasets
Support from various tech. organization	Lacks extensive community outreach.

TensorFlow is, in fact, widely appreciated for its capability for both high-level and low-level API integrations, which is why it is preferred when it comes to analyzing various neural networks. TensorFlow: Originally designed by the Google's Brain team with codes In C++, CUDA, and Python, this is often used for high performant models and large scale data sets. It is also boosted by the support of several technology firms since it's an online platform.

Keras, on the other hand is an API that can work with TensorFlow, CNTK and Theano as its back ends. It is good for developing a pitch of models, initially designed by François Chollet and later on with the help of community. Keras is primarily implemented in IDL and Python further acting as an interface for Theano, TensorFlow, and CNTK. It is mainly used for managing comparatively modest-sized data sets, and not as much support from a community as TensorFlow has. They point out that deep learning actually encompasses a range of approaches and frameworks, each with its own utility and application.

RELATED WORKS

Theano, a CPU- and GPU-optimized Python library for efficient mathematical computation of large-scale neural networks, was built for high-level development with the ability to easily incorporate deep learning models and research. This section presents the literature selection with focus on the application of data mining using Keras in recent research activities. The following table presents an overview of Keras work-related approaches to the various methods:

DL refines system performance through comprehensive layers of abstracted data representation and capture of hierarchical structures at various levels of detail. It enhances the traditional ML process by a large margin and gives better results than conventional methods. DL is good at feature learning where they tend to learn features by themselves from the raw data hence the formation of higher level features from the lower level features. The fact that it can indeed solve complicated problems at a brisk pace is attributed to the presence of elaborate systems with deep-seated parallelism. DL methodologies can contain several components could be convolutions, pooling layers, fully connected layers, gates, the memory cell, activation functions, and encode/decode algorithms that are based on a specific network type. It offers a pragmatic and efficient way to build large and even deep ANNs since the biological neuron can act as a global optimizer.

There is Artificial Neural Networks, models which mimic the functions of the di140 mammalian brain, are hosts of connected computing units that work simultaneously. Thus, the adapted structures of ANNs are

decentralized and flexible with respect to information flow and adaptive laws. Initially, ANNs were designed to mimic functional characteristics of a brain during pertinent cognitive tasks; later, they were successfully implemented in ranging fields including pattern recognition, data compression, or optimization. Python is also used in programming ANNs and generating graph neural networks, and Keras is a fundamental tool in accomplishing this. Keras respects guidelines and paradigms of usability and the idea of creating a rapid prototype.

Researchers have applied the Keras in various ways and the list is very large. For instance, [28] employed deep learning using Keras to a plant classification task and came up with high precision. Some examples include; [29] who proposed SciANN, a Python toolbox for scientific computing and physics involving DL, and [30] for cardiovascular disease prediction employing DL algorithms like Keras. Furthermore, [9] has provided an open implementation of the CDNEAT algorithm by using the Keras framework, whereas Versaci [31] has discussed Wavelet TensorFlow for enhanced wavelet handling. The above cases demonstrate the multipurpose applicability of Keras and its adaptability to many fields.

Furthermore, Keras is used in a wide range of cross-disciplinary approaches; for instance, Yang et al., employed Keras in developing an effective vehicle tracking system [32], the novel m-arcsinh activation function discussed via Parisi et al. [33], developments toward graph neural networks by Grattarola and Alippi [34]. Applying convolutional and pooling layers, as well as, deep learning layers, is most straightforward implemented by use of the TensorFlow-Keras technique for significant graph models which has been done by Reiser et al [35]. Other studies involving its use include disease prediction by Manapure and coworkers [38], skin cancer/melanoma classification by Benbrahim and coworkers [39], as well as the classification of Javanese characters by Harjoseputro [40].Last, [45] presented a supervised learning approach based on Keras for the development of digital differentiators; [46] applied Big Data and DL techniques for optimizing intrusion detection systems. The wide range of successful uses of Keras is demonstrated by these examples, demonstrating its usefulness and versatility in several industries.

ANALYZING AND DISCUSSING

Many research surveys carried out in the recent past have discussed and highlighted the various possibilities and prospects of deep learning (DL) libraries for future programming processes, and many of them have envisioned healthy and constructive contributions in different fields. These investigations involve the use of various tools and algorithms to treat relevant issues with assessment methods providing insights on aspects of these methodologies.

Table 2 is a summary of aspects of a number of studies, which are taken based on six attributes that have indicated trends in the attainment of objectives in the deep learning Keras. For example, [30] employed Keras with TensorFlow to make a prediction of cardiovascular disease with next to perfect accuracy, 80%, at that, from the Stanford online healthcare repository dataset. On the other hand, [35] used Keras coupled with TensorFlow as the tool for image processing and discussed flexibility of the graph as a particular issue. Moreover, [33] demonstrated successful implementation of ai frameworks Keras and TensorFlow to attain high accuracy and dependability with 'MNIST' that recorded optimum accuracy of 99%.

Likewise, in [37], Keras was used in the analysis of plants images, attaining the maximum efficiency of 96%. As shown in table 1 with which achieves the accuracy of 3% with only 250 images. Similarly, in another study [39], theDeep Convolutional Neural Network,TensorFlow, and Keras structures were used for the classification of skin tumours in cancer images. On the other hand, Deep Learning, Keras, and TensorFlow were used for identifying COVID-19 cases in one of the previous studies [38] with high accuracy levels being recorded tochest X-ray image sets.

From the existing literature, the research indicates both benefits and drawbacks of the Keras library in the Python language. The benefits include the clear structure of the UI, the ability to deploy within minutes, well-written documentation, the availability of many tutorials and examples, the extensibility by modules and the compatibility with a number of backend frameworks and pre-trained models. However, there are a set of them

as follows: problems with API at the lowest level, some prerequisites for certain features, relatively slow performance compared to backend systems.

In conclusion, implications confirmed high levels of compatibility between Keras and TensorFlow, in which Keras was majorly used in conjunction with TensorFlow. Table 2, focuses on the application of Keras in various image processing and classification tasks, which are relevant to face detection using Convolutional Neural Networks (CNNs). The following studies can help in gauging the effectiveness of Keras with instances of TensorFlow integrated within the neural network framework, for image classification and other tasks. This is useful in face detection using CNNs directly, and further shows how well Keras works for high-accuracy image data.

Table 2. Review of Studies Related to Face Detection Using CNNs with Keras

Ref	Year	Objectives	Challenges	Dataset(s)	Approach	Key Findings	Accuracy
[35]	2021	Incorporating TensorFlow-Keras Technology	Flexibility Challenges in Large Graphs	Image datasets (graph processing)	Keras, TensorFlow	KGCNN package offers transparent tensor representation and seamless integration, with mean absolute validation error reduced to 0.148.	MAE: 0.148
[37]	2020	Automating Herbicide Usage Reduction	Challenges in Health Problem Solving	Plant images	Keras	Attained maximum efficiency of 96.3% with just 250 images.	96.3%
[38]	2020	Automated COVID-19 Identification	Challenges in COVID-19 Identification	X-ray images	DL, Keras, TensorFlow	Achieved 90-92% accuracy on a dataset of X-ray images.	90-92%
[39]	2020	Skin Tumor Recognition in Photographs	Challenges in Skin Tumor Detection	Cancer images (HAM10000)	DCNN, TensorFlow, Keras	Attained validation accuracy of 94.06% and test accuracy of 93.93%.	Validation: 94.06% Test: 93.93%
[43]	2019	Automatic Classification of Chest Diseases	Challenges in Disease Classification	X-ray images	Keras, TensorFlow	Attained accuracy of 88.76% for disease classification.	88.76%

CONCLUSION

Library of Keras, it becomes apparent that it is an easy-to-use and effective instrument in works on Python that enable the creation of deep learning (DL) models. Keras is designed to take advantage of powerful computing libraries such as Theao and TensorFlow while encapsulating most of the neural network building and training process into its simple, easy to use API. This study will seek to offer a broad perspective of the different facets captured within the Keras context included in the body of literature especially in deep learning. The review outlines some major issues faced by Keras: a number of problems with its Low-level Application Programming Interface (API) that should be dealt with; issues with the backend speed of operations. Carrying out surveys in this stratum is necessary for providing readers and users with further and as diverse an understanding of various topics as possible. Keras is built following virtuous practices, which are focused on reducing cognitive load, as well as following common practices of the software guidelines that promote the simplicity and consistency of the commonly executed tasks. Also, it leads to the formation of few but easily understandable and implementable techniques in case of failures that further increases usability. Gaining knowledge from a wealth of methods, a set of values, measures of accuracy, and substantial results expands our knowledge of this valuable library.

REFERENCES

Ott, J., Pritchard, M, Best, N., Linstead, E., Curcic, M. and Baldi, P. (2020). A Fortran-Keras deep learning bridge for scientific computing, Sci. Program., vol. 2020

Tan, S. W. B., Naraharisetti, P. K., Chin, S. K. and Lee, L. Y. (2020). Simple Visual-Aided Automated Titration Using the Python Programming Language, ACS Publications, 2020

Baptista L. (2021). Using Python and Google Colab to Teach Physical Chemistry During Pandemic. ChemRxiv. Cambridge: Cambridge Open Engage

Haji, S. H., & Sallow, A. B. (2021). IoT for Smart Environment Monitoring Based on Python: A Review. Asian Journal of Research in Computer Science, 9(1), 57-70. https://doi.org/10.9734/ajrcos/2021/v9i130215

Lee H, Song J. (2019). Introduction to convolutional neural network using Keras; an understanding from a statistician. CSAM 2019; 26:591-610

Ramasubramanian K., Singh A. (2019) Deep Learning Using Keras and TensorFlow. In: Machine Learning Using R. Apress, Berkeley, CA

Drakopoulos, G. and Mylonas, P. (2020). Evaluating graph resilience with tensor stack networks: A Keras implementation, Neural Comput. Appl., pp. 1–16

Moolayil, J. (2019). An introduction to deep learning and keras, Learn Keras for Deep Neural Networks, Springer, pp. 1–16

Bohrer, J. S., Grisci, B. I. and Dorn, M. (2020). Neuroevolution of Neural Network Architectures Using CoDeepNEAT and Keras, ArXiv Prepr. ArXiv200204634

Petra, V. and Neruda, R. (2017). Evolving KERAS Architectures for Sensor Data Analysis, Federated Conference on Computer Science and Information Systems (FedCSIS), 2017, pp. 109-112

Alyousfi, A., Deep Learning in Keras - Building a Deep Learning Model, Stack Abuse. https://stackabuse.com/deep-learning-in-keras-building-a-deep-learning-model/ (accessed May 06, 2021)

Sahin, Ö. (2021). Integrating Keras Models, in Develop Intelligent iOS Apps with Swift, Springer, 2021, pp. 137–164

Atienza, R. (2020). Advanced Deep Learning with TensorFlow 2 and Keras: Apply DL, GANs, VAEs, deep RL, unsupervised learning, object detection and segmentation, and more. Packt Publishing Ltd, 2020

"Denoising autoencoders with Keras, TensorFlow, and Deep Learning," PyImageSearch, Feb. 24, 2020. https://www.pyimagesearch.com/2020/02/24/denoising-autoencoders-with-keras-tensorflow-and-deep-learning/ (accessed May 06, 2021)

Peris, A. and Casacuberta, F. (2018). NMT-Keras: A Very Flexible Toolkit with a Focus on Interactive NMT and Online Learning, Prague Bull. Math. Linguist., vol. 111, no. 1, pp. 113–124, Oct. 2018, doi: 10.2478/pralin-2018-0010

Infante, A. and Bergel, A. (2018). Experience in Bridging Keras for Python with Pharo, p. 7

"Python Keras Advantages and Limitations," DataFlair, Apr. 22, 2020. https://data-flair.training/blogs/python-keras-advantages-and-limitations/ (accessed Mar. 25, 2021)

Bisong, E. (2019). Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners. Berkeley, CA: Apress, 2019. doi: 10.1007/978-1-4842-4470-8

Maseer, Z. K., Yusof, R., Mostafa, S. A., Bahaman, N., Musa, O., & Al-rimy, B. A. S. (2021). DeepIoT. IDS: Hybrid Deep Learning for Enhancing IoT Network Intrusion Detection. CMC-Computers Materials & Continua, 69(3), 3945-3966

Abdulazeez, A., Ed., (2021). Classification Based on Decision Tree Algorithm for Machine Learning, J. Appl. Sci. Technol. Trends, vol. 2, no. 01, pp. 20–28, 2021.

Minaee, S., Boykov, Y. Y., Porikli, F., Plaza, A. J., Kehtarnavaz, N. and Terzopoulos, D. (2021). Image Segmentation Using Deep Learning: A Survey,IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2021.3059968

Charbuty, B. and Abdulazeez, A. (2021). Classification based on decision tree algorithm for machine learning, J. Appl. Sci. Technol. Trends, vol. 2, no. 01, pp. 20–28, 2021

Yuan, Q. Shen, H., Li, T., Li, Z., Li, S., Jiang, Y., Xu, H., Tan, W., Yang, Q., Wang, J., Gao, J., Zhang, L. (2020). Deep learning in environmental remote sensing: Achievements and challenges, Remote Sens. Environ., vol. 241, p. 111716, 2020

Chicho, B. T., Abdulazeez, A. M., Zeebaree, D. Q., and Zebari, D. A. (2021). Machine Learning Classifiers Based Classification for IRIS Recognition, Qubahan Acad. J., vol. 1, no. 2, pp. 106–118

Ab Aziz, M. F., Mostafa, S. A., Foozy, C. F. M., Mohammed, M. A., Elhoseny, M., & Abualkishik, A. (2021). Integrating Elman Recurrent Neural Network with Particle Swarm Optimization Algorithms for an Improved Hybrid Training of Multidisciplinary Datasets. Expert Systems with Applications, 115441

Abdullah, D. M., Abdulazeez, A. M. and Sallow, A. B. (2021). Lung cancer prediction and classification based on correlation selection method using machine learning techniques, Qubahan Acad. J., vol. 1, no. 2, pp. 141–149, 2021

Susanty, Sahrul, M., Rahman, A. F., Normansyah, M. D., and Irawan, A. (2019). Offensive Language Detection using Artificial Neural Network, in 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), Mar. 2019, pp. 350–353. doi: 10.1109/ICAIIT.2019.8834452

Aggarwal, S., Bhatia, M., Madaan, R., and Pandey, H. M. (2021). Optimized Sequential model for Plant Recognition in Keras, in IOP Conference Series: Materials Science and Engineering, vol. 1022, no. 1, p. 012118.

Haghighat, E. and Juanes, R. (2020). SciANN: A Keras wrapper for scientific computations and physics-informed deep learning using artificial neural networks, ArXiv Prepr. ArXiv200508803, 2020

Gupta, D., Khanna, A., Bhattacharyya, S., Hassanien, A. E. Anand, S. and Jaiswal, A. (2021). Eds., International Conference on Innovative Computing and Communications: Proceedings of ICICC 2020, Volume 1, vol. 1165. Singapore: Springer Singapore, 2021. doi: 10.1007/978-981-15-5113-0

Versaci, F. (2020). WaveTF: a fast 2D wavelet transform for machine learning in Keras, ICPR Workshops.

Yang, X., Roop, P., Pearce, H. and Ro, J. W. (2020). A compositional approach using Keras for neural networks in real-time systems, in 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2020, pp. 1109–1114

Parisi, L., Ma, R., RaviChandran, N. and Lanzillotta, M. (2020). Hyper-sinh: An Accurate and Reliable Function from Shallow to Deep Learning in TensorFlow and Keras, ArXiv Prepr. ArXiv201107661, 2020

Grattarola, D. and Alippi, C. (2021), Graph Neural Networks in TensorFlow and Keras with Spektral [Application Notes], IEEE Comput. Intell. Mag., vol. 16, no. 1, pp. 99–106, Feb. 2021, doi: 10.1109/MCI.2020.3039072

Reiser, P., Eberhard, A. and Friederich, P. (2021). Implementing graph neural networks with TensorFlow-Keras, ArXiv Prepr. ArXiv210304318, 2021

Balan, K., Santora, M., Faied, M. and Carmona-Galindo, V. D. (2020). Study of Evolution by Automating Hardy Weinberg Equilibrium with Machine Learning Techniques in TensorFlow and Keras, in 2020 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA), Jul. 2020, pp. 14–19. doi: 10.1109/ACCTHPA49271.2020.9213202

Yashwanth, M., Chandra, M. L., Pallavi, K., Showkat, D. and Kumar, P. S. (2020). Agriculture Automation using Deep Learning Methods Implemented using Keras," in 2020 IEEE International Conference for Innovation in Technology (INOCON), Nov. 2020, pp. 1–6. doi: 10.1109/INOCON50539.2020.9298415.

Manapure, P., Likhar, K. and Kosare, H. (2020). Detecting covid-19 in x-ray images with keras, tensor flow, and deep learning, assessment, vol. 2, no. 3, 2020.

Benbrahim, H., Hachimi, H. and Amine, A. (2020). Deep Convolutional Neural Network with TensorFlow and Keras to Classify Skin Cancer Images, Scalable Comput. Pract. Exp., vol. 21, no. 3, pp. 379–390, 2020

Harjoseputro, Y. (2020). A Classification Javanese Letters Model using a Convolutional Neural Network with KERAS Framework, International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, pp. 106–111, 2020

Senthil Kumar K., Kumar S., Tiwari A. (2021) Realtime Handwritten Digit Recognition Using Keras Sequential Model and Pygame. In: Mahapatra R.P., Panigrahi B.K., Kaushik B.K., Roy S. (eds) Proceedings of 6th International Conference on Recent Trends in Computing. Lecture Notes in Networks and Systems, vol 177. Springer, Singapore

Nagisetty, A. and Gupta, G. P. (2019). Framework for detection of malicious activities in IoT networks using Keras deep learning library, in 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), pp. 633–637

Mondal, S., Agarwal, K. and Rashid, M. (2019). Deep learning approach for automatic classification of X-ray images using convolutional neural network, in 2019 Fifth International Conference on Image Information Processing (ICIIP), pp. 326–331

Refianti, R., Mutiara, A. B. and Priyandini, R. P. (2019). Classification of melanoma skin cancer using convolutional neural network, IJACSA, vol. 10, no. 3, pp. 409–417

Tseng, C. and Lee, S. (2019). Design of Digital Differentiator Using Supervised Learning on Keras Framework, in 2019 IEEE 8th Global Conference on Consumer Electronics (GCCE), Oct. 2019, pp. 162–163. doi: 10.1109/GCCE46687.2019.9014634

Faker, O. and Dogdu, E. (2019). Intrusion detection using big data and deep learning techniques, in Proceedings of the 2019 ACM Southeast Conference, pp. 86–93.