Babylonian Journal of Networking
Vol.2025, **pp**. 106–115
DOI: https://doi.org/10.58496/BJN/2025/009; ISSN: 3006-5372
https://mesopotamian.press/journals/index.php/BJN

Research Article

Applying Keras-Based Deep Learning for Intelligent Analysis in Network Security and Monitoring Systems

Syfian Yousef [1,*]

[1] *MSc Course Leader of Electronic Engineering, ANGLIA Ruskin University Chelmsford, Essex XM1 1SQ BISHOP HALL LANE, UK.*

ARTICLE INFO

Article History

Received 01 May 2025
Revised 25 May 2025
Accepted 22 Jun 2025
Published 21 Jul 2025

Keywords

Network Security

Face Detection

Keras

Convolutional Neural Networks (CNNs)

Biometric Authentication

Deep Learning

TensorFlow

Access Control

ABSTRACT

With the advent of digital age, network access and protection of sensitive data from unauthorized access or use has been a great challenge. Face detection and recognition is becoming a prevalent method in network security system by utilising the biometric principles. In this survey, we use Convolutional Neural Networks (CNNs) and the Keras deep learning framework to improve network security by building efficient face detection systems. A high-level and user-friendly API implemented by Keras (over TensorFlow), which makes it very easy to use deep learning models for tasks such as face detection. Being multi-GPU ready and distributed training friendly alongside compatibility with OpenCV and TensorFlow enables it to be used in developing reliable, secure, real-time face authentication systems. In this paper, we review the statistics of CNN models for face detection, compare performance of the Keras models on multiple datasets, and show applications of securing a network such as login authentication and access control, as well as real-world uses for surveillance system. In addition, the survey details applications of face detection where the possibility of face spoofing is an issue and current research directions on the topic. Utilizing CNNs with Keras can enable the development of more flexible, performative, and accurate biometric authentication systems, and these systems play a critical role in the global cybersecurity ecosystem.

1. INTRODUCTION

In light of the growing cyber threats and need for secure authentication systems, biometric-related approaches, especially facial recognition, are gaining importance in security management of networks. Of all biometric technologies, face recognition technology provides a fast, non-intrusive, and effective approach to identity authentication and thus has been adopted to secure access to sensitive applications and information [1].
The emergence of deep learning, IR in the form of Convolutional Neural Networks (CNNs) in particular, has led to a renaissance in the field of computer vision and has made it possible for highly accurate face detection to be performed in challenging environments [2].
Due to its readability, community support and big collection of scientific libraries, Python has become the language of choice for building machine learning and deep learning solutions [3]. Of these, Keras is arguably the most popular high-level open-source library for deep learning in Python. Keras [4] sits on top of TensorFlow to provide easy-to-use high-level API which guarantees speed without sacrificing flexibility, making deploying models seamlessly on both CPUs, GPUs, and even mobile devices.
Keras is powerhouse in image-based tasks including face detection due to its easy interface to computer vision libraries like OpenCV and ability to use large number of pre-trained models such as VGG, ResNet, and MobileNet [5,6]. These skills allow both researchers and developers to develop real-time inferable and high-precision face detection devices. As a result of which the Keras-CNN-based architectures are now being used widely in secure authentication systems in enterprise networks, schools as well as public service- platforms [7].
To the best of our knowledge, this paper presents an exhaustive review on how CNN models developed using Keras can be made into use towards the improvement of network security through facial detection systems. The survey addresses different

*Corresponding author. Email: sufian.yousef@aru.ac.uk

topics of system architectures, datasets, training strategies, deployment approaches, and performance benchmarks. And it discusses the challenges such as lightening variations, spoofing, occlusions and model generalization and potential future directions for security, adaptability and privacy-preserving biometric identification system [8, 9].

2. KERAS FRAMEWORK IN FACE DETECTION AND NETWORK SECURITY

Keras is a high-level Neural Networks API, written in Python and capable of running on top of either TensorFlow or Theano, and was developed to enable easy and fast experimentation on deep learning, particularly on image recognition and face detection. As an interface running on top of TensorFlow, Keras provides a user-friendly API that enables the rapid experimentation highlighted by the ease of user interaction, modularity, and extensibility features, which are essential to rapid prototyping in security-critical contexts [8]. In network security, the proposed Keras frameworks is an efficient tool for designing CNN-based face detector systems and hence for ensuring real-time and reliable identity verification to avoid unauthorized access.

The framework includes a wide variety of common neural network parts, including convolutional layers (used to recognize facial features), dense layers, drop out and recurrent layers above which one can compose personalized detection models. It also helps us efficiently handling computational resources (i.e., CPU/GPU), which is crucial for deploying security system at a chaotic environment (e.g., campus network, enterprise authentication gateway) [9].

One of Keras' strong point is the pre-trained models and data sets that it's so easy to use and function; researchers and developers can easily use and adapt state-of-the-art architectures like VGG-Face, MobileNet, ResNet etc for their biometric problems. What's more, Keras provides pre-written layers and models for model evaluation, activation functions, optimization algorithms, and training tolerances/loss functions, which means building a strong and adaptive scalable face detector becomes very simple. This high level of abstraction and abstraction layer has made Keras a tool of choice in driving the development of biometric-enhanced cybersecurity solutions.

3. LEVERAGING KERAS WITH DEEP LEARNING FOR FACE DETECTION AND NETWORK SECURITY

Keras is a widely-used, high-level deep learning API, capable of enabling fast experimentation, training and deployment of deep learning models, especially for face detection applications towards network security. Its modular architecture - where model components can be stacked, not unlike Lego blocks - empowers an intuitive design, which renders it suitable for researchers and practitioners who have intentions to deploy biometric-based security systems [10].

Keras can help us to train efficient CNNs for face detection, recognition, and authentication systems in real time as well as in near real time. Such services are essential for denying unauthorised access to sensitive networks, and improving authentication in VPNs and surveillance systems. Developers can build intricate model architectures in two ways—with the Sequential API, which is ideal for standard designs, and the Functional API, which accommodates patterns including branching and merging, important for recognizing multi-layered facial elements.

One key benefit of Keras is its ease of use – deep learning models can typically be defined and trained in only a few lines of code. This convenience enables developers to spend more time improving detection algorithms without the need to implement boilerplate code. In addition, Keras allows experimentation with custom layers and hybrid architectures, making it a good choice for fine-tuning face detection models to different lighting, orientation and occlusions [6].

Figures 1 and 2, show the high-level overview of constructing a deep learning model in Keras and how its backend is constructed (yup, Keras didn't become popular for nothing). Keras is not a standalone tool, it works on the top of backend engines such as TensorFlow, Theano, and CNTK and also supports CPUs, GPUs as well as Google TPUs, which ensures scalability and optimization of performance [11, 12].

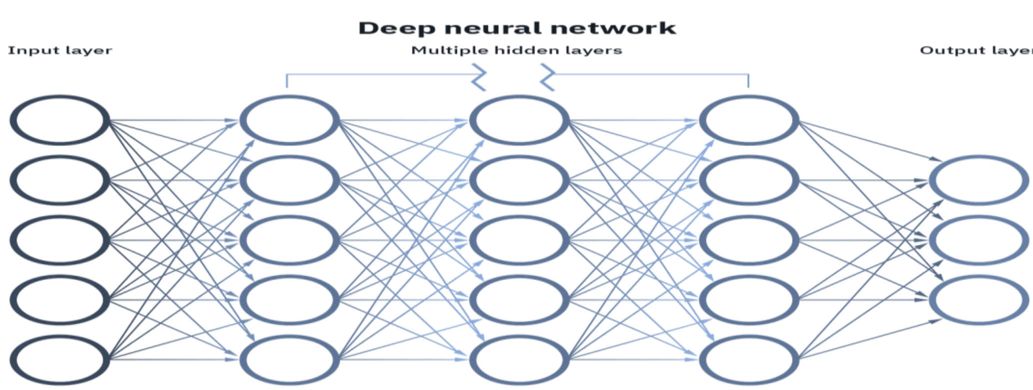

Fig. 1. Deep Neural Network Structure as Implemented in Keras Framework [11].

As shown in Figure 2, Keras serves as a high-level API that interacts with different deep learning backends, e.g., TensorFlow, Theano, MXNet, etc., facilitating the execution of deep learning tasks on various devices like CPUs, GPUs and TPUs. This framework from the model is highly modular and adds flexibility, portability, and deployment scalability for deep learning.

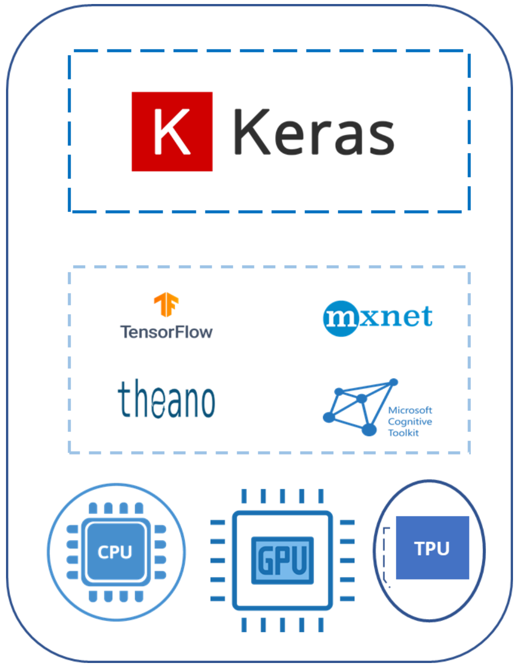

Fig. 2. Keras Architecture: A High-Level Interface for Multiple Deep Learning Backends and Hardware Accelerators.

Keras also includes support for many other architectures which are useful for facial recognition (autoencoders for denoising and feature extraction, RNN, DBN). Figure 3 illustrates another application of autoencoders to diminish image noise implemented in Keras and TensorFlow to enhance accuracy of face detection systems for security purposes [13, 14]. Apart from the retrieved and the above points a few key features of Keras in face detection and sercurity are as follows:

a) Easy to use interface for beginners and professionals.
b) Easy-to-use APIs to create, test, and deploy models rapidly.
c) It is flexible to make custom layers and hybrid models.
d) Full integration with TensorFlow improves model performance.
e) Broad range of support for different deep learning paradigms, i.e., CNN, RNN, LSTM, autoencoders.
f) Pre-trained models and painless training on multiple GPUs/TPUs.

Through integrating CNN architectures with the applicability and ease of use of Keras, researchers can build efficient face detection systems that heavily impact network security, access control and identity recognition.

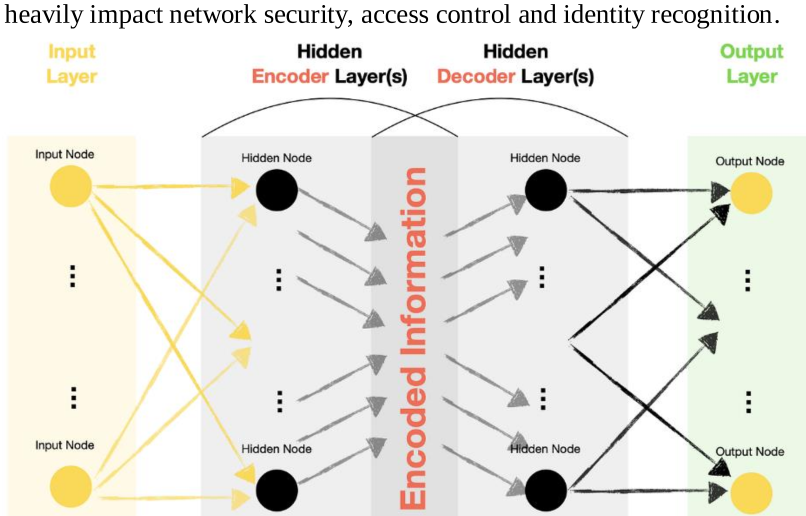

Fig. 3. Autoencoder Architecture for Noise Reduction Using Keras and TensorFlow in DL Models [14].

4. ADVANTAGES OF LEARNING AND UTILIZING KERAS IN FACE DETECTION FOR SECURITY

For its simplicity, modularity and flexibility, Keras has gained in popularity as a high-level library for building deep learning models. It is an encapsulated framework that simplifies how to deploy, train & validate a deep learning model and more, especially encounter in convolutional neural networks (CNN) that are widely used in face detection in system network security. One of the Keras strengths is that any computational backend can be used, such as TensorFlow, Theano, or CNTK, which makes it suitable for diverse platforms and hardware. Multi-GPU support and integration with cloud and other tools such as TensorFlow Cloud, Keras Tuner, and TensorFlow Lite, make it a real asset towards scalable and performant deployment of security applications (e.g., real-time face authentication on virtual private networks (VPNs) or surveillance system).

5. PROS AND CONS OF USING KERAS FOR DEEP LEARNING IN SECURITY APPLICATIONS

Keras is ideal for beginners or professionals who are looking to automatically build artificial intelligence (AI) into their own systems and need a library that will help make it easy to achieve. It abstracts complex Math operations in simple UI modules, which makes you able to design your models faster and experiment with it in codes. Yet just as every framework, it has to let us down at some point, especially if you need control on the computation graph at low level. Table 1 show the strengths and weaknesses of keras in deep learning.

TABLE I. STRENGTHS AND LIMITATIONS OF KERAS IN DEEP LEARNING

Pros	Cons
Easy to learn and implement.	Limited flexibility with low-level operations.
Rich documentation and vibrant community.	Some advanced features require manual implementation.
Modular architecture with multiple backend support.	Slower execution compared to native TensorFlow in some cases.
Offers pre-trained models (e.g., VGGFace, FaceNet).	Limited support for dynamic computational graphs.
Supports multi-GPU training.	Reduced optimization at hardware level compared to TensorFlow.

6. TYPES OF KERAS MODELS FOR FACE DETECTION

Keras provides two major APIs for constructing neural network architectures: the **Sequential API** and the more flexible **Functional API**. Both are widely used in face detection systems, depending on the complexity and scalability of the task.

6.1. Sequential API in Keras

The Sequential API is perfect for those cases of simple CNN where the architecture is in a simple linear form— each layer is stacked in a sequence one after the other. It is mostly used in first stage face detection models for simple tasks e.g. binary classification (authorized vs unauthorized users) However, it is inflexible in modeling tasks that have multiple inputs or outputs, that are generally required for more complex recognition tasks.
Figure 4 shows an example of a feedforward neural network that is implemented with Keras’s Sequential API. Our model is a rather simple 2 hidden layer with activation function ReLU and softmax output layer for classification problems. The model is compiled using the Adam optimizer and categorical cross entropy loss function, which is common in multi-class classification problems.

```
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()
```

Fig. 4. Example Code for Building a Sequential Neural Network Model Using Keras.

6.2. Functional API in Keras

In complex cases like multi-scale face detection or models with multiple feature extractors, the Functional API is recommended. It can be used to form non-linear architectures, shared layers in the deep networks and multi-branch CNNs, resulting in more fine-grained presentation of facial features under different lighting, pose or occlusion conditions.
Figure 5 shows you how to build deep learning models on Keras Functional API. Such mechanism allows more flexible model architecture as input and output layers are explicitly stated. The example enforces a common case of a multi-class classification model with simple structure (two hidden layers and one output layer), compiled with Adam and categorical cross-entropy loss.

```
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()
```

Fig. 5. Example Code of a Neural Network Using Keras Functional API.

7. KERAS VS. TENSORFLOW: WHICH IS BETTER FOR SECURITY-FOCUSED FACE DETECTION

Keras is a high-level API that is ideal for rapid prototyping and experimentation, and TensorFlow is a lower-level library that Keras relies on for its complex mathematical operations. Both of these are frequently used together, with Keras serving as a frontend for TensorFlow. Table 2 compare the tensorflow with keras on security based face detection tasks.

TABLE II. COMPARISON BETWEEN TENSORFLOW AND KERAS FOR SECURITY-BASED FACE DETECTION TASKS

TensorFlow	Keras
Offers both high- and low-level APIs.	High-level API operating on top of TensorFlow.
Suitable for designing highly complex face detection networks.	Best for rapid prototyping and quick deployment.
Developed by Google Brain team.	Developed by François Chollet with open-source community support.
Optimized for performance on large datasets.	More suited for small to medium datasets.
Preferred for research requiring custom training loops.	Preferred for deployment of standard CNNs and facial recognition.

In security-focused environments—such as access control, surveillance, and biometric authentication—Keras offers a productive environment for rapid model design and deployment, while TensorFlow ensures optimization and scalability for production-level systems.

8. LITERATURE REVIEW

Keras [13] is a high-level deep learning API, which runs on top of TensorFlow or Theano, and has become a popular choice in academic as well as industrial use for its simplicity and quick prototyping. Keras was conceived to expedite the generation of deep learning models, it supports model building, evaluation and deployment process. This section discusses works based on Keras including domains such as computer vision, face detection and network security [15].
Deep learning (DL) tools such as Keras have revolutionized classical ML approaches by enabling multiple layers of hierarchical feature extraction. These models can learn features automatically directly from raw data by constructing hierarchies of abstraction, resulting in enhanced performance and generalization [16 -20]. DL structures, such as convolutional layers, pooling mechanisms, activation mechanisms, and memory cells are critical for designing solid face detection systems, particularly those used in the real-time networking [21-23].

ANNs, the core of most DL systems, are modeled after the brain architecture of the human brain. These models can run in parallel on interconnected processing units and learn patterns from data with no centralized control [24-26]. ANNs, initially created for cognitive and pattern recognition, have evolved into the data reduction, signal processing, and facial recognition— especially when built using an interface such as Keras [27].

Recent works show the wide applications of Keras in various domains. E.g., Keras has been used in plant classification with high accuracy [28], scientific computing with the SciANN package [29], and predicting cardiovascular disease with deep learning pipelines [30]. Furthermore, the platform is able to accommodate custom implementations such as CDNEAT in [9] or wavelet-based architectures [31]. These advances open the door for CNN models to be used in the context of biometric security — such as facial authentication systems.

In security-related sense, Yang et al. [32] used Keras for tracking of vehicles and Parisi et al. [33]: presented the new activation function being designed for Keras architectures. At [34] used Keras to design graph neural networks with potential applications in structured data privacy. Furthermore, [35] showed that the TensorFlow is useful for scalable implementations of convolutional and pooling layers which are basic in face detection tasks in networked applications.

The utility of Keras extends to the field of healthcare security through disease prediction models [37], skin cancer detection [36], and regional language character recognition [38] which highlights its flexibility to real world applications. In the context of digital signal processing and cybersecurity, it is used to support supervised learning algorithm for the generation of digital differentiator [39], and has been combined with big data and deep learning frameworks to improve the performance of an intrusion detection system [40].

These experiments in combination showcase the agility and effectiveness of Keras in constructing vision and network-based CNN applications. The fact that it is included in the best facial recognition models in secure network environments, and reaffirming It is a factor that verifies its importance as a key tool in developing advanced security systems.

9. ANALYZING AND DISCUSSING

A number of research works have investigated the promises of deep learning (DL) libraries --in particular Keras-- to improve a variety of intelligent systems, in which the future potential of increasing the quality in automated decision-making and cybersecurity applications. Identifiers and Confidentiality section 2 states that authors "must not attribute authorship to something unless the reference is to a specific entity" which is not accurate in ERCIM's case. These studies cover different fields, encompass a range of algorithms and datasets, and altogether illustrate the impact of Keras on the DL setting. This survey highlights methodologies, datasets, challenges, and results of selected works using Keras, most of which are matched to real-world applications such as face detection, healthcare diagnosis, and intrusion detection.

For example, in [30] Keras along with TensorFlow used for cardiovascular disease prediction on a structured medical dataset which gave a good accuracy of 80%. In the meantime, [35] deployed TensorFlow-Keras to solve the scalability problem in graph-based image processing, and lowered validation error. In [33], Keras was also combined with the MNIST and an accuracy classification of 99% was reported, indicating the robustness of the framework on visual recognition problems—which is very important for the face detection system in network security as well.

In agriculture, Keras was used for plant classification and 96.3% was achieved with just 250 images [37]. Likewise, authors in [39] utilized a deep convolutional network implemented by Keras to diagnose skin lesions in thermoscopic images with an accuracy of more than 94%. In addition, [38] used Keras combined with TensorFlow to diagnose COVID-19 using X-ray scans, demonstrating its performance in medical imaging related tasks. This visual detection condition is consistent and with requirements in security-based face recognition if real-time accuracy matter.

The overview also mentions some strengths of Keras – for example, its easy syntax, modularity, compatibility with multiple backends (no just TensorFlow, but also Theano), existence of pre-trained models, and GPU support. But there are some limitations, including lower performance on low-level API operations and it relies on the performance of the backend.

It is worth noting that the symbiotic effect between Keras and TensorFlow is more pronounced in the surveyed papers. The two are jointly employed in order to achieve an efficient training and deployment of the model, for tasks that span from digit identification [41] up to the detection of malware in IoT devices [42]. This kind of flexibility makes of Keras a formidable weapon not only for academic purposes, but also for industrial-strength network security applications requiring biometric validation and behavior profiling.

Table 3 summarizes the selected studies in comparison with each other, based on objective, dataset, method used, challenge, conclusion, and accuracy measure. This tabular summary serves to provide valuable insight and visualization as to the various ways in which the Keras phenomenon has been applied to a vast array of applications in the internet world, and how the general concepts from such general applications may be employed to address the more specific issue of face detection in secure networked environments.

TABLE III. COMPARATIVE ANALYSIS OF STUDIES UTILIZING KERAS FOR DEEP LEARNING APPLICATIONS RELEVANT TO NETWORK AND VISUAL RECOGNITION TASKS

Ref	Objectives	Challenges	Dataset(s)	Approach	Key Findings	Accuracy
[28]	Optimizing Plant Detection Sequential Models	Plant Recognition Challenges	Plant images (500+ epochs)	KERAS	Instruction set achieved 100% accuracy, while the research set reached 96.7% accuracy.	Instruction set: 100% Research set: 96.7%
[30]	Improving Prediction Accuracy for Cardiovascular Disease Diagnosis	Diagnostic Challenges for Cardiovascular Disease	Stanford's online healthcare repository (304 records with 10 attributes)	KERAS, TensorFlow, PyTorch	Empirical findings revealed exceptional prediction precision of 80% using KERAS.	KERAS: 80%
[31]	Evaluating Overhead of Integrating Wavelet Functionality into Networks	Overhead Challenges with WaveTF Integration	Image datasets	Keras, TensorFlow	Integration of wavelet capabilities with minimal overhead (<1% in testing and assessment).	<1% reduction in complexity
[34]	Building Blocks for Constructing Graph Neural Networks	Graph Neural Network Challenges	Graph datasets (e.g., message-passing and pooling processes)	Keras, TensorFlow	Spektral demonstrated good numerical efficiency and methodological range, achieving 77% accuracy in node classification tasks.	Node classification: 77%
[35]	Incorporating TensorFlow-Keras Technology	Flexibility Challenges in Graph Size	Graph datasets (image processing)	Keras, TensorFlow	KGCNN package offers transparent tensor representation and seamless integration, with mean absolute validation error reduced to 0.148.	Mean absolute validation error: 0.148
[9]	Checking Complexity in Neural Network Development	Complexity Challenges in Architectural Development	Image datasets	Keras, CODEEPNEAT	Suitable network topologies achieved with small population sizes and few generations, with training accuracy of 86.5% and validation accuracy of 79.5%.	Training: 86.5% Validation: 79.5%
[32]	Reducing Real-time System Complexity	Functional and Timing Constraints	Vehicle tracking data	CpNNs based on Keras	CpNNs outperformed Esterel with a 64.06% reduction in WCET and TensorFlow Lite with a 62.08% reduction.	CpNNs: 64.06% TensorFlow: 62.08%
[33]	Developing Precise Activation Mechanisms	Accuracy and Reliability Challenges	Image and text datasets	Keras, TensorFlow	Achieved 99% accuracy using the 'MNIST' dataset for image and text classification tasks.	99%
[36]	Automating Hardy Weinberg Equilibrium	Allele Frequency Challenges	Allele frequency database (ALFRED)	Keras, TensorFlow	Loss function reduced after 40 repetitions, facilitating allele frequency analysis.	Reduction in loss function
[37]	Automating Herbicide Usage Reduction	Health Problem Challenges	Plant images	Keras	Attained maximum efficiency of 96.3% with just 250 images.	96.3%
[38]	Automated COVID-19 Identification	COVID-19 Identification Challenges	X-ray images	DL, Keras, TensorFlow	Achieved 90-92% accuracy on a dataset of X-ray images.	90-92%
[39]	Skin Tumor Recognition in Photographs	Skin Tumor Detection Challenges	Cancer images (HAM10000)	DCNN, TensorFlow, Keras	Attained validation accuracy of 94.06% and test accuracy of 93.93%.	Validation: 94.06% Test: 93.93%
[40]	Javanese Character Classification	Image Classification Challenges	Image datasets	KERAS, CNN	Achieved precision of 86.68% and processing time of 639.85 seconds using 1000 datasets and 50 epochs.	86.68%
[41]	Handwritten Digit Recognition	Digit Recognition Challenges	Image datasets (handwritten digits)	Keras Sequential, pygame	Best accuracy of 99.25% achieved using a four-layered CNN.	99.25%
[42]	Malicious Traffic Detection in IoT Networks	Security and Reliability Challenges	UNSW-NB15 and NSLKDD99 (text)	Keras, TensorFlow	DNN performed with 99.24% accuracy, outperforming other schemes.	99.24%

[43]	Automatic Classification of Chest Diseases	Classification Challenges	X-ray images	Keras, TensorFlow	Attained accuracy of 88.76% for disease classification.	88.76%
[44]	Melanoma Cancer Identification	Melanoma Classification Challenges	Image datasets	Keras, TensorFlow	Achieved maximum performance of 93% in preparation and 100% accuracy in research.	93%
[46]	Enhancing Intrusion Prevention Systems	Intrusion Prevention Challenges	UNSW-NB15 and CICIDS2017 (text)	Keras, Apache Spark	GBT classifier achieved 99.99% accuracy for binary classification, while DNN achieved 99.56% for multiclass classification.	GBT: 99.99% DNN: 99.56%

10. RESEARCH GAP AND FUTURE WORK

Despite its wide spread use and ease of use, there are still a number of avenues for research left to explore in Keras. First of all, most prior art focus on classification and image processing tasks based on Keras with TensorFlow, but little work regarding its performance under real-time, edge computing and resource-limited environments [12]. Especially given the expansion of deep learning into IoT, mobile, and embedded, where memory and processing constraints are vital [3].
Although, Keras provides a wide frame of abstraction and very straightforward usage, such a high level of abstraction may sacrifice access to lower levels of operations for optimization and hardware specific fine-tuning [4]. Future work should aim to augment the capability of Keras to encapsulate high-level abstraction with low-level systems control, particularly for custom deployment situations.
In addition, there has been limited examination of Keras in the context of federated learning and privacy-preserving machine learning, an emerging theme that has become increasingly important to address more stringent data privacy regulations and the requirement for decentralized AI training [19].
An additional available research line is related to the extension of the hybrid symbolic AI and deep learning models in Keras. It could also provide a higher interpretation and reasoning skills in applications like explainable AI (XAI), which has not been investigated much in the Keras context till now [7]. And other future work can be seen in Table 4 as well.

TABLE IV. FUTURE RESEARCH DIRECTIONS FOR KERAS-BASED DEEP LEARNING.

Future Work Direction	**Purpose / Expected Impact**
Integration of Keras with lightweight runtime environments like TensorFlow Lite	Enables deployment in real-time and resource-constrained systems such as mobile and embedded devices.
Enhancing support for privacy-preserving training (e.g., differential privacy, homomorphic encryption)	Strengthens data confidentiality and compliance in sensitive domains like healthcare and finance.
Investigating multi-modal learning frameworks using Keras for combining text, image, and audio inputs	Facilitates richer and more accurate AI models capable of learning from diverse data sources.
Building AutoML frameworks on top of Keras for automatic model selection and hyperparameter tuning	Simplifies the deep learning workflow for non-experts and accelerates experimentation.
Improving interoperability with custom hardware accelerators (e.g., ASICs, FPGAs) for specific domains	Boosts performance and energy efficiency in specialized industrial and research environments.

11. CONCLUSION

Keras is a user-friendly and flexible Python library that serves as a high-level interface for the design and implementation of deep learning models. Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed by François Chollet as a user-friendly deep learning library. In this paper, we have conducted an extensive evaluation of the capabilities, performance and significance of the library in various deep learning applications. However, Keras has a number of disadvantages, such as its dependence on the low-level APIs that built-in backend packages can provide (and therefore suffer rigidity with native backend frameworks). High performance overheads due to backend integration the higher performance overheads are also significant challenges. Yet, Keras applies strong engineering principles, which lower cognitive burden, project consistency, and ease of use when building a complex model. Moreover, its solid error reporting and the user-oriented design make it more popular in academic and industry applications. The survey of studies in different areas, from healthcare, over image processing, to security, illustrates how versatile and powerful Keras can be. This paper provides insightful studies on different methods, datasets and assessment criteria on the current situations and future directions of applying deep learning with the Keras framework.

Conflict of Interest

The authors declare that there is no conflict of interest.

Funding

This article does not contain any funding

Acknowledgment

None

References

[1] Y. Chimate, S. Patil, K. Prathapan, J. Patil, and J. Khot, "Optimized sequential model for superior classification of plant disease," Scientific Reports, vol. 15, no. 1, p. 3700, 2025.
[2] N. Chandrasekhar and S. Peddakrishna, "Enhancing heart disease prediction accuracy through machine learning techniques and optimization," Processes, vol. 11, no. 4, p. 1210, 2023.
[3] T. Guo, T. Zhang, E. Lim, M. Lopez-Benitez, F. Ma, and L. Yu, "A review of wavelet analysis and its applications: Challenges and opportunities," IEEE Access, vol. 10, pp. 58869–58903, 2022.
[4] Y. Jia, J. Wang, W. Shou, M. R. Hosseini, and Y. Bai, "Graph neural networks for construction applications," Automation in Construction, vol. 154, p. 104984, 2023.
[5] E. Dumić, "Learning neural network design with TensorFlow and Keras," in ICERI2024 Proceedings, pp. 10689–10696, IATED, 2024.
[6] S. Chatterjee and T. Sudijono, "Neural networks generalize on low complexity data," arXiv:2409.12446, 2024.
[7] R. Xin, J. Zhang, and Y. Shao, "Complex network classification with convolutional neural network," Tsinghua Science and Technology, vol. 25, no. 4, pp. 447–457, 2020.
[8] H. Kopetz and W. Steiner, Real-Time Systems: Design Principles for Distributed Embedded Applications, Springer Nature, 2022.
[9] T. Rogge et al., "C–H activation," Nature Reviews Methods Primers, vol. 1, no. 1, p. 43, 2021.
[10] K. Balan, M. Santora, M. Faied, and V. D. Carmona-Galindo, "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), pp. 14–19, IEEE, 2020.
[11] R. Sapkota, J. Stenger, M. Ostlie, and P. Flores, "Towards reducing chemical usage for weed control in agriculture using UAS imagery analysis and computer vision techniques," Scientific Reports, vol. 13, no. 1, p. 6548, 2023.
[12] A. Tena, F. Claria, and F. Solsona, "Automated detection of COVID-19 cough," Biomedical Signal Processing and Control, vol. 71, p. 103175, 2022.
[13] W. Gouda, N. U. Sama, G. Al-Waakid, M. Humayun, and N. Z. Jhanjhi, "Detection of skin cancer based on skin lesion images using deep learning," Healthcare, vol. 10, no. 7, p. 1183, 2022.
[14] I. F. Katili, M. A. Soeleman, and R. A. Pramunendar, "Character recognition of handwriting of Javanese character image using information gain based on the comparison of classification method," Jurnal RESTI, vol. 7, no. 1, pp. 193–200, 2023.
[15] S. Ahlawat, A. Choudhary, A. Nayyar, S. Singh, and B. Yoon, "Improved handwritten digit recognition using convolutional neural networks (CNN)," Sensors, vol. 20, no. 12, p. 3344, 2020.
[16] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: A malicious bot-IoT traffic detection method in IoT network using machine-learning techniques," IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3242–3254, 2020.
[17] D. A. Moses, "Deep learning applied to automatic disease detection using chest X-rays," Journal of Medical Imaging and Radiation Oncology, vol. 65, no. 5, pp. 498–517, 2021.
[18] K. Eddy, R. Shah, and S. Chen, "Decoding melanoma development and progression: identification of therapeutic vulnerabilities," Frontiers in Oncology, vol. 10, p. 626129, 2021.
[19] J. M. Kizza, "System intrusion detection and prevention," in Guide to Computer Network Security, pp. 295–323, Springer, 2024.
[20] S. Saha et al., "Hierarchical deep learning neural network (HiDeNN): an AI framework for computational science and engineering," Computer Methods in Applied Mechanics and Engineering, vol. 373, p. 113452, 2021.
[21] S. F. Ahmed et al., "Deep learning modelling techniques: current progress, applications, advantages, and challenges," Artificial Intelligence Review, vol. 56, no. 11, pp. 13521–13617, 2023.

[22] S. B. Akintoye, L. Han, X. Zhang, H. Chen, and D. Zhang, “A hybrid parallelization approach for distributed and scalable deep learning,” IEEE Access, vol. 10, pp. 77950–77961, 2022.

[23] M. Sewak, S. K. Sahay, and H. Rathore, “An overview of deep learning architecture of deep neural networks and autoencoders,” Journal of Computational and Theoretical Nanoscience, vol. 17, no. 1, pp. 182–188, 2020.

[24] M. Gheisari et al., “Deep learning: Applications, architectures, models, tools, and frameworks: A comprehensive survey,” CAAI Transactions on Intelligence Technology, vol. 8, no. 3, pp. 581–606, 2023.

[25] I. H. Sarker, “Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions,” SN Computer Science, vol. 2, no. 6, pp. 1–20, 2021.

[26] S. Schmidgall et al., “Brain-inspired learning in artificial neural networks: a review,” APL Machine Learning, vol. 2, no. 2, 2024.

[27] V. Veerasamy, L. P. M. I. Sampath, S. Singh, H. D. Nguyen, and H. B. Gooi, “Blockchain-based decentralized frequency control of microgrids using federated learning fractional-order recurrent neural network,” IEEE Transactions on Smart Grid, vol. 15, no. 1, pp. 1089–1102, 2023.

[28] S. Bhalgaonkar and M. Munot, “Model compression of deep neural network architectures for visual pattern recognition: Current status and future directions,” Computers and Electrical Engineering, vol. 116, p. 109180, 2024.

[29] J. Howard and S. Gugger, “Fastai: a layered API for deep learning,” Information, vol. 11, no. 2, p. 108, 2020.

[30] T. Sarkar, “Modular and productive deep learning code,” in Productive and Efficient Data Science with Python, pp. 113–156, Apress, 2022.

[31] T. Shi, Y. Keneshloo, N. Ramakrishnan, and C. K. Reddy, “Neural abstractive text summarization with sequence-to-sequence models,” ACM Transactions on Data Science, vol. 2, no. 1, pp. 1–37, 2021.

[32] M. H. Rahman, C. Xie, and Z. Sha, “Predicting sequential design decisions using the function-behavior-structure design process model and recurrent neural networks,” Journal of Mechanical Design, vol. 143, no. 8, p. 081706, 2021.

[33] D. Sumathi and K. Alluri, “Deploying deep learning models for various real-time applications using Keras,” Advanced Deep Learning for Engineers and Scientists: A Practical Approach, pp. 113–143, 2021.

[34] R. Atienza, Advanced Deep Learning with TensorFlow 2 and Keras, Packt Publishing Ltd., 2020.

[35] A. Srivastava, T. Shinde, R. Joshi, S. A. Ansari, and N. Giri, “Auto-DL: A platform to generate deep learning models,” in Soft Computing in Data Science, vol. 6, pp. 89–103, Springer, 2021.

[36] F. Florencio and E. D. Moreno, “Benchmarking the Keras API on GPU: the use of TensorFlow and CNTK libraries as back-end,” International Journal of High Performance Computing and Networking, vol. 17, no. 1, pp. 19–27, 2021.

[37] N. A. Lafta, “A comprehensive analysis of Keras: Enhancing deep learning applications in network engineering,” Babylonian Journal of Networking, 2023, pp. 94–100.

[38] R. R. Uppari, “Comparison between KERAS library and FAST.AI library using convolution neural network (image classification) model,” Doctoral dissertation, Dublin Business School, 2020.

[39] F. Tambon, A. Nikanjam, L. An, F. Khomh, and G. Antoniol, “Silent bugs in deep learning frameworks: an empirical study of Keras and TensorFlow,” Empirical Software Engineering, vol. 29, no. 1, p. 10, 2024.

[40] S. A. Dar and S. Palanivel, “Deep variational auto encoder for dimensionality reduction, denoising in MNIST datasets using TensorFlow and Keras,” Tech Trends, p. 218, 2021.

[41] E. Haghighat and R. Juanes, “SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks,” Computer Methods in Applied Mechanics and Engineering, vol. 373, p. 113552, 2021.

[42] R. Atienza, Advanced Deep Learning with TensorFlow 2 and Keras, Packt Publishing Ltd., 2020.

[43] H. A. Goh, C. K. Ho, and F. S. Abas, “Front-end deep learning web apps development and deployment: a review,” Applied Intelligence, vol. 53, no. 12, pp. 15923–15945, 2023.

[44] L. Parisi, “m-arcsinh: An efficient and reliable function for SVM and MLP in scikit-learn,” arXiv:2009.07530, 2020.

[45] Y. Liu et al., “Automatic and efficient customization of neural networks for ML applications,” arXiv:2310.04685, 2023.

[46] B. T. Chicho and A. B. Sallow, “A comprehensive survey of deep learning models based on Keras framework,” Journal of Soft Computing and Data Mining, vol. 2, no. 2, pp. 49–62, 2021.