International Journal of Advance and Innovative Research
Volume 12, Issue 2 (XXII): April - June 2025
ISSN 2394 - 7780

A STUDY ON DEEP LEARNING BASED IMAGE CLASSIFICATION TECHNIQUE USING RANDOM FOREST ALGORITHM

Pawan Kumar and Prof. (Dr.) Durgesh Pant
Department of Computer Science and Engineering Uttarakhand Open University, Haldwani Uttarakhand, India

ABSTRACT—
One essential technique for handling image classification is a convolutional neural network (CNN). It is capable of effectively representing features and has made significant progress in the field of computer vision. However, CNNs often require extensive training time. On the other hand, High classification accuracy and quick training speed are two benefits of random forests (RF). To tackle image classification challenges. This study suggests a hybrid approach that extracts characteristics using CNN, which are then input into RF for classification.

Deep neural networks are used in this study to classify images, also well-known as deep learning, through the framework of Keras. Python's integration with Keras makes it the programming language of choice. The study specifically examines input data. DNN was chosen as the optimal method for the training process because it achieved a high level of precision. The findings are examined using the image classification accuracy.

Keywords: *Deep learning (DL), Convolutional neural network (CNN), Artificial neural network (ANN), Keras*

I. INTRODUCTION

The field of image classification has been expanding quickly and is currently gaining popularity among IT developers, particularly in light of the growing availability of data in sectors like e-commerce, and healthcare. Facebook is a well-known example of this technology; with just a few tagged photos and Facebook albums, it can already recognize faces with high accuracy. This technology is approaching, and in some cases surpassing, human capability in image classification and recognition.

One of the leading approaches behind this advancement is artificial intelligence (AI) is that mimics human thought processes. Deep learning systems are typically trained with hundreds or thousands of input data points to optimize the efficiency and speed of the "training" phase. This process involves feeding large amounts of data to the system to improve its performance in recognizing and classifying images.

II. RELATED WORKS

In many different tasks, Convolutional Neural Networks (CNNs) are utilized extensively and have demonstrated excellent performance across different applications. One of the earliest successful implementations of CNN architecture was in the recognition of handwritten digits [2].

Since the development of CNNs, there have been ongoing advancements in these networks, incorporating different computer vision algorithms and adding new layers, among other things. These innovations have continuously upgrade the performance and capabilities of CNNs in solving complex image recognition and classification tasks. [3].

In the ImageNet Challenge, Convolutional Neural Networks (CNNs) are commonly utilized, where they are applied to various combinations of datasets, including those containing sketches. These models excel in recognizing and classifying complex visual data, making them highly effective for tasks that require distinguishing between different types of images, including abstract or hand-drawn sketches. [4].

On image datasets, several researchers have contrasted trained networks' detection capabilities with those of human participants. The output indicated that humans achieved a much higher accuracy rate, while the trained network's performance was slightly lower, with a 64% accuracy rate. [5].

The methods used often rely on the order of strokes to achieve higher accuracy rates. The goal of ongoing research is to better understand how Deep Neural Networks (DNNs) behave in different situations, exploring how these networks perform under different conditions to improve their overall accuracy and effectiveness [6].

The development of feature detectors and descriptors has advanced significantly, giving rise to a wide variety of technique and algorithms for classifying objects. The effectiveness of texture filters, object detectors, and filter banks is frequently compared by researchers. The study on object detection and scene classification is extensive, with a wealth of studies contributing to the advancement of this field. [11].

Researchers often utilize the most current and advanced descriptors, such as those developed by Felzenszwalb, as well as context classifiers like those introduced by Hoeim. These tools have become

ISSN 2394 - 7780

popular in the field of classify the object detection and classification due to their effectiveness in improving accuracy and contextual understanding in image analysis tasks. [12].

The concept of creating many object classifiers for image analysis is comparable to the methodology used in the multimedia society, where a huge set of "semantic concepts" is used for semantic indexing and the annotation of images and videos. Both methods aim to improve the understanding and classification of visual data by associating it with meaningful, predefined concepts. [13].

Researchers suggest that representing an picture based on the objects detected within it could be highly beneficial for complex visual recognition tasks, especially in scenes cluttered with multiple objects, where classification becomes challenging for networks. These networks not only enhance high-level recognition but also provide additional information by extracting minimum features. Usually, they are typically trained on datasets consisting of many of small images, which help improve their performance in recognizing and classifying intricate scenes. [14].

For multiclass severity classification, a researcher has put out a brand-new deep learning model named Bug Severity Classification. This model uses a combination of Random Forest with Boosting (BCR) and Convolutional Neural Networks (CNN) to address the difficulties of severity categorization. The model can more accurately and efficiently classify the severity levels of bugs or issues since it directly learns latent, highly representative features. [16].

In order to create a Hierarchical CNN based Random method for face super-resolution in a coarse-to-fine fashion, the researcher blends novel convolutional neural network and random forest methodologies. This suggested technique focuses on a broad strategy that can handle facial photographs in a variety of settings without requiring any prior processing. The HCRF method makes use of the advantages of both CNNs and random forests in an effort to efficiently improve the resolution of facial images while maintaining adaptability to various input conditions. [17].

To map glaciers coated in debris, the researcher used a novel hybrid deep learning architecture. Using various combinations of the thermal band is one of the multispectral bands of Landsat 8, as well as topography and textural characteristics for feature extraction, the framework integrates alternative CNN architectures. This will improve the accuracy of glacier mapping in regions covered in debris by combining the variety of data sources utilized in the framework. [18].

III. CONVOLUTIONAL NEURAL NETWORKS

The three main layers convolutional, pooling, and fully connected make up the fundamental structural design of a convolutional neural network (CNN). To compute the retrieved features, a nonlinear activation function is added after linear convolution is completed in the convolutional layer using a linear filter kernel. This combination allows the network to learn complex representations and patterns from the input data.

In a Convolutional Neural Network, the first layer is the convolutional layer. The fully connected layer is often the last layer, however there may be additional convolutional or pooling layers after it. The CNN's complication increases with each layer, enabling it to identify bigger portions of the image. The CNN gradually identifies bigger components or forms of the item as the visual input is processed through the layers, eventually recognizing the desired object.

The layers in a Convolutional Neural Network (CNN) include:

1. **Convolutional Layer**: Filters are applied to the input, identifying key features like edges and textures through convolution operations.
2. **Activation Layer (ReLU)**: After each convolution, to assist the network learn more intricate characteristics, non-linearity is introduced using an activation function like ReLU.
3. **Pooling Layer**: This layer down samples the feature maps to make them smaller, usually with max or average pooling, retaining important features while reducing the computational cost.
4. **Fully Connected Layer**: At the final stage, the classification result is produced by feeding the flattened feature maps into fully connected layers.
5. **Dropout Layer (optional)**: Neurons are randomly turned off during training to prevent overfitting and improve generalization.

International Journal of Advance and Innovative Research
Volume 12, Issue 2 (XXII): April - June 2025
ISSN 2394 - 7780

6. **Batch Normalization Layer (optional)**: Inputs to each layer are normalized to enhance training efficiency and stability.

These layers work together to progressively capture features needed for accurate image classification. Following each convolution operation, a Convolutional Neural Network (CNN) transforms the feature map using a Rectified Linear Unit (ReLU), which gives the model nonlinearity. This helps the network capture complex patterns in the data, enhancing its ability to learn intricate relationships.

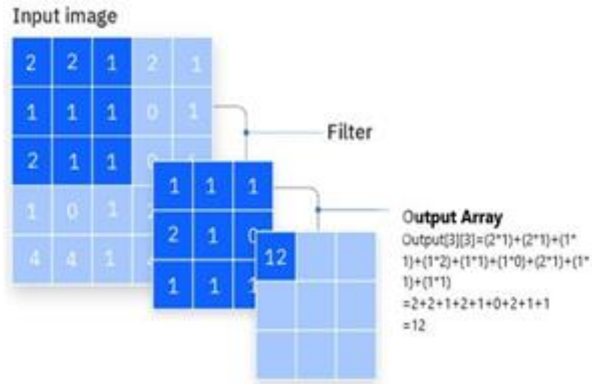

Fig. 1 – ReLu transformation to feature map

IV. RANDOM FOREST

Random forests are decision tree algorithms and comprise the family. Decision tree uses a tree-structured approach to model both regression or classification tasks, with the node of such structure represented by a feature of input space, a leaf at the end of the branch representing the relevant output values, and the branching as a single choice. Such decision trees take hierarchal order for predicting according to multiple derived decisions from given features.

The optimal binary cut point for a feature is determined using the Gini index. The impurity of the set D is the Gini index, which is expressed as Ggini(D). In an N-class classification issue, The following is the definition of the Gini index for a certain set of samples D:

$$Ggini(D) = 1 - \sum_{i=1}^{N} p_i^2$$

In the set D, Where p_i is the part of samples belonging to class i. This formula calculates the probability that a randomly selected sample will be misclassified if it is randomly assigned labels based on the distribution of labels in the subset D. A lower Gini index indicates a purer node, while a higher index suggests greater impurity.

V. HYBRID MODEL BASED ON DEEP LEARNING AND RANDOM FOREST

The hybrid model structure is shown in the image below; the sole enhancement is the fact that the RF classifier uses the CNN output layer for classification. Initially, convolutional and pooling layers with random weights are used to extract features from the picture. To obtain classification outputs, the retrieved feature is subsequently fed into the classifier RF. The amount of filters in the convolutional layer has a major effect on the model's ability to generalize; empirical observations indicate that this parameter may be changed to maximize performance.

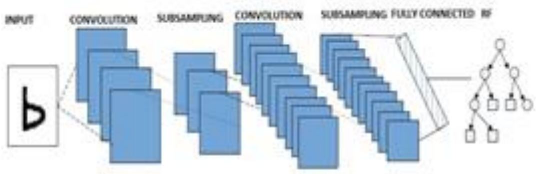

Fig. 2 – Hybrid model structure

ALGORITHM.

Input

1. Sample set *D* = $x_1, x_2, \cdots, x_n$; number of divided characteristic.
2. Use sampling to choose n samples from sample set D.

ISSN 2394 - 7780

3. To create a decision tree, randomly choose k characteristics and select the best split attributes.
4. To create a decision tree, repeat Steps 1 and 2 m times.
5. Create a random forest tree and use majority voting to determine which class the data belongs to the test set T.
6. Random forest of *m* trees.

OUTPUT

VI. FRAMEWORKS AND LIBRARIES

Keras: Keras is a Python-based high-level neural network API, compatible with TensorFlow and other lower- level frameworks. It is designed with user-friendliness, modularity, and flexibility in mind, enabling quick experimentation and prototyping. Dense layers, convolutional layers, recurrent layers, dropout layers, and their variants are among the neural network components that Keras supports. To optimize performance, it dynamically manages system resources like the GPU and CPU.

Keras streamlines the creation and training of models, making model testing simple by requiring only basic specifications, training epochs, and evaluation metrics. This ease of use reduces the amount of code necessary to implement most deep learning models. By using Keras, users can boost productivity and focus more on refining deep learning algorithms or other essential tasks. Standard models can be quickly built using its sequential model API with just a few lines of code.

Additionally, Keras allows for the creation of complex, graph-based structures, supports layer reuse, and facilitates model development with its Functional API, which enables models to behave like Python functions. It also integrates new or experimental deep learning frameworks and layers. With its customizable Model and Layer classes, Keras has become a powerful tool for building sophisticated neural networks with minimal effort.

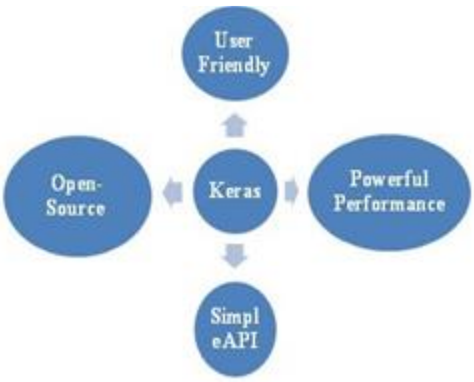

Fig. 3 - Features of Keras

VII. TRAINING IMAGES

The input data for this paper consists primarily of thousands of images sourced from Google Image. Google Images was created by Google and was originally known as Google Image Search. It allows users to do targeted online searches for photos. It allows users to find and view visual content from across the internet by entering keywords or phrases.

In this study, cat and dog images were freely obtained from the Google Images website for research purposes. The focus of this paper is to classify these images into distinct categories. The dataset includes cat and dog, each represented by thousand of images with varying angles and colors. In total, there are 20,000 images.

 ISSN 2394 - 7780

Fig. 4 – Pictures of Cat

Fig. 5 - Pictures of Dog

VIII. IMPLEMENTATION OF CNN

As illustrated in Figure 6, the model uses cat and dog images as input data, which are processed through multiple hidden layers during training. Each input image is resized in RGB format. The convolution process is performed using, an efficient architecture for building convolutional neural networks.

Fig. 6 – The Block diagram for Image Classification

As shown in Figure 7, the flowchart outlines the process of image classification implemented using Tensor Flow, with Python as the programming language. The process begins with the collection of dog and cat images. These images are then fed into a Deep Neural Network for model training. The system undergoes validation or testing, and if the image is not classified correctly as cat and dog, the process restarts with DNN training. The procedure concludes when the output is correctly classified as cat and dog.

 ISSN 2394 - 7780

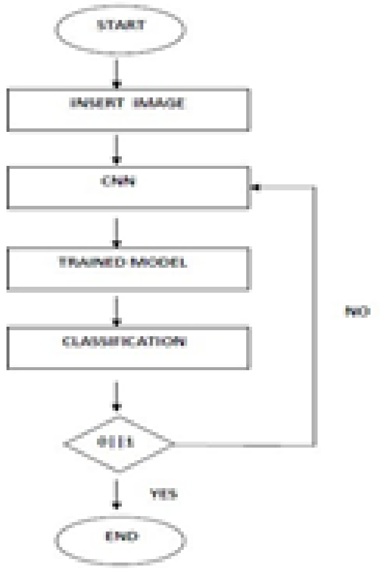

Fig. 7 – The Flow Chart for Image Classification

These images undergo a training phase using the deep neural network (DNN). The DNN is trained on the entire set of 20,000 images until the system can accurately recognize cat and dog. During classification, the model tests each image to determine cat and dog.

CNN model summary shown in Fig. 8 in this model we take total parameters 14,848,193 (56.64 MB), trainable parameters 14,847,745 (56.64 MB) and non-trainable parameters 448 (1.75 KB).

Layer (type)	Output Shape	Param #
conv2d_11 (Conv2D)	(None, 254, 254, 32)	896
batch_normalization_3 (BatchNormalization)	(None, 254, 254, 32)	128
max_pooling2d_11 (MaxPooling2D)	(None, 127, 127, 32)	0
conv2d_12 (Conv2D)	(None, 125, 125, 64)	18,496
batch_normalization_4 (BatchNormalization)	(None, 125, 125, 64)	256
max_pooling2d_12 (MaxPooling2D)	(None, 62, 62, 64)	0
conv2d_13 (Conv2D)	(None, 60, 60, 128)	73,856
batch_normalization_5 (BatchNormalization)	(None, 60, 60, 128)	512
max_pooling2d_13 (MaxPooling2D)	(None, 30, 30, 128)	0
flatten_4 (Flatten)	(None, 115200)	0
dense_11 (Dense)	(None, 128)	14,745,728
dropout_2 (Dropout)	(None, 128)	0
dense_12 (Dense)	(None, 64)	8,256
dropout_3 (Dropout)	(None, 64)	0
dense_13 (Dense)	(None, 1)	65

Fig. 8 – CNN model summary

IX. EXPERIMENTS AND RESULTS

We experimented with the well-known datasets to evaluate our hybrid model's classification performance. This training dataset consists of 20,000 images organized into folders, with each folder representing 10,000 images of cat and dog while test dataset consists of 5,000 images organized into folders, with each folder representing 2,500 images of cat and dog. The images have been sourced from Google Images. The final procedure concludes when the output is correctly classified as cat and dog. To benchmark the proposed model for image recognition, we used metrics such as accuracy, precision and loss. These metrics provide a useful way to evaluate the model's performance from a variety of perspectives. Furthermore, we used the trained model to classify individual images from the training and testing data sets, and then we calculated the prediction accuracy. We did this by feeding each image into the model, which generated outputs that predicted the class of the image.

Fig. 9 – Classified as Dog

Fig. 10 – Classified as Cat

To pinpoint regions that require improvement, Fig. 11 shows a graph that displays the model's accuracy distribution over individual images in both the training and testing data. While the y-axis shows value accuracy, the x-axis shows the range of accuracy values from 0 to 4. The accuracy values of the majority of the images in both data sets are within a small range, suggesting that the model performs consistently. The model appears to have trained to recognize correctly based on the high average accuracy on the training data. However, the average accuracy in the testing data is somewhat lower, indicating that there could be room for improvement and further optimization.

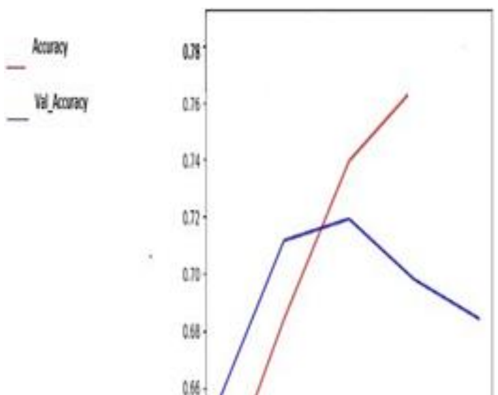

Fig. 11 – Graph Accuracy vs Val_Accuracy

To find areas for improvement, Fig. 12 shows a graph that shows how the model's loss is distributed over individual images in both the training and testing data. The y-axis shows value loss, while the x-axis shows the range of loss values from 0 to 4. The last tiny difference between the red and blue lines indicates that the loss is negligible.

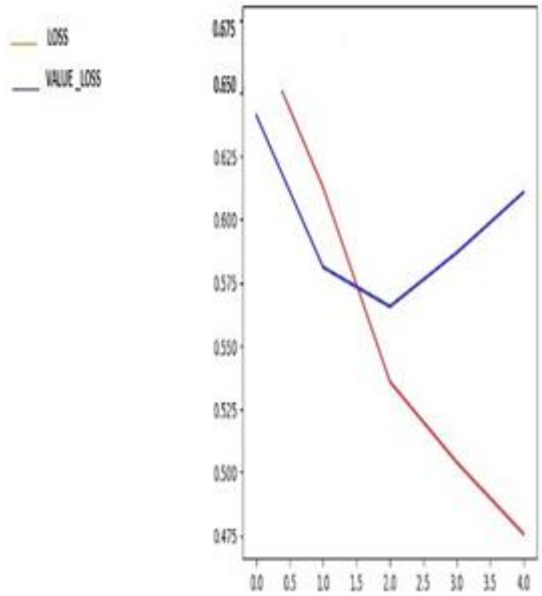

Fig. 12 – Graph Loss vs Val_Loss

X. CONCLUSION

In this work, unique approach to image recognizes using Random Forest (RF) decision trees and Convolution Neural Networks (CNN) is presented. To improve the suggested method's classification power, we use numerous decision trees of an RF. The CNN in the suggested method uses 3 max-pooling layers, 2 convolutional layers, 3 batch normalization layers, flatten layers, 2 dense and drop out layers to extract features from images and use those features to create a decision tree. The experimental findings demonstrate that the approach successfully learns characteristics to produce good results for images. Our results show that the model can successfully identify images.

REFERENCES

[1] Said Nawar, S.; Mouazen, A.M. (2017): Comparison between Random Forests, Artificial Neural Networks and Gradient Boosted Machines Methods of On-Line Vis-NIR Spectroscopy Measurements of Soil Total Nitrogen and Total Carbon. In: Sensor, volume 17, pp. 2428-2450.

[2] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998) “Gradient- based learning applied to document recognition.” proceedings of the IEEE 86(11): 2278-2324.

[3] Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., &Salakhutdinov, R. (2014) “Dropout: a simple way to prevent neural networks from overfitting.”Journal of machine learning research 15(1): 1929-1958.

[4] Eitz, M., Hays, J., & Alexa, M. (2012) “How do humans sketch objects?”ACM Trans. Graph., 31(4).

[5] Yang, Y., &Hospedales, T. M. (2015)“Deep neural networks for sketch recognition”.

[6] Ballester, P., & deAraújo, R. M. (2016, February)“On the Performance of GoogLeNet and AlexNet Applied to Sketches.” in AAAI.

[7] Fernández-Delgado, M.; Cernadas, E.; Barro, S. (2014): Do we Need Hundreds of Classifers to Solve Real World Classifcation Problems? In: Journal of Machine Learning Research, volume 15, pp. 3133-3181.

[8] Ahmad, M.W.; Mourshed, M.; Rezgui, Y. (2017):Trees vs Neurons: Comparison between Random Forest and ANN for high-resolution prediction of building energy consumption. In: Energy and Buildings, volume 147, pp. 77–89.

[9] Banerjee, M.; Ding, Y.; Noone, A. (2012): Identifying representative trees from ensembles. In: Statistics in medicine, volume 31, pp. 1601- 1616.

[10] Ellis, P. (2016): Extrapolation is tough for trees! http://freerangestats.info/blog/2016/12/10/extrapolation.

[11] Li, L. J., Su, H., Lim, Y.,& Li, F. F. (2010, September) “Objects as Attributes for Scene Classification.”ECCV Workshops(57-69).

[12] Srinivas, S., Sarvadevabhatla, R. K., Mopuri, K. R., Prabhu, N., Kruthiventi, S. S., &Babu, R. V. (2016) "A taxonomy of deep convolutional neural nets for computer vision."

[13] Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., &Fei-Fei, L. (2014)"Large-scale video classification with convolutional neural networks." in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.

[14] Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997) "Face recognition: A convolutional neural-network approach."IEEE transactions on neural networks, 8(1):98-113.

[15] Ciresan, Dan, Ueli Meier, and JurgenSchmidhuber, (2012) "Multi- column deep neural networks for image classification."2012 IEEE Conference on Computer Vision and Pattern Recognition.

[16] Ashima Kukkar, Rajni Mohana, Anand Nayyar, Jeamin Kim, Byeong-Gwon Kang, and Naveen Chilamkurti, "A Novel Deep- Learning-Based Bug Severity Classification Technique Using Convolutional Neural Networks and Random Forest with Boosting" sensors(2019), doi:10.3390/s19132964.

[17] Zhi-Song Liu, Wan-Chi Siu, and Yui-Lam Chan, "Features Guided Face Super-Resolution via Hybrid Model of Deep Learning and Random Forests" IEEE Transactions on Image Processing 2021 (Volume: 30), DOI:10.1109/TIP.2021.3069554

[18] Rahul Nijhawan, Josodhir Das & Raman Balasubramanian, "A Hybrid CNN + Random Forest Approach to Delineate Debris Covered Glaciers Using Deep Features" Journal of the Indian Society of Remote Sensing, Volume 46, pages 981–989, (2018).

[19] Saqib Qamar, Rasmus Öberg, Dmitry Malyshev & Magnus Andersson, "A hybrid CNN-Random Forest algorithm for bacterial spore segmentation and classification in TEM images", nature, scientific reports (2023).

[20] Mr. Pawan Kumar, Prof. (Dr.) Durgesh Pant, 2025, Frame Works and Libraries of AI and Machine Learning: A Comprehensive Survey, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 13, Issue 02 (April 2025).

[21] Varun Kumar, Deepak Banerjee, Deepak Upadhyay, Mukesh Singh, and Ravi Chythanya,"CNN-Random Forest Hybrid Model for Improved Ginger Leaf Disease Classification" 2024 International Conference on E-mobility, Power Control and Smart Systems (ICEMPS), DOI: 10.1109/ICEMPS60684.2024.10559352.

[22] Sharma, N., Jain, V., & Mishra, A. (2018). An Analysis Of Convolutional Neural Networks For Image Classification. Procedia Computer Science, 132, 377–384. https://doi.org/10.1016/j.procs.2018.05.198

[23] Miranda, E., Aryuni, M., & Irwansyah, E. (2016). A survey of medical image classification techniques. 2016 International Conference on Information Management and Technology (ICIMTech). https://doi.org/10.1109/icimtech.2016.7930302

[24] Xi, E. (2022). Image Classification and Recognition Based on Deep Learning and Random Forest Algorithm. Wireless Communications and Mobile Computing, 2022, 1–9. https://doi.org/10.1155/2022/2013181