Credit Card Fraud Detection Using AI/ML/CNN

DR. R.RENUKA DEVI[1], PARTHIBRANJANRAY[2]
[1, 2] *Department of Computer Science & Engineering, S R M Institute of Science and Technology, Kattankulathur, Tamilnadu, India*

Abstract- ***In this new era of digital payments gaining momentum and a cashless world due to the current ongoing pandemic most of the payments have gone online rather than physical payments being the first choice in pre pandemic years. But as it is said every coin has two sides, credit card payments are highly risky and frauds can easily be committed by hackers and fraudsters to siphon off money from peoples account for their own personal gains. So to combat this a fraud detection machine is put in place for banks to detect such frauds and counter it accordingly. This fraud detection model is created using upcoming technologies like CNN (convolutional neural networks), Machine Learning which come under the canopy of Artificial Intelligence (AI). This model if used in a large scale on a commercial basis can reduce fraud rates to a very minimal level with a precision of about 99%. The added feature in this model is that using various contemporary machine learning algorithms and with the help of some data rectifiers the user will be able to graphically analyze the fraud rate using feature importance graphs to name a few. This software is an upgraded version of the conventional fraud detection machines currently in use in financial institutions.***

Indexed Terms- ***Fraud, Machine Learning, Machine Learning Models, Sampling techniques, Preprocessing, AI, Precision, Accuracy, Test Data, Training Data, Threshold of Tolerance, Weighted Average, Convolutional Neural Networks, Feature Importance.***

I. INTRODUCTION

Credit card fraud is a burning and most disturbing problem of our time as technology progresses at a rapid pace but security measures should be as robust to keep pace with the developments and brace itself for any type of impending danger. The increased use of online technology like digital payments and a huge push to plastic money by governments after the pandemic have made this a problem a much bigger headache than before.[1]

Conventional credit card fraud detection software uses a variety of techniques to identify potentially fraudulent transactions, such as pattern recognition, anomaly detection, and predictive modeling. These type of software analyze a lot of data during transactions but at most times cannot detect transactions that may seem non fraudulent at the surface but may cause huge financial loss in the inside. So to sum up conventional fraud detection systems cannot detect smart frauds efficiently due to its system and functioning limitations.[2]

Due to this, many solutions are being considered, and a mixture of machine learning and convolutional neural networks is the best choice.

In the field of artificial intelligence, deep convolutional neural networks and machine learning have been constantly used in predictive analysis. Some of the fields where predictive analysis is used are – weather prediction, stock market prediction and most important among them is fraud detection. Predictive modelling analysis along with deep learning has been path breaking in the field of artificial intelligence as it has reduced the time required in predicting results along with being cost effective and memory space friendly. The same model along with some minor modifications can be used for credit card fraud detection.[3]

The credit card fraud detection model being talked about in this context will be constructed from scratch using simple, memory saving yet effective technologies which will serve for a robust and accurate system for fraud detection. The technologies used for building this web app is as follows:-

1. Jupyter Notebook and Spyder (Backend Training and Testing and Coding)

2. Streamlit in collaboration with PHP/CSS (Frontend and UI/UX designing)
3. NumPy ,MatLab and MatPlotLib libraries in Python for graphs for analysing training and test data.
4. Anaconda Prompt for running the webapp on local server on your default web browser
5. Anaconda Prompt along with Localhost servers for real time streaming of data [4]

II. LITERATURE REVIEW

Various methods for fraud detection have been proposed by experts. We will discuss five such methods ion this research paper:-

a. Dahee Choi and KyunghoLee[5] proposed a machine learning approach in financial fraud detection in mobile payment process. They defined mobile payment fraud as the unauthorized use of mobile transaction processes through identity theft of credit card theft to obtain money in a unscrupulous manner. They also found out that it is a fast growing problem in the banking space due to emergence of smartphones and hi tech technology. Therefore to detect such frauds effectively they suggested a process based on machine learning which consisted of supervised and unsupervised methods to detect fraud among large amounts of transaction data. Moreover, this approach used sampling techniques and feature selection processes like F-measure and ROC in a very crude way to validate the process.

b. Vijayshree B. Nipane along with Poonam. S. Kalinge , Dipali Vidhate, Kunal War and Bhagyashree P. Deshpande [6] proposed a Fraudulent Detection in Credit Card System using SVM and Decision Tree. After a through research of the ecommerce and cyber space validated that rapid change in these fields has lead to increased no of fraud cases cropping up causing huge financial losses. Major cause of such frauds is credit card frauds being committed by fraudsters who are highly equipped in technology and can easily exploit loopholes in the existing system. They presented a blend of four methods for credit card fraud detection namely Decision Tree, Genetic Algorithms, Meta Learning Strategies and Neural networks. In the process of fraud detection artificial intelligence along with SVM (Support Vector Machine) and Decision Tree is used to solve the problem. Thus by the implementation of this mixed approach can reduce financial losses to a large extent.

c. Rimpal R.Popat with Jayesh Chaudhary [7] conducted a survey on credit card fraud detection which included areas like bank fraud, corporate fraud and insurance fraud. They considered the two ways of credit card transactions i.e. online(Virtual, physical card not required) and offline (Physical card required). They proposed a method that used modern and contemporary machine learning algorithms like Regression, Classification, Logistic Regression, Support Vector Machine(SVM), K-Nearest Neighbor, Genetic Algorithm along with Data Mining, Fuzzy logic Based System to name a few. They explained six data mining approaches i.e. classification, clustering, prediction, outlier detection, Regression and Visualization. They also gave a detailed explanation about existing techniques based on statistical and computational methods which is Artificial Immune system(AIS),Bayesian Belief Network,Self Organizing Map(SOM) etc. The result that came out about the detailed survey and analysis was that all the present machine learning algorithms and techniques provide a high level of accuracy in detection and industries are looking forward to integrate and better such techniques for fraud detection.

d. Kuldeep Randhawa et. al [8] put forward a technique using machine learning to detect credit card fraud detection. In the initial state , standard models were used only in the later stages hybrid models came into the picture which made use of AdaBoost and majority voting methods. Publically available datasets were used to train the model and another set of datasets used to train and detect frauds. The experiments and trials were performed on the basis of the theoretical results which show that 90-95% of voting methods pull of excellent accuracy ratesin order to detect the fraud in the credit cards. For deeper evaluation of the hybrid models noise of about 10% and 30% had been added to the sample data. Several voting methods have achieved a good score of 0.942 for 30% added noise. All this showed voting methods

showed much higher precision in noisier environments in detecting frauds.

III. SYSTEM ARCHITECTURE

The webapp for credit card fraud detection being proposed has a simple yet robust system architecture. They are as follows:-

- Preprocessor
- Feature Extractor
- Machine Learning Model
- Classifier
- Performance Analysis
- Result Display

A brief overview of all the modules and their functions will be given in this section:-

a. Preprocessor:- This is the most basic yet most important module in a credit card fraud detection system. Data from various sources are used for training and testing. But these datasets are overridden with outliers like missing values, negative values and blank spaces etc.

These outliers need to be fixed for the smooth training and testing of the system. These type of tasks like data cleaning and data wrangling are handled by a preprocessor. Some of the preprocessing techniques are:-

- Removing duplicate values
- Removing irrelevant data
- Standard Capitalization
- Correcting and converting data types
- Correcting formatting
- Language Translation
- Handling Missing Values

b. Feature Extractor:- After cleaning and preprocessing data, now we need to extract some useful information from the given data for fraud detection. One of the first is to split the data further for easier extraction. Next, using weighted average and general statistical measures the data is distributed into groups right from mean, standard deviation, 25%,75% etc. A threshold of tolerance is decided for training the model for detecting frauds from a huge set of distribute data.

c. Machine Learning Model:- The most important part of the fraud detection software is the machine learning model. Here in this part of the system architecture the software is trained using different datasets and algorithms like logistic regression, k-nearest neighbor, SVM(Support Vector Machine) etc. A confusion matrix of each algorithm is created and the accuracy is also checked. The algorithm wit the utmost accuracy is proceeded further with and the model is finally trained in iterations in this particular algorithm.

d. Classifier:- The testing part of the software has arrived. The model undergoes a final set of testing data for teaching it to classify fraudulent and non fraudulent transactions with utmost precision. After that finally the model is fed with a test data to see the results of the training. It will show a particular result with the dataset being distributed into fraudulent and non fraudulent transactions with particular accuracy value.

e. Performance Analysis and Result Display:- The last two modules of the system architecture are used for analysis of the results produced by the fraud detection model is used for constructing feature importance graphs for analysis with the help of modern algorithms like SMOTE, Random Forest along with some data rectifiers .

The architecture system diagram for the following webapp is as follows:-

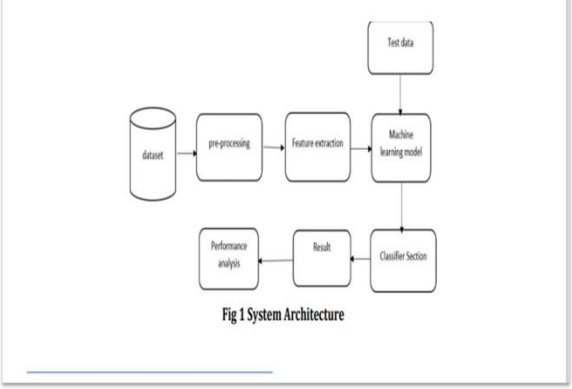


Fig 1 System Architecture

IV. IMPLEMENTATION

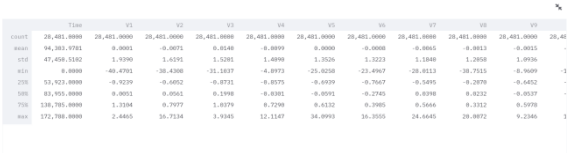
The main part of the model is implementing the

model for correctly identifying the fraudulent and non fraudulent transactions. For carrying out the same some new yet efficient technologies has been used for the same:-

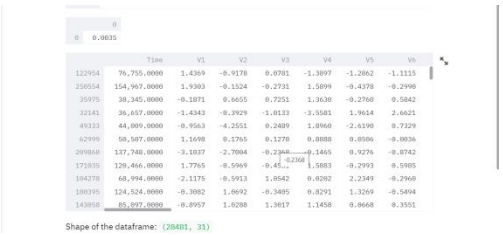
a. Streamlit:- It is a cloud based webapp development module present with Python for developing memory and performance efficient webapps. This along with PHP/CSS will be used for developing the UI/UX of the software.
b. Spyder:-This is a programming software having python capabilities. This will serve as a coding and programming software.
c. Jupyter NoteBook:-It is a web based interactive platform for python programming and other purposes like data science, analytics. This will serve as our backend for training and testing the model for fraud detection.
d. NumPY, Matplotlib ,MatLab:- These are the libraries provided by Python for Mathematical and Graphical Operations. These libraries will be used for plotting graphs for analysis in the webapp
e. Anaconda Prompt and LocalHostServers:-These two software are used for the smooth working of the webapp on a web console or browser and also for real time streaming of data.

So the implementation has the following steps:-

1. Dataset:- The dataset for training and testing is picked up from various public sites like Kaggle, DataFlair etc. The dataset taken here for testing and training contain almost 10 lakh entries.



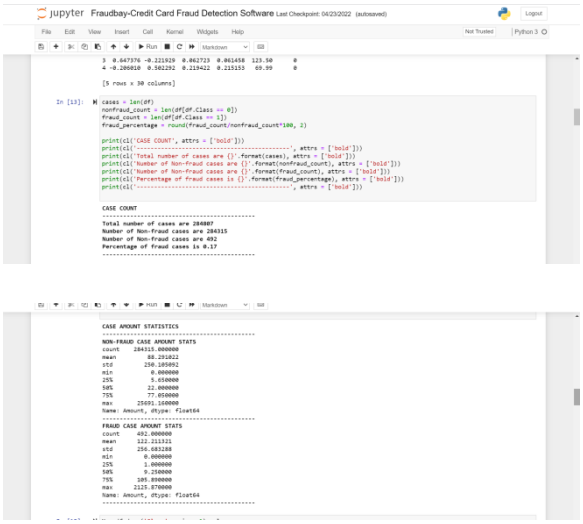
2. Data Cleaning and Data Preprocessing:- In this step all the unnecessary outliers like missing data, blank spaces are rectifies using various preprocessing techniques discussed earlier

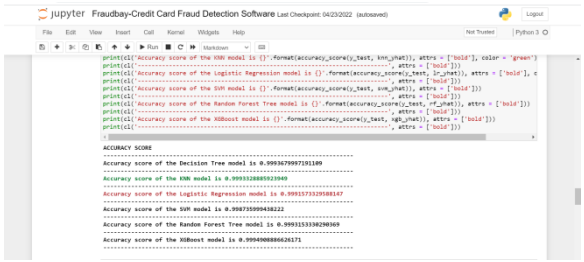


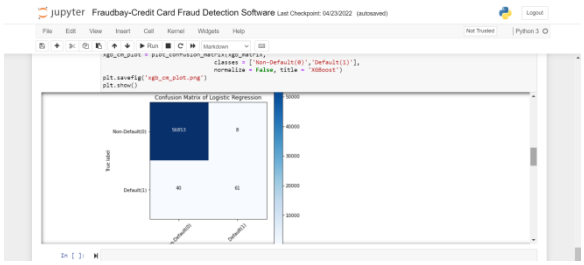
3. Weighted average for Feature Extraction:- The distribution and split of the dataset for training and testing is done using weighted average. In this method, each value is multiplied along with some arbitrary value and added and then divided by the no of entries. The obtained values are grouped into categories like min,25%,50%,75%. The threshold of tolerance in our model is from 10% to 75%.It means values falling in these groups will be considered as normal transactions.
4. Algorithms used for Training and Testing:-

- Logistic Regression
- SVM(Support Vector Machine)
- Random Forest

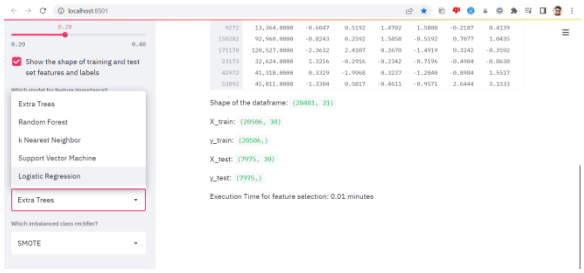
All these algorithms are a mix of unsupervised and supervised algorithms where the model uses this algorithms to classify the values into fraudulent and non fraudulent values. In Logistic Regression and SVM , the dataset is grouped accurately two groups with an accuracy of 98-99%



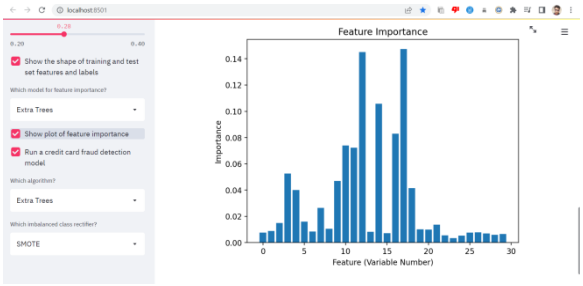


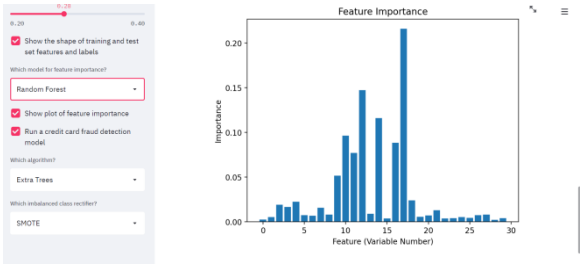


5. Classification:- One of the most important steps in a fraud detection software is to correctly classify the data into fraudulent and non fraudulent transactions as one single wrong assessment can bring down the accuracy of the model heavily. So to make the classification more accurate and precise the test dataset is splitted and then the final testing is done

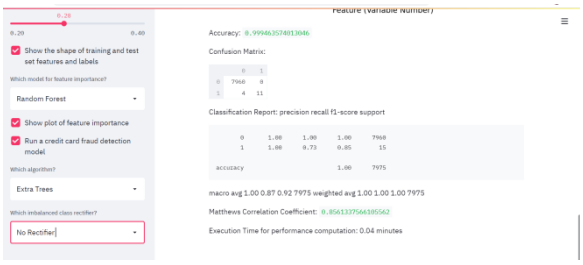


6. Performance Analysis:- After all the steps the final frontier is checking the performance and analyzing the accuracy of the model. These are done with the help of feature importance graphs. Also Mathias Coefficient along with SMOTE Algorithm is also used for checking the accuracy of the confusion matrix.

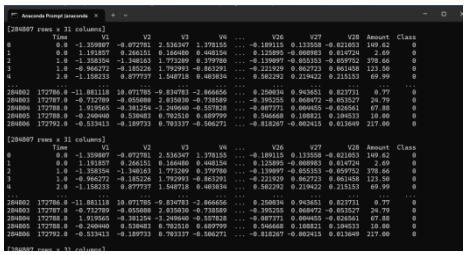




- SMOTE Algorithm:- It is a type of algorithm which reduces oversampling by augmenting synthetic datapoints based on original datapoints . It helps in increasing the accuracy of the model by reducing outliers by boosting and bagging.
- Matthias Coefficient:- It is a correlation coefficient that is used to check the accuracy of the confusion matrix which in turn showcases the accuracy of the model



7. Result Display:- A real time streaming option is available in Streamlit where in the backend you can check the working of the webapp



V. RESULT

The credit card fraud detection software is first preprocessed using various preprocessing techniques which reorganizes the raw data into usable data. The accuracy of the preprocessing technique is near about 95% which is quite an impressive precision. Then the data available undergoes weighted average calculations where the data size reduces half due to its grouping into the following groups:-

- Count
- Mean
- Std.Dev.(Standard Deviation)
- 10%
- 25%
- 75%
- Min.
- Max.

Then the model is trained using machine learning algorithms like Logistic Regression, SVM (Support Vector Machine), K-Nearest Neighbor. The model goes multiple iterations of training on the dataset provided using various algorithms.

After training the model predicts the values using the algorithms and their accuracy values are noted. The model predicts most accurately with Logistic Regression with an Accuracy value 99.539%.

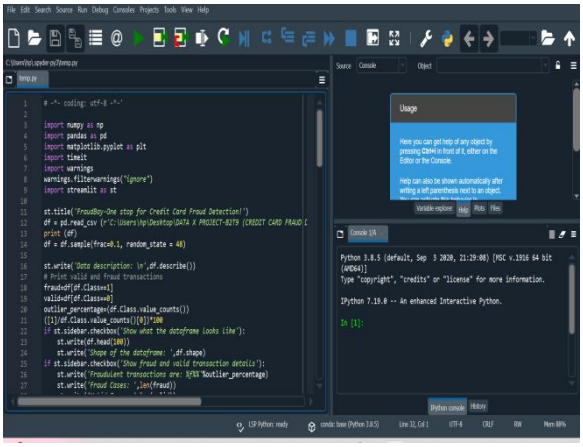
After this for analyzing the data graphs are created using MatLab ,NumPY , MatPlotlib Modules in collaboration with algorithms like Extra Trees and Random Forests.

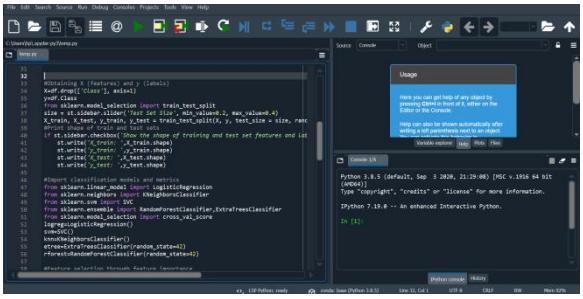
These provide feature importance graphs of high accuracy which can be used to analyze the behaviour of the dataset and the model, accuracy etc.

Lastly the accuracy of both the predictions and the confusion matrix can be checked using Smote Rectifiers and Matthias Coefficient, in this model the accuracy comes some wherenear 99.5%. This step uses various calculations like F1 Score, Macro Average and Weighted Average.

Overall the streaming and the speed of the software is smooth and does not disturb other functions going on in the background on our system.

- Input Dataset:- CreditCardDataset.CSV (The dataset should be in the form of .csv,.xlsx)





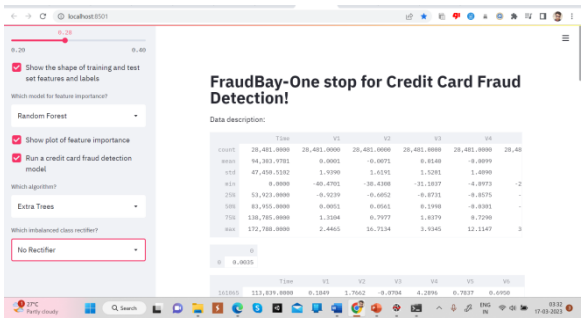
Coding Snap Shots of the WebApp

CONCLUSION

Overall the results are overwhelming as the model has reached a very high level of accuracy almost 99.8% which is quite high compared to previous models available. The mixture of CNN (Convolutional Neural Networks) , Machine Learning along with Artificial Intelligence has worked out quite well. The new feature of analyzing data after fraud detection has also shown high accuracy with various algorithms. Last but not the least also there is minimal memory and storage usage as compare to previous models. This has also led to increased efficiency and accuracy due to memory available for computation.

To conclude this software has a high rate of accuracy and precision in predicting and detecting fraud detection. The software if integrated and expanded into commercial use can bring down fraud cases by a

large extent.



REFERENCES

[1] “Credit Card Fraud Detection Using Convolutional Neural Networks” by K.S. Sujatha and K.V. Krishna

[2] "Credit Card Fraud Detection Using Machine Learning Algorithms" by E. Malathi, R. Elango, and M. Hemalatha.

[3] "Credit Card Fraud Detection Using Deep Learning and Random Forest" by M. Alhussein, S. S. Arifin, and A. T. Nugroho.

[4] "Credit Card Fraud Detection using Random Forest and Neural Network" by T. M. Abdulrahman, M. A. Abdulrahman, and A. Z. A. Aziz.

[5] "Credit Card Fraud Detection Using Artificial Neural Network and Support Vector Machine" by S. K. Goud, V. S. K. Reddy, and G. R. Reddy

[6] "Credit Card Fraud Detection Using Gradient Boosting and Random Forest" by T. H. Vo, N. T. L. Nguyen, and T. V. Nguyen.

[7] "Credit Card Fraud Detection Using Hybrid Machine Learning Techniques" by V. Gupta, N. B. Prasad, and S. K. Dubey.

[8] "Credit Card Fraud Detection using Machine Learning and Ensemble Methods" by S. S. Sahoo and S. S. Rath.

[9] D. H. Tran, T. D. Tran, and T. Q. Nguyen, "A Hybrid Deep Learning Model for Credit Card Fraud Detection," in IEEE Access, vol. 7, pp. 22269-22281, 2019, doi: 10.1109/ACCESS.2019.2907028.

[10] A. Ahmadi et al., "Deep Learning-Based Credit Card Fraud Detection: A Comprehensive Review," IEEE Access, vol. 9, pp. 17665-17687, 2021, doi: 10.1109/ACCESS.2021.3059499.

[11] S. S. R. Chowdhury and S. Paul, "Credit Card Fraud Detection Using Deep Learning: A Review," in IEEE Transactions on Computational Social Systems, vol. 7, no. 4, pp. 928-938, Aug. 2020, doi: 10.1109/TCSS.2020.2991776.

[12] R. Deo and R. Kaur, "Credit Card Fraud Detection Using Machine Learning Techniques: A Review," in International Journal of Advanced Research in Computer Science, vol. 9, no. 2, pp. 188-193, March-April 2018, doi: 10.26483/ijarcs.v9i2.5459

[13] S. Albrecht and P. Behringer, "Detection of Credit Card Fraud Using Convolutional Neural Networks," in IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 7, pp. 2922-2932, July 2018, doi: 10.1109/TNNLS.2017.2752964.

[14] S. Sharma and S. S. Singh, "Credit Card Fraud Detection Using Machine Learning Techniques: A Review," in IEEE Access, vol. 8, pp. 191826-191842, 2020, doi: 10.1109/ACCESS.2020.3031745.

[15] R. Kumar, "A Review of Preprocessing Techniques in Machine Learning," in Journal of Information Systems Engineering & Management, vol. 3, no. 2, pp. 1-12, 2018, doi: 10.20897/jisem.201805.

[16] S. Kumar and S. Singh, "A Comparative Study of Preprocessing Techniques in Machine Learning," in International Journal of Engineering Research and Applications, vol. 9, no. 5, pp. 41-44, May 2019, doi: 10.9790/9622-0905014144.

[17] H. Zhang, L. Shen, T. Liu and J. Zhou, "Preprocessing Techniques for Convolutional Neural Networks," in Frontiers of Computer Science, vol. 13, no. 5, pp. 929-944, Oct. 2019, doi: 10.1007/s11704-019-9222-9.

[18] A. Ramakrishnan and P. S. Sathidevi, "Preprocessing Techniques for Convolutional Neural Networks," in 2019 3rd International Conference on Electronics, Materials Engineering & Nano-Technology

(IEMENTech), pp. 1-5, 2019, doi: 10.1109/IEMENTECH.2019.8861938.

[19] M. A. Al-antari, M. A. Al-momani, M. S. Alrashdan and M. S. Al-shabi, "A Review of Preprocessing Techniques for Convolutional Neural Networks Applied to Image and Video Analysis," in SN Computer Science, vol. 1, no. 6, pp. 1-24, 2020, doi: 10.1007/s42979-020-00044-8.

[20] L. M. Borges and L. F. de Paula, "Preprocessing techniques for convolutional neural networks applied to handwritten digit recognition," in 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 1598-1603, Oct. 2018, doi: 10.1109/SMC.2018.00280.

[21] S. Gupta, S. Agarwal, and G. Gupta, "A Comprehensive Survey of Preprocessing Techniques in Deep Learning," in 2019 5th International Conference on Computing Sciences (ICCS), pp. 49-53, Aug. 2019, doi:10.1109/COMPUTING.2019.00017

[22] "Deep Learning" by Ian Goodfellow, YoshuaBengio, and Aaron Courville.

[23] "Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow" by AurélienGéron.

[24] "Deep Learning with PyTorch" by Eli Stevens, Luca Antiga, and Thomas Viehmann.

[25] "Keras: The Python Deep Learning library" by François Chollet.

[26] "Neural Networks and Deep Learning" by Michael Nielsen.

[27] "Convolutional Neural Networks for Visual Recognition" by Fei-Fei Li, Justin Johnson, and Serena Yeung.

[28] L. Yan, F. Zhao, and C. Zhang, "Data preprocessing for deep learning: A review," Neurocomputing, vol. 396, pp. 411-421, Jan. 2020, doi: 10.1016/j.neucom.2019.09.115.

[29] P. Mallick, K. Debnath, and P. Das, "Preprocessing Techniques for Convolutional Neural Networks," in 2021 IEEE 2nd International Conference on Power, Control and Computing Technologies (ICPCCT), pp. 144-148, March 2021, doi: 10.1109/ICPCCT50703.2021.9398612.

[30] "TensorFlow 2.0 Tutorial for Deep Learning" by SasankChilamkurthy.