ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

DEEP FAKE IMAGES AND VIDEOS DETECTION USING DEEP LEARNING TECHNIQUES

[1]M.Harshavardhan, [2] N.Madhu, [3] M.Chanikya, [4] M. Sai Hrushikesh, [5] Mrs. Sandhya Rani,

[1,2,3,4] U.G.Scholor, Department of ECE, Sri Indu College Of Engineering & Technology, Ibrahimpatnam, Hyderabad.

[5]Professor, Department of ECE, Sri Indu College Of Engineering & Technology, Ibrahimpatnam, Hyderabad.

ABSTRACT

Using the Deep Fake dataset, this research will evaluate Convolutional Neural Networks (CNNs) with Decision Tree classifiers for detecting bogus photos. Tools and Procedures: In this study, the Deep Fake dataset is used. There are 20,000 training photos, 4,000 photographs for development, and 4,500 images for testing. The dataset is then divided in half. To increase accuracy, half of the dataset is trained using decision tree and convolutional neural network techniques for 10 iterations. By adjusting the following values in GPower 3.1: 80% power, 95% confidence interval, alpha = 0.05, and beta = 0.2, we may estimate that there are 40 participants in each of the training set classes's groups. Regarding accuracy, Using a significance threshold of 0.0293' ($p<0.05$), the data shows that the Decision Tree achieves an accuracy of 80.13% & the Convolutional Neural Network 82.22%. Because the two datasets are so different, it's clear that the Convolutional Neural Network is better at identifying fake photos than the Decision Tree. Using the precision measure, the study concludes that two ML models, a Decision Tree and a Convolutional Neural Network (CNN), are effective at detecting fake photos. The results show that when it comes to identifying phoney images, Convolutional Neural Networks are far better than Decision Trees.

Issues such as web-based, deep learning, decision trees, vulnerable, deceptive pictures, and machine learning are all connected to this matter.

INTRODUCTION

As image-editing technology evolves, the prevalence of false claims is on the rise. When it comes to spotting deceptive pictures. Identification trail demosaicing operations are one example of a deep learning-based machine learning approach [1]. The secret to finding demosaicing traces in a modified picture is to look for differences in color values. An online tool capable of identifying fake images is the goal of this project, which intends to use web technologies in conjunction with a deep learning algorithm [2]. Liu, Qi, and Torr (2020) reported that 72% of the time, the web app was able to identify fake photographs by using this data.

There were 346 scholarly publications published in prestigious journals (e.g., IEEE Xplore, Elsevier, Springer, etc.) between 2017 and 2022 that focused on the analysis of fake photographs. Therefore, the majority of the academic papers that were included for the study are from that time period. With an accuracy of 99.9 percent or higher, it can differentiate between actual and fraudulent photos; nonetheless, it could be much better than the discriminator's usual Convolutional Neural Network. The study's empirical comparison of genuine and fake faces yields two significant conclusions. third point: Genuine faces are visually and aurally distinct from artificial ones. In addition, global texture statistics may use generative adversarial networks and a variety of deep learning datasets to create false faces that are less susceptible to image manipulation. [4]. In light of these results, a novel Gram-Net architecture is created that uses global texture representations to detect fake images with remarkable accuracy. [5].

In this experiment, we use Convolutional Neural Networks (CNNs) and decision trees to detect bogus photos. The classification accuracy of the Convolutional Neural Network is enhanced by its repeated training using samples from various datasets. The expansion of image-editing software has coincided with an increase in the number of claims that are susceptible. Improving pictures [6]. Identification trail demosaicing is one form of machine learning technology that has been developed for usage in web applications [1]. One approach to detecting demosaicing artifacts in the edited region of a picture is to search for shifts in color values [2]. A web-based technology and a machine learning algorithm were used to create an app that can identify publicly accessible false images [5, 7]. The internet program was able to identify counterfeit photographs with a 72% success rate when fed this data [5].

Even if a standard Convolutional Neural Network discriminator can differentiate between authentic and false photos with a 91.9% or better accuracy, it might still need some fine-tuning when it comes to precision. This study's authors identify two crucial characteristics by comparing synthetic and real faces in their tests. When comparing real and fake faces, the textural difference is striking [8]. An assortment of web-based tools, Generative Adversarial Networks, and datasets may be used to build artificial faces using global texture statistics, which are more resistant to photo alteration [8,9]. These results show that the Gram-Net architecture can use global representations of image textures to detect fake pictures accurately [1].

MATERIALS AND METHODS

The Deep Learning group of the Saveetha Institute of Health and Technology Sciences was responsible for this. You might locate it at the Saveetha School of Engineering. Twenty thousand images distributed over three sets (training, development, and testing) make up the set of unique false photos. The dataset was obtained using the open-source platform GitHub [10].

Here, we take a look at how well Convolutional Neural Networks and decision trees perform when it comes to categorization jobs. After comparing the two control groups, the Gpower algorithm determined that a sample size of forty was sufficient. To evaluate the strategy's efficacy, we split the population into two groups and compare their results [2]. Utilizing the GPower 3.1 program [11], [12] with the parameters α=0.05 and power=0.85, forty participants were divided into each group. Each of the two groups received forty samples from the total eighty that were used in the research. A Decision Tree and a Convolutional Neural Network are two machine learning methods that are implemented using Python and Jupyter.

Python was crucial in the development and implementation of the suggested work, especially the OpenCV module [13]. cited as [14]. Windows 10's UI for deep learning [1]. A quad-core Intel Core i7 CPU and four gigabytes of RAM made up the hardware.
Advanced System for Deep Learning
Convolutional neural networks (CNNs) are a sort of network architecture used by deep learning machine learning algorithms to perform tasks such as picture recognition. Picture categorization, object identification, face recognition, and countless more computer vision applications have all seen an invasion of convolutional neural networks [2].

Pseudo code

for loading the dataset() and (x_train, y_train) for testing purposes You may load the dataset using these code snippets: A collection of inputs that have been preprocessed is represented by x_train. # Make sure the data is ready to be analyzed. initialize x_test to be the same as x_test prepared for input processing. You may think of y_train as the preprocessing output. By setting y_test = preprocess output(y_test), the output data may be preprocessed.

at different levels. layers where input shape is the shape attribute. Procedures, layers, and activation="relu" on 64 separate units. An activation="softmax" model with density parameters set to the number of classes is being considered. put these settings into the program: optimizer="adam", loss="categorical_crossentropy", metrics=["accuracy"]).the result is: Both test_loss and test_acc make use of the model.rscan is assessed.x_run and y_run The parameters that comprise the fit function are (x_train and y_train), batch_size & num_epochs, validation_data and (x_test and y_test), respectively. output: "Analyze precision: ".output: format(test_acc). to act as forecasts. Assessing the x-test

Decision Tree

If you need to organize data or make predictions, a decision tree is the way to go [15]. The test scores are the branches of the decision tree, and the class names are the nodes at the very top. Flowcharts and decision trees are both hierarchical diagrams. A comparable method for trees to be "learned" involves using attribute value checks to divide the source set into subsets. Recursive partitioning is a method where the operation is repeated for each subset that arises from it. If the value of the subset at a node is equal to the target variable, then splitting does not enhance predictions in recursion [16].

Building a decision tree classifier doesn't need domain knowledge or parameter selection, making it perfect for exploratory knowledge discovery. For multi-dimensional data, decision trees are a lifesaver. Decision tree classifiers often reach very high rates of accuracy. One inductive method for gathering data is decision tree induction [17].

Pseudo code

Use GenDecTree with S and F as arguments, and set F as a property set and data set, respectively.
Sort By Parts A and B of the leaf node description So long as the stopping condition (S, F) holds, then (s)
B. createNode = send the result of root() to the node at the very bottom of the tree
The set of possible outcomes from cfroot is used as the test condition for findBestSpilt, which is (S,F) with V = v.
Test condition(s) = v & s €S, where S1 = s | root. is the first set of instructions when Parent = Tree Growth(Sv,F) and v equals €V. Part two of the instructions is B. Make a new child node a descendant of the root and give it the name v. Instruction set C is the third.

Statistical Analysis

Decision trees and CNNs are statistically analyzed using the T-test. Things that can stand on their own as variables include pictures, distance, frequency, modulation, loudness, and decibels. Course and images serve as the dependent variables. When comparing the two approaches, I used IBM SPSS 22.0.1 to do an independent sample T-test [12].

RESULTS

Utilizing Deep Fake dataset samples, the suggested Convolutional Neural Network & Naive Bayes were trained using Anaconda Navigator and Python 3.10. During the training set, The Random Forest & Convolutional Neural Network methods are fine-tuned via ten rounds. In comparison to the decision tree's 80.13% average accuracy, the Convolutional Neural Network achieved 82.11%. A sample size of forty people was used. Specifically, the p-value is 0.0293. The average accuracy values of convolutional neural networks are shown in Table 1, whereas decision trees' average accuracy values are presented in Table 2. According to Zhuo et al. (2018), the suggested Convolutional Neural Network and Decision Tree were extensively tested using Anaconda Navigator. We use statistical analysis to compare the loss values of several machine learning algorithms utilizing these data sets. Results of the statistical analysis using the ML approach are shown in Table 3 [18]. The standard deviation of decision trees is 6.27997, while that of convolutional neural networks is 6.24072. Table 4 displays the decision tree's average accuracy values. Table 3 displays the results of an Independent Sample T-test that was evaluated with the help of a Convolutional Neural Network and decision trees. Figure 1 displays the decision tree and the mean of the Convolutional Neural Network.
The mean of the standard errors is 91.6260, the standard deviation is 6.24072, while the standard deviation of a convolutional neural network is 1.97349. According to [11], these are the decision tree parameters: The average, standard deviation, and standard error mean are 88.5950, 6.27997, and 1.98590,

respectively. A standard deviation of 6.27997, an average standard error of 1.98590, and a loss average of 11.4050 were all included in the decision tree output. (1) and (18) Both ML methods have their own data sets with statistics like group values, standard deviation, mean standard error, and standard deviation. Last but not least, To make it easier to compare the two algorithms visually, we've used the loss mechanisms of convolutional neural networks and decision trees. suggests that the Convolutional Neural Network attains a much higher accuracy of 91.6260% when contrasted with the decision tree's 88.5950%.

DISCUSSION

It proves that compared to decision trees, convolutional neural networks are better, since the significance value achieved in the provided research is 0.0293 ($p<0.05$). A decision tree achieves an accuracy of 88.5950%, whereas a convolutional neural network achieves 91.6260%.

The trust that people have in the media has been steadily declining due to the spread of false information. The dissemination of real and false news on social media is quite different, according to recent studies[6]. Therefore, propagation-based detection methods have garnered a lot of attention. Using Graph Neural Networks (GNNs), these systems simultaneously learn propagation patterns[11] with user preferences. These networks rely on user nodes and chains of news to function. By understanding the connections between seemingly unrelated nodes in the network, one may more easily retrieve consumers' preferences [8]. Because they can learn efficient node representations while recognizing connections among nodes in the original network, Network Transformer Networks (GTNs) have the ability to identify bogus news [9].[5] The recommended strategy was validated by researchers by comparing it to the real-world dataset, which contains photographs taken from Twitter [18].

The issue is that adversarial attacks may easily compromise Convolutional Neural Networks. which may change the classification or legitimacy of a picture by subtle and unintentional changes. This makes it difficult to use convolutional neural networks as the only method for detecting fake pictures.

An area of potential future research is the enhancement of Convolutional Neural Networks' capability to identify false pictures via the incorporation of contextual information. Metadata and other contextual information might be used in this regard to aid in the determination of an image's legitimacy.

CONCLUSION

The accuracy rate of convolutional neural networks is 91.6260%, while that of decision trees is 88.5950%. Convolutional Neural Networks perform better than decision trees. It is accomplished by using 85% of G-power, a beta of 0.2, an interval of confidence of 95%, and an alpha of 0.05. With a p-value of just 0.0293, we can say that the two data sets are significantly different. and Convolutional Neural Networks outperform Decision tree classifiers.

One drawback of working with large datasets is the lengthier training period of Convolutional Neural Networks. Therefore, this study's ultimate objective is to enhance the system's architecture, speed up neural network training, and include a bigger image collection.

DECLARATIONS

Possible Interest Conflicts

A conflict of interest has not been identified in this paper.

The Writers' Contribution

All stages of data gathering, analysis, and paper writing were supervised by Author PRV. All the way from brainstorming to data validation and critical assessment, author VVS was there.

Praise and Honor

The Saveetha School of Engineering at what is now Saveetha University (Saveetha Institute of Medical and Technical Sciences) is tremendously appreciated for providing the funding that enabled us to carry out this study.

Because to the kind support of the following organizations, we were able to conduct this study.

Software development firm based in India Solution by Infysec

Members of this group include Hongxia Wang, Haoliang Li, and Peisong, as well as faculty members from Saveetha University and the Saveetha Institute of Medical Research and Technology. Multi-Color Space Composition of Deep Representations for Attack Detection in Phishing Images (2019). The 2019 IEEE (ICIP) Proceedings, pages 2299-2303.an engineering institution

REFERENCES

[1] A. Korkmaz and C. Hanilçi, “Image Forgery Detection Based On Parallel Convolutional Neural Networks,” in *2022 30th Signal Processing and Communications Applications Conference (SIU)*, May 2022, pp. 1–4.

[2] M. R. Bachtiar, D. N. Gusti, I. Wijaya, and M. Hidajat, “Web-Based Application Development for False Images Detection for Multi Images Through Demosaicing Detection,” in *2018 International Conference on Information Management and Technology (ICIMTech)*, Sep. 2018, pp. 277–280.

[3] N. Bhakt, P. Joshi, and P. Dhyani, “A Novel Framework for Real and Fake Smile Detection from Videos,” in *2018 Second International Conference on Electronics, Communication and*

ISSN: 2456-1134 www.isjcresm.com
Vol-10 Issue-01 Mar 2025

Aerospace Technology (ICECA), Mar. 2018, pp. 1327–1330.

[4] V. Wesselkamp, K. Rieck, D. Arp, and E. Quiring, "Misleading Deep-Fake Detection with GAN Fingerprints," in *2022 IEEE Security and Privacy Workshops (SPW)*, May 2022, pp. 59–65.

[5] Z. Liu, X. Qi, and P. H. S. Torr, "Global Texture Enhancement for Fake Face Detection in the Wild," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 8057–8066.

[6] C.-W. Tan and A. Kumar, "Integrating ocular and iris descriptors for fake iris image detection," in *2nd International Workshop on Biometrics and Forensics*, Mar. 2014, pp. 1–4.

[7] S. A. Fezza, M. Y. Ouis, B. Kaddar, W. Hamidouche, and A. Hadid, "Evaluation of Pre-Trained CNN Models for Geographic Fake Image Detection," in *2022 IEEE 24th International Workshop on Multimedia Signal Processing (MMSP)*, Sep. 2022, pp. 1–6.

[8] P. He, H. Li, and H. Wang, "Detection of Fake Images Via The Ensemble of Deep Representations from Multi Color Spaces," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, pp. 2299–2303.

[9] S. Selva Birunda, P. Nagaraj, S. Krishna Narayanan, K. Muthamil Sudar, V. Muneeswaran, and R. Ramana, "Fake Image Detection in Twitter using Flood Fill Algorithm and Deep Neural Networks," in *2022 12th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Jan. 2022, pp. 285–290.

[10] A. Jaiswal, "Real and Fake Face Detection Dataset." Feb. 2020. [Online]. Available: https://github.com/SkyThonk/real-and-fake-face-detection

[11] L. Zhuo, S. Tan, J. Zeng, and B. Lit, "Fake Colorized Image Detection with Channel-wise Convolution based Deep-learning Framework," in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, Nov. 2018, pp. 733–736.

[12] H. Kang, "Sample size determination and power analysis using the G*Power software," *J. Educ. Eval. Health Prof.*, vol. 18, p. 17, Jul. 2021.

TABLES AND FIGURES

Analyzing the Convolutional Neural Network's Accuracy and Loss (Table 1)

S.NO	Precision (in percentage form)	Percent loss
1	82.11	17.89
2	84.53	15.47
3	86.14	13.86
4	88.25	11.75

5	89.21	10.79
6	95.41	4.59
7	96.54	3.46
8	97.31	2.69
9	98.01	1.99
10	98.75	1.25

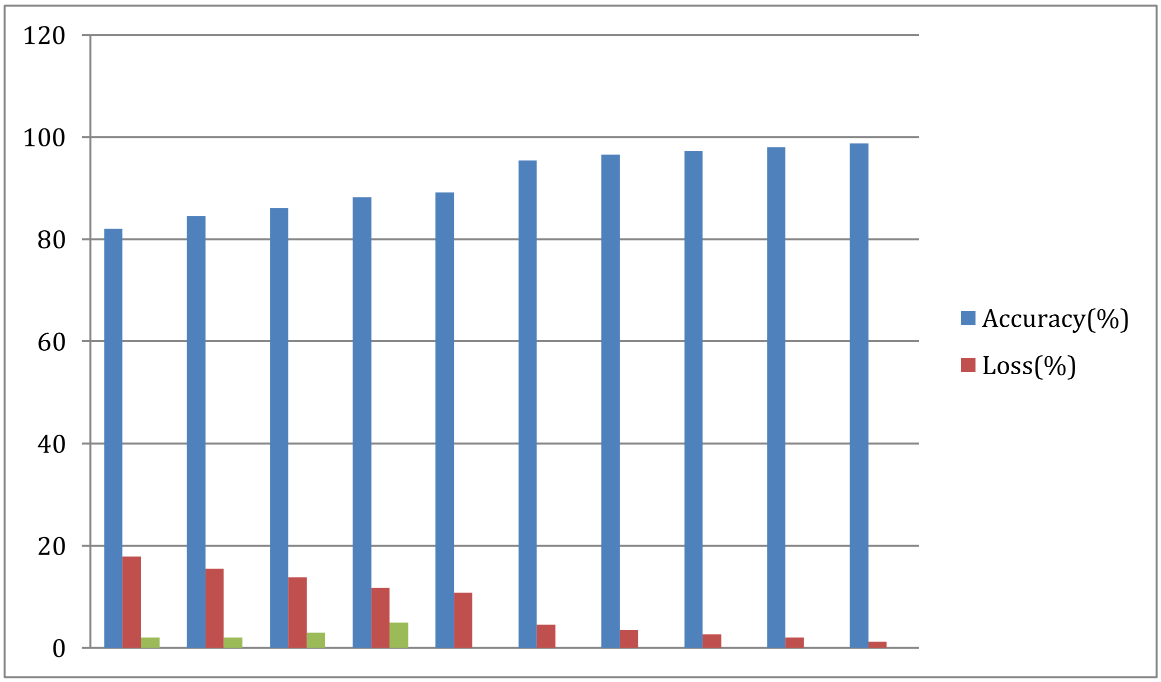

FIG.1 Convolutional Neural Network's Accuracy and Loss

Table 2: Loss Analysis and Decision Tree Accuracy

S.NO	**Precision (in percentage form)**	**Percent loss**
1	80.13	19.87

2	82.35	17.65
3	83.33	16.67
4	84.47	15.53
5	86.54	13.46
6	88.33	11.17
7	91.49	8.51
8	95.41	4.59
9	96.31	3.69
10	97.59	2.41

Joint Statistical Evaluation of Decision Trees and Convolutional Neural Networks (Table 3). The mean, standard error, and standard deviation are calculated for each of the forty samples. When comparing decision trees to Convolutional Neural Networks, the latter perform better with respect to mean loss and accuracy.

	Group	**N**	**Mean**	**Average Distinction**	**Average Error**
Accuracy	Brain Network with Convolutional Layers	40	91.62	6.24072	1.97349
	Decision tree	40	88.59	6.27997	1.98590

Loss	Convolutional Neural Network	40	8.37	6.24072	1.97349
	Decision tree	40	11.40	6.27997	1.98590

According to Table 4, the Convolutional Neural Network does not outperform decision-making structure of the Independent Sample T-test. P<0.05 with a p-value of 0.0293 and a 95% confidence range.

	A comparison of variances using Levine's test		**Finding out whether the means are identical using a 95% confidence interval T-test**						
	f	**Sig.**	**t**	**df**	**Remarkable two-tailed**	**Average variation**	**The standard deviation of errors**	**Lower**	**Upper**

Accuracy	Assumed to be equal variances	0.068	0.0798	1.083	18	0.0293	3.03100	2.79972	-2.85100	8.91300
	No assumption of equal variances made			1.083	17	0.0293	3.03100	2.79972	-2.85102	8.91302
Loss	Assumed to be equal variances	0.068	0.0798	-1.083	18	0.0293	-3.03100	2.79972	-8.91300	2.85100
	No assumption of equal variances made			-1.083	17	0.0293	-3.03100	2.79972	-8.91302	2.85102

Error Bars: 95% CI

Error Bars: +/- 2 SD

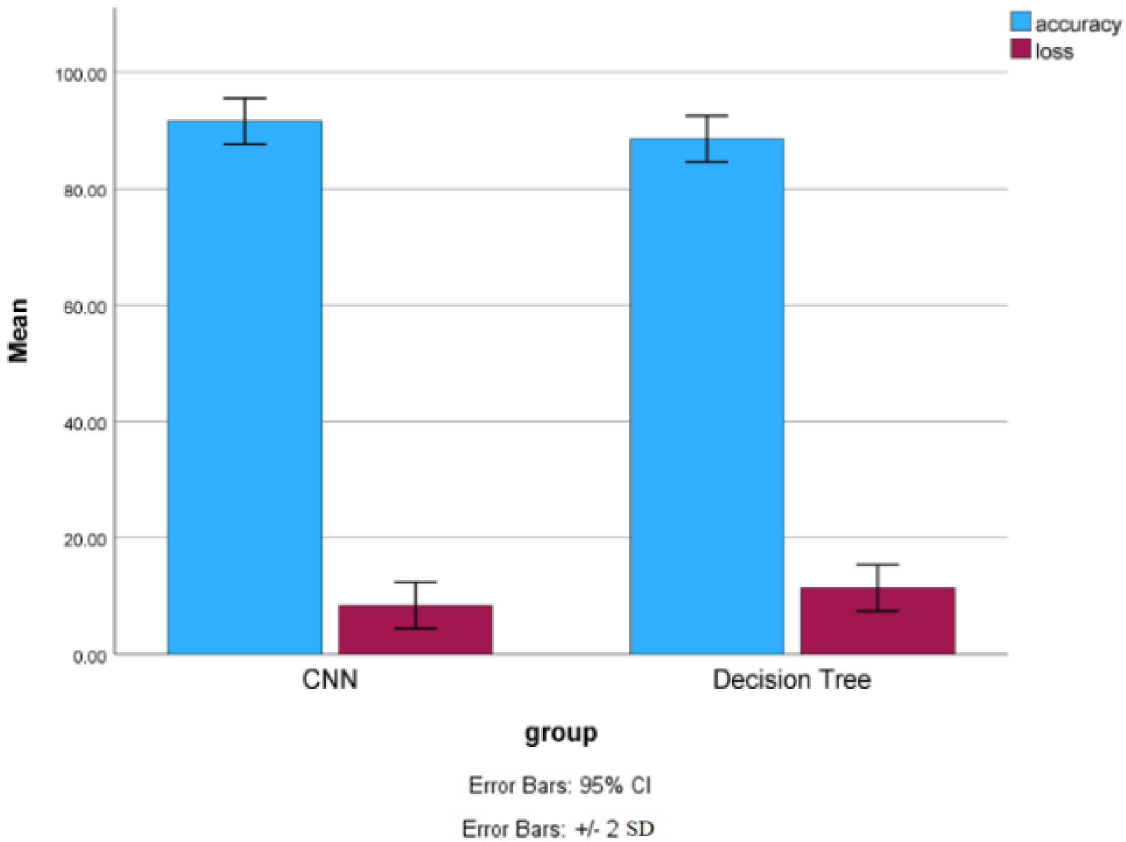

Fig 2. We compare decision trees with convolutional neural networks.. Compared to decision trees, convolutional neural networks have higher mean accuracy. Classifier: In comparison to the Decision Tree, the CNN performs better with respect to standard deviation. The X-Axis compares decision trees to convolutional neural networks. A classifier with a Y-axis showing the mean detection accuracy with a standard deviation of plus or minus two and a 95% confidence range.