Deep Threat Eye

A Vision-based Analysis Using CNN

M. Lakshmi Gayathri[1], M. Sion Kumari[2], Md. Afrin Fathima[3],

M. Venkata Yamini[4], M. Prasanna Lakshmi[5]

[1,3,4,5]Student, Department Of Computer Science And Systems Engineering, Andhra University College Of Engineering For Women, Visakhapatnam, Andhra Pradesh, India.

[2]Assistant Professor, Department Of Computer Science And Systems Engineering Andhra University College Of Engineering For Women, Visakhapatnam, Andhra Pradesh, India.

ABSTRACT: The existing static and dynamic analysis techniques largely depend upon the time-consuming manual extraction of features and are less effective on the rapidly developed malware variants. To tackle these issues, this study presents an approach for vision-based malware detection using convolutional neural networks (CNNs). In the proposed approach, binary executable files are transformed into grayscale images that CNNs can use directly to learn high-level patterns that separate malware from benign applications. The architecture mainly contains sequential convolutional layers for extracting informative visual features from the transformed images, max pooling layers to reduce the computation cost, and fully connected layers for determining whether the files are benign or malicious. The Dataset with over 200,000 samples, include benign files, and malicious files is used to train and test our model. Test results revealed an 88.64% accuracy of the CNN-based method to detect malware from benign software applications.

KEYWORDS: Malware detection, convolutional neural networks (CNNs), deep learning, vision-based analysis, grayscale images, real-time detection.

INTRODUCTION

Cyber threats are rapidly rising in today's world. It has become a big deal to detect all sorts of threats. Recent cybersecurity reports indicate that the losses caused by these malware attacks are increasing by billions annually. The old methods of detecting them aren't enough to deal with the latest threats.

To tackle these issues, Deep Learning algorithms like CNN, are used to learn high-level patterns and can learn features on their own from raw data, which saves time since you don't have to manually extract features.

This paper dives into a CNN-based approach for malware detection that transforms binary executable files into grayscale images. This method allows the model to automatically learn distinctive patterns, without any manual feature extraction. The model uses convolutional layers for feature extraction, max pooling layers to reduce dimensionality, and fully connected layers for classification. After being trained on the dataset with over 200,000 benign and malicious samples, it achieved an accuracy rate of 88.64%. By providing vision-based analysis with deep learning, this approach presents a scalable, real-time solution for cybersecurity that can keep up with new and evolving malware threats.

LITERATURE REVIEW

Signature-based and heuristic approaches for malware detection are based on pre-defined patterns and static rules, so they lag behind attacks such as polymorphic, metamorphic, and zero-day malware. These models were used successfully in several studies, Saxe and Berlin (2015)[1] and Huang and Stokes (2016) to identify known malware leveraging statistics and behavior-based analysis.

Nataraj et al. (2011) were the first to research and apply the concept of visualizing malware as images and used texture analysis on grayscale images to distinguish between malware families and achieved excellent accuracy.

In this study, Tobiyama et al. [3] (2016) and Kriegel et al. (2018) confirmed the promise of CNNs in classifying malware, using pattern recognition of images for the identification of complex byte-level structures.

Recent advancements have led to better classification accuracy in networks that use deeper architectures like ResNet, InceptionNet, and DenseNet for automatic feature extraction. Using Transfer Learning with pre-trained models, such as VGGNet or ResNet for feature extraction and fine-tuning is effective, especially when dealing with limited or unbalanced datasets. Raff et al. [5](2018) and Gibert et al. (2019) showed that Transfer learning in malware detection, leveraging knowledge from large-scale image datasets, is highly effective for malware detection.

SCOPE OF THE STUDY

The project mainly focuses on developing an intelligent and autonomous system that detects malware in computer files through Convolutional Neural Networks (CNN) and vision-based analysis.

The scope of this project is to give a smart, automated, and precise malware detection system based on deep learning methods. It provides a novel method where binary files are translated into grayscale images. These images are then processed by a CNN model to determine whether they are benign (safe) or malware (malicious). It is intended to improve cybersecurity by efficiently detecting both known and unknown malware, offering an easy-to-use web-based interface for convenient access and usage.

METHODOLOGY

This paper provides a vision-based malware detection system based on Convolutional Neural Networks (CNNs) for binary file classification as either benign or malign. The model allows the CNN to learn spatial hierarchies and patterns.

This methodology is organized into the following stages: Data Collection, Image Generation, Model Architecture, Training. Every stage guarantees precise and quick malware detection while keeping a user-friendly interface for real-time analysis.

1. DATA COLLECTION

The dataset consists of binary files from malware samples and benign software with a well-balanced class representation to mitigate bias. The files are gathered from well-known publicly available repositories that have verified malware samples.

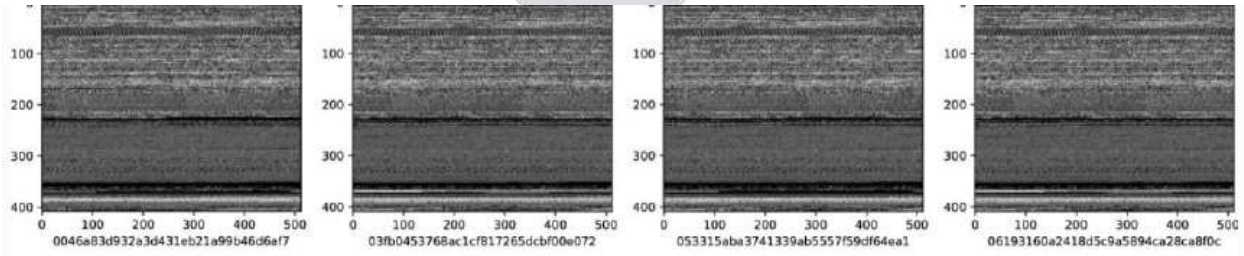
Table 1: Malware families in the dataset

Family Name	# Train Samples	Type
Ramnit	1541	Worm
Lollipop	2478	Adware
Kelihos_ver3	2942	Backdoor
Vundo	475	Trojan
Simda	42	Backdoor
Tracur	751	TrojanDownloader
Kelihos_ver1	398	Backdoor
Obfuscator.ACY	1228	Any kind of obfuscated malware
Gatak	1013	Backdoor

2. IMAGE GENERATION

The binary files are converted to grayscale images to perform feature extraction. This form of flexibility allows CNNs to capture and learn the hierarchies and patterns over space that cannot be captured using traditional feature engineering. The transformation is done with a script called generate_image.py which turns each binary file into a 2D grayscale image one at a time.

The conversion process starts with reading each binary file as a byte array. In this case, the primary structure of the binary file is retained because each byte is converted to an 8-bit grayscale pixel value. This conversion allows CNN to capture and learn the fundamental patterns related to malicious and benign files. Subsequently, the byte array is converted to a 2D NumPy array, where the elements are pixel values corresponding to the grayscale image. Images are resized to a standard size of 350 x 350 pixels to guarantee consistent input to the CNN.



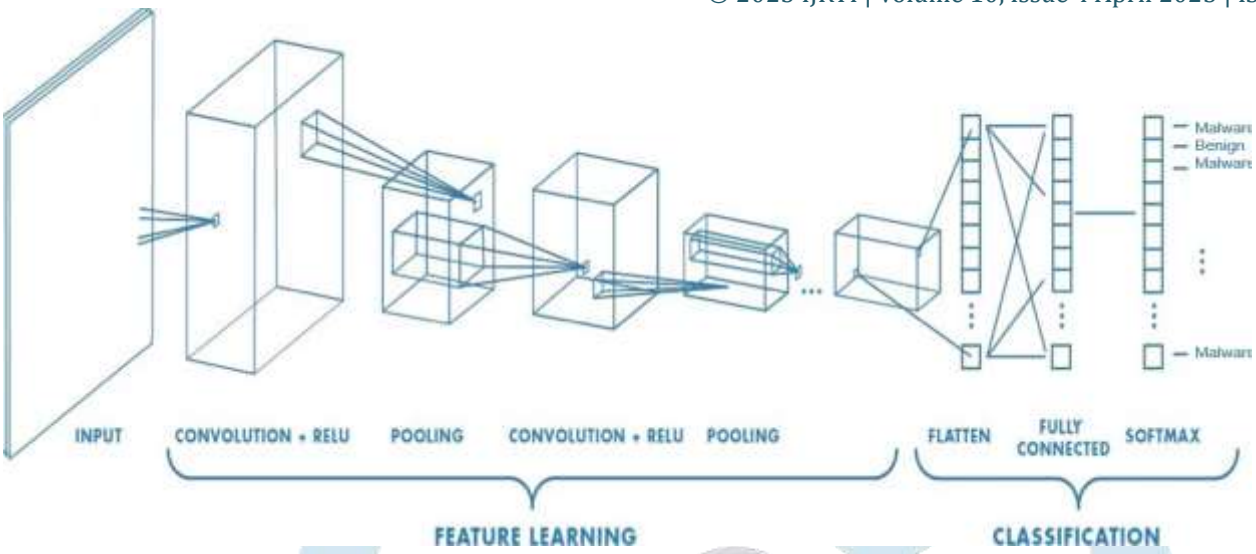
3. MODEL ARCHITECTURE

The proposed model consists of two main phases:

1. Analysis phase.
2. Classification phase.

During the analysis phase, Binary files are converted into grayscale images to allow visual representation of patterns unique to malware.

During the classification phase, a Convolutional Neural Network (CNN) is used to classify these patterns as benign or malicious files. But CNNs are particularly good for this task since they can learn hierarchical features from raw pixel data, which is perfect for image-based classification tasks.



4. TRAINING

Step 1: Data Preparation & Preprocessing - A high-quality dataset is the cornerstone of any deep learning model. We worked with a huge 117GB dataset, which has both malicious and benign files. The following categories were applied to these files, which were retrieved from practicalsecurityanalytics.com[7]:

- **Benign Files**: 86,812
- **Malicious Files**: 114,737
- **Total Samples**: 201,549

We transformed each file into an image representation to deal with raw binary files efficiently.

- A grayscale pixel (0–255) represents each byte in a file.
- By aligning these pixels into a 350x350 image, the malware file's structure and patterns are made visually identifiable.
- CNNs are able to identify distinct patterns that are typical malware families.

Step 2: Building the CNN Model - Given the data, we created a Convolutional Neural Network (CNN) that uses the extracted image representations to classify malware. Our model includes the following components:

- **Convolutional Layers** – Extracts patterns from the images.
- **Max Pooling Layer** – Makes the network more computationally efficient by reducing dimensionality.
- **Fully Connected Layers (Dense Layers)** – Decides on classification after processing the extracted features.
- **Soft-max Layer** – Both benign and malicious groups are given likelihood scores.

A dataset split of 80% training, 10% validation, and 10% testing was used to train this model. For stability, the Adam optimizer and categorical cross-entropy loss function were employed.

Step 3: Training & Optimization - It takes a lot of processing power to train deep learning models. In order to speed up processing, we used GPUs to train our model. We also conducted numerous tests to optimize hyperparameters including learning rate, batch size, and dropout rates.

- **Epochs: 50**
- **Batch Size: 64**
- **Learning Rate: 0.001**
- **Final Accuracy on Test Data: 88.64%**

Step 4: Model Evaluation - Following training, the model's efficacy in identifying malware was assessed using precision, recall, and F1-score.

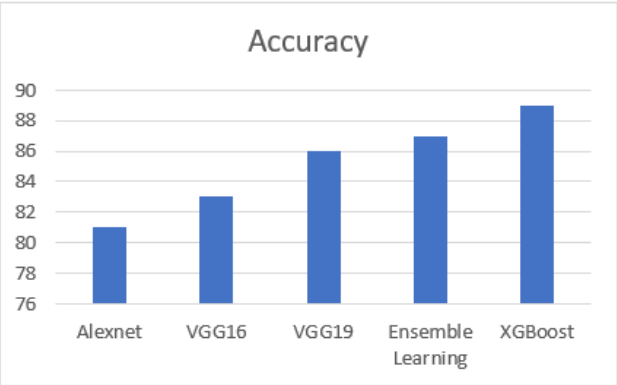
Metric	Score
Accuracy	88.64%
Precision	89.21%
Recall	87.80%
F1-Score	88.49%

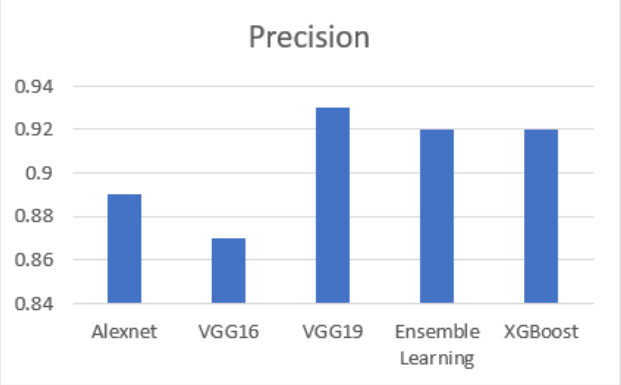
Step 5: Deployment & API Integration –

- Frontend: JavaScript, HTML, and CSS provide a straightforward and easy-to-use user interface.
- Backend: File uploads are processed by Flask and sent to the trained model for categorization.
- Model Deployment: The backend loaded the TensorFlow/Keras model, which was then utilized to make predictions.

PERFORMANCE ANALYSIS

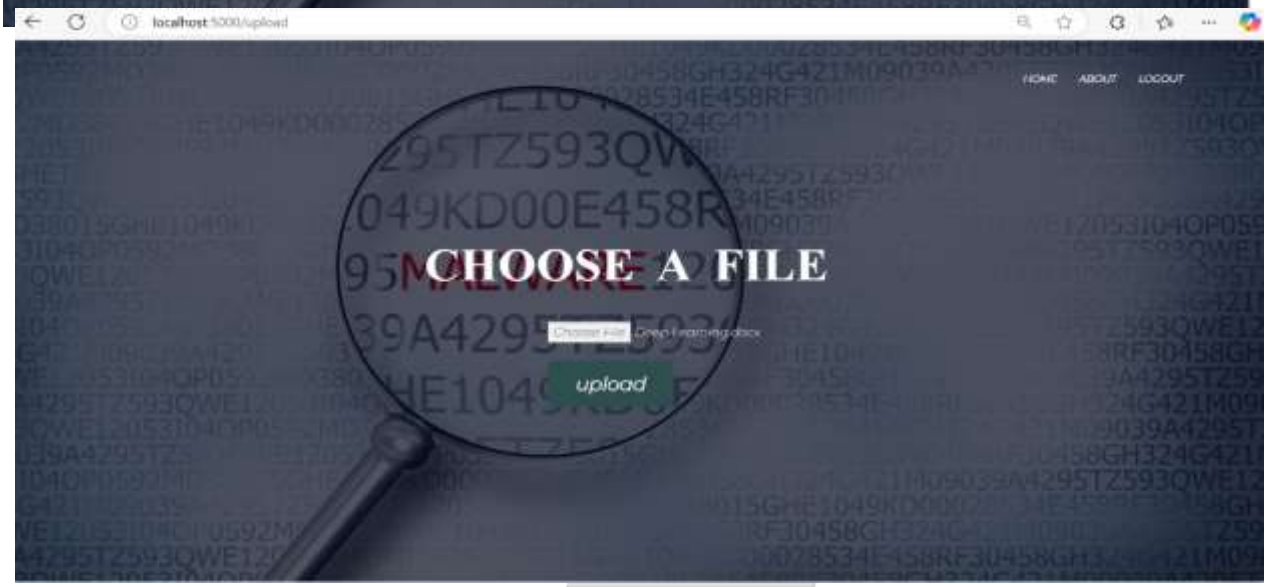
In the model's performance analysis, important metrics like accuracy, sensitivity, specificity, precision, recall, F1-score, True Positive Rate (TPR), False Positive Rate (FPR), and detection rate are assessed. These metrics aid in evaluating the model's capacity to minimize false positives and false negatives while accurately classifying malicious and benign files. The dataset is also used to compare the model's efficiency to that of other deep learning architectures, such as VGG16, VGG19, AlexNet, and Ensemble Learning. Graphical representations are used to depict the data, giving a clear comparison of different performance characteristics between models. This examination guarantees a thorough assessment of the model's effectiveness, pointing out its advantages and possible shortcomings in malware identification.





RESULTS


localhost:5000
HOME ABOUT LOGIN LOGOUT
DEEP THREAT EYE
In the face of rising cyber threats, our innovative malware detection project harnesses advanced machine learning to swiftly identify and eliminate malicious software. With real-time threat analysis, our system ensures robust protection and data integrity. Continuously adapting to emerging threats, we safeguard critical systems and maintain seamless operational continuity.
Join us in fortifying the digital landscape against malicious intrusions.
Scan for malware


localhost:5000/upload
HOME ABOUT LOGOUT
CHOOSE A FILE
upload


localhost:5000/scan
HOME ABOUT LOGOUT
RESULT
Result:
Return home

CONCLUSION

This project aimed to develop a deep learning-based malware detection system using vision-based analysis. Currently, conventional malware detection methods still rely on signature-based approaches that are inefficient in dealing with rapidly increasing threats. Modern malware variants are increasingly dependent on sophisticated evasion strategies such as code obfuscation, or polymorphism, thus defeating traditional methods of detection. In order to overcome this limitation, a Convolutional Neural Network (CNN) was used to classify malware by looking at grayscale images that were generated from binary files.

The Core idea of this methodology is that files when displayed as an image, can be represented as distinct patterns that can be recognized by a Deep Learning model. In this, binary malware files are converted into two-dimensional grayscale images and the resultant images are processed in Convolutional Neural Network (CNN) to classify the images. We trained the model on a dataset with 201,549 samples comprising benign and malicious files.

The results showed an accuracy of 88.64% and proved the feasibility of vision-based analysis for malware detection. A web application was created to make it simple for users to upload files and use the trained model for file classification. This enhances the accessibility of the system, empowering the users in real-time malware identification without requiring significant technical skills.

REFERENCES

[1] Saxe, J., & Berlin, K. (2015). "Deep neural network-based malware detection using two-dimensional binary program features" 10th International Conference on Malicious and Unwanted Software (MALWARE) (pp. 11–20).

[2] Huang, W., & Stokes, J. W. (2016) "MtNet: A multi-task neural network for dynamic malware classification" in Proceedings of the 2016 International Joint Conference on Neural Networks (IJCNN)(pp. 1951–1958).

[3] Tobiyama, S., Yamaguchi, Y., Shimada, H., Ikuse, T., & Yagi, T. (2016) in Malware detection with deep neural network using process behavior.

[4] Kriegel, M., Rossow, C., & Holz, T. (2018) "Image-based Malware Classification using VGG Networks and Transfer Learning" in Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA 2018), Lecture Notes in Computer Science, vol. 10885.

[5] Raff, E., Barker, J., Sylvester, J., Brandon, R., Catanzaro, B., & Nicholas, C. (2018) "Malware detection by eating a whole EXE" in Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence.

[6] Gibert, D., Mateu, C., & Planes, J. (2019). *The rise of machine learning for the detection and classification of malware: Research developments, trends, and challenges", Journal of Network and Computer Applications.

[7] Practical Security Analytics. (n.d.). *PE Malware Machine Learning Dataset*. Retrieved from https://practicalsecurityanalytics.com/pe-malware-machine-learning-dataset/