ISSN: 2582-7219

International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Impact Factor: 8.206

Volume 8, Issue 4, April 2025

© 2025 IJMRSET | Volume 8, Issue 4, April 2025 | DOI:10.15680/IJMRSET.2025.0804243

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Malware Detection Using Machine Learning

Mr. Sowbaran S

III-B.Sc., Department of Computer Science with Data Analytics, Dr. N.G.P. Arts and Science College, Coimbatore, India

Ms. A. Roselin

Assistant Professor, Department of Computer Science with Data Analytics, Dr. N.G.P. Arts and Science College Coimbatore, India

ABSTRACT: The project titled "Malware Detection Using Machine Learning and Binary Visualization″ aims to enhance malware detection by transforming binary files into grayscale images and using convolutional neural networks (CNNs) for classification. Traditional antivirus methods often fail to detect new or obfuscated threats, but this approach allows deep learning models to identify hidden patterns in binary data. The system includes modules for generating malware and benign binary data, training the CNN model, visualizing important features using Grad-CAM, and converting binaries into image datasets. It also features a Streamlit-based interactive dashboard for real-time file analysis and a performance viewer for metrics like accuracy, precision, and recall. This innovative solution improves detection accuracy and provides visual explanations, making it a powerful tool for cybersecurity professionals.

KEYWORDS: Malware detection, Machine learning, Binary visualization, Convolutional neural networks (CNN), Deep learning, Binary file analysis, Image-based classification, Computer-aided cybersecurity, Grad-CAM, Real-time threat detection.

I. INTRODUCTION

Malware poses a significant threat to cybersecurity, with traditional antivirus methods often struggling to detect new or obfuscated threats. early and accurate detection is essential to prevent data breaches and system compromise. this project leverages machine learning, specifically convolutional neural networks (cnns), to automate and enhance malware classification. by converting binary files into grayscale images, cnns learn hidden patterns that distinguish malware from benign software. image-based analysis improves detection accuracy and overcomes the limitations of manual or signature-based approaches. techniques like data augmentation further strengthen model performance. this ai-driven system offers real-time, consistent threat detection, reducing reliance on manual analysis and boosting overall cybersecurity efficiency. Integrating deep learning with binary visualization transforms malware detection into a smarter, faster, and more reliable process.

II. RELATED WORK

Research on enhanced Convolutional Neural Networks (CNNs) for malware detection has led to significant advancements in automated binary analysis and cybersecurity. Deep learning techniques, particularly CNNs, have been effectively applied to classify malware by converting executable binary files into grayscale image representations. This image-based approach allows the model to learn structural and behavioral patterns within malicious code that traditional signature-based methods may overlook. Transfer learning using pre-trained models such as VGG16, ResNet, and InceptionV3 has improved classification accuracy by leveraging knowledge from large-scale image datasets. Attention mechanisms, including Vision Transformers and Grad-CAM, have been integrated to enhance interpretability by highlighting the most influential regions in binary images that guided the model's decisions. To improve model generalization and robustness, preprocessing and augmentation techniques such as normalization, resizing, and grayscale intensity adjustment have been applied. Researchers have also explored hybrid architectures, combining CNNs with models like Recurrent Neural Networks (RNNs) and Capsule Networks, to boost feature extraction and detection accuracy. Additionally, Generative Adversarial Networks (GANs) have been employed to synthesize diverse malware samples, addressing challenges posed by limited or imbalanced datasets.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

III. CONVOLUTIONAL NEURAL NETWORK ALGORITHM

The proposed work employs CNN as a deep learning technique to classify OCT images to four categories. The first three categories include the mentioned diseases: DME, CNM, and Drusen. The fourth category represents the normal eye images. The adopted layers are listed below.

1. Convolutional Layer.
2. ReLU Layer.
3. Pooling Layer.
4. Flatten Layer.
5. Softmax Layer.

Convolutional Layer: A **Convolutional Layer** is the core component of a CNN that extracts important features from an image by applying small filters (kernels). These filters slide over the input, performing a **convolution operation** where they multiply pixel values and sum them up, producing a **feature map** that highlights patterns like edges, textures, and shapes. The **ReLU activation function** is commonly applied to introduce non-linearity, helping the network learn complex patterns. The layer's output size depends on parameters like filter size, stride, and padding.

ReLU Layer: A **ReLU (Rectified Linear Unit) Layer** is an activation function commonly used in CNNs to introduce non-linearity. It applies the function **f(x) = max (0, x)** to each input value, replacing all negative values with zero while keeping positive values unchanged. This helps the network learn complex patterns by preventing vanishing gradients, which can slow down training in deep networks. Unlike sigmoid or tanh activations, ReLU is computationally efficient and allows faster convergence. Variants like **Leaky ReLU** and **Parametric ReLU** address the issue of "dying ReLUs," where neurons output only zeros

Pooling Layer: A Pooling Layer is used in CNNs to reduce the spatial dimensions of feature maps while retaining important information. It helps decrease computation, prevent overfitting, and make the model more robust to small variations in the input. The most common type is Max Pooling, which selects the maximum value in a given window, preserving the strongest features. Average Pooling, on the other hand, computes the average of values within the window. Pooling layers typically use a 2×2 filter with a stride of 2, reducing the feature map size by half.

Flatten Layer: A Flatten Layer is used in CNNs to convert the multidimensional feature maps into a 1D vector, making the data compatible with fully connected (dense) layers for classification. After convolution and pooling layers extract spatial features, the flatten layer removes the spatial structure and arranges all values into a single continuous vector. This allows the model to process the extracted features using fully connected layers, which then map them to the final output classes. It acts as a bridge between the convolutional layers and the classification layers in a CNN.

Softmax Layer: A Softmax Layer is used in the output layer of a CNN for multi-class classification. It converts raw scores (logits) from the previous layer into probabilities by applying the Softmax function, which ensures that all output values sum to 1, making them interpretable as class probabilities.

IV. METHODOLOGY

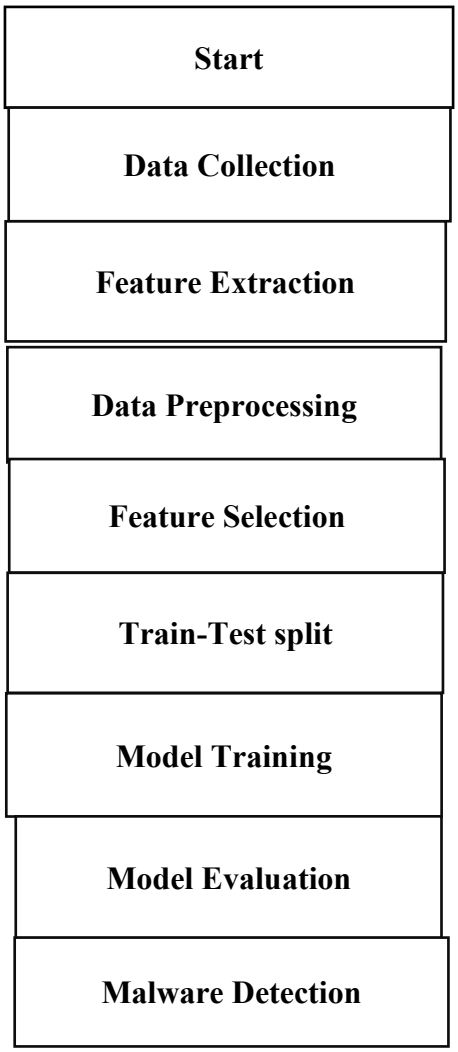
The methodology follows a deep learning pipeline that includes data preprocessing, augmentation, model development, training, and evaluation. The images are resized, normalized, and augmented before being fed into a CNN model. The CNN architecture consists of multiple convolutional layers for feature extraction, followed by fully connected layers for classification. Categorical cross-entropy loss is used for training, while the Adam optimizer is used for optimization. Optimal accuracy is attained through hyperparameter adjustment.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

V. FLOW DIAGRAM



DATASET DESCRIPTION

•**File Name**: Name of the malware or benign executable file
•**Byte Sequence**: Hexadecimal or binary content of each file
•**Image Representation**: Grayscale image created by mapping byte values (0–255) to pixels
•**Label**: Class label — *Malware* or *Benign*
•**File Size**: Size of the file, which may correlate with malware family
• **Source**: Public malware datasets such as Malimg Dataset, VirusShare, and Kaggle MalwareImage dataset

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VI. RESULT AND DISCUSSION

Pseudo code :

Start the program

Import required libraries and custom modules

Function: load_model
  Load the saved CNN model weights
  Initialize the CNN model
  Load weights into the model
  Return the model

Function: preprocess_file(file, type)
  If file is image:
    Convert to grayscale and resize
    Convert to tensor
  Else if file is binary:
    Convert binary data to grayscale image
    Resize and convert to tensor
  Else if file is CSV:
    Read CSV as matrix
    Handle invalid or missing values
    Convert to grayscale image and resize
  Return preprocessed tensor and original image

Function: save_predictions_to_csv(file_path, predictions)
  Save prediction results in CSV format

Function: main (Streamlit app)
  Show app title
  Show sidebar with options: "Malware Detection" or "Show Statistics"

  If "Malware Detection" selected:
    Load the model
    Upload file
    Preprocess the file
    Make prediction using model
    Show prediction result and confidence
    Display bar chart for confidence
    Generate and show Grad-CAM heatmap
    Save prediction to CSV

  If "Show Statistics" selected:
    Input CSV file path
    Load and show statistics from the file

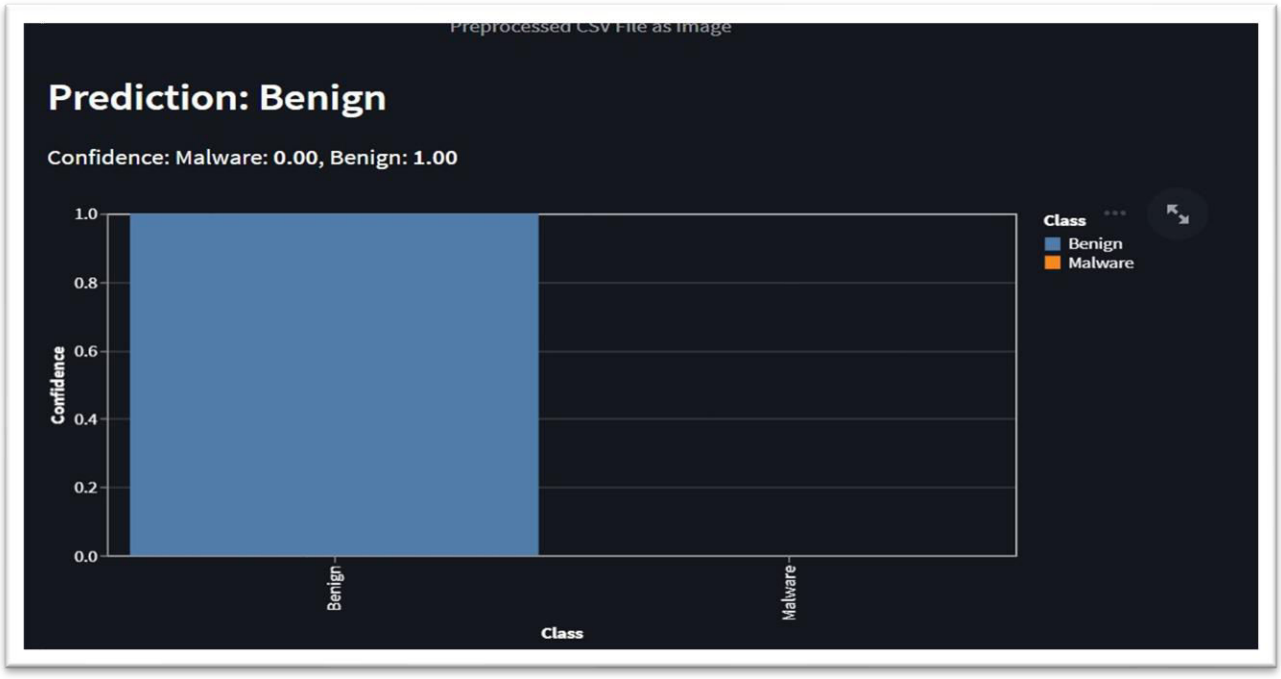
Call main function

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

VII. RESULTS

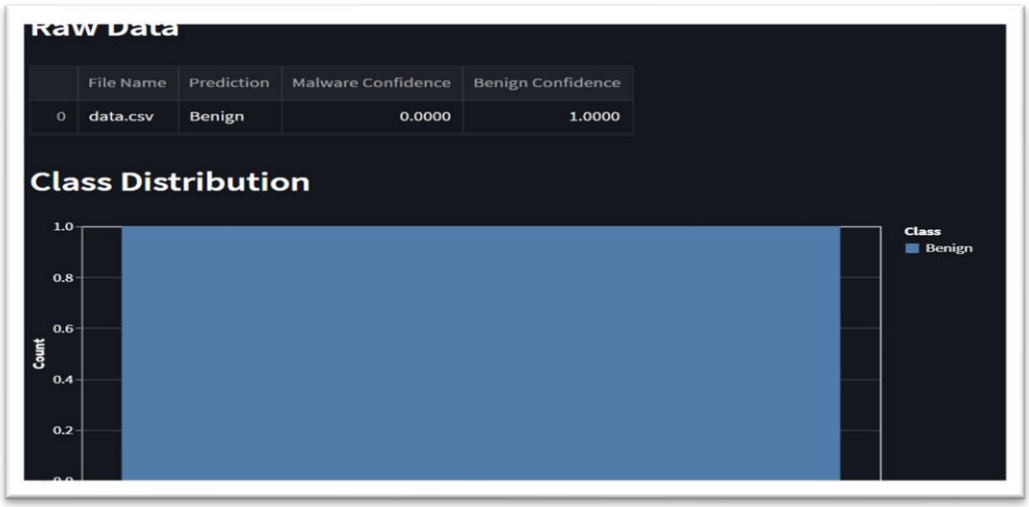


© 2025 IJMRSET | Volume 8, Issue 4, April 2025 | DOI:10.15680/IJMRSET.2025.0804243

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)



VIII. CONCLUSION & FUTURE-ENHANCEMENT

This project demonstrates the effectiveness of using CNNs for automated eye disease classification. By leveraging deep learning with TensorFlow and Keras, the system accurately identifies cataracts, diabetic retinopathy, glaucoma, and normal cases. The use of a React.js frontend, Flask backend, and cloud deployment via AWS and Docker ensures a smooth, scalable, and accessible user experience.

Future enhancements include:

- Expanding the dataset for better accuracy and generalization.
- Adding more eye disease categories.
- Exploring advanced models like Vision Transformers and hybrid CNN-RNNs.
- Improving scalability with cloud auto-scaling.
- Speeding up predictions using TensorRT or ONNX Runtime.

REFERENCES

[1] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.
[2] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).
[3] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
[4] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer international publishing, 2015.
[5] Gulshan, Varun, et al. "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs." *jama* 316.22 (2016): 2402-2410.
[6]Ting, D. S., Cheung, C. Y., Lim, G., et al. (2017). Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations. JAMA, 318(22), 2211-2223.
[7] Esteva, A., Kuprel, B., Novoa, R. A., et al. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115-118.
[8] Rajalakshmi, R., Subashini, R., Anjana, R. M., & Mohan, V. (2018). Automated diabetic retinopathy detection in smartphone-based fundus photography using artificial intelligence. Eye, 32(6), 1138-1144.
[9] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems, 1097- 1105. Howard, J., & Gugger, S. (2020). Fastai: A layered API for deep learning. Information, 11(2), 108.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com