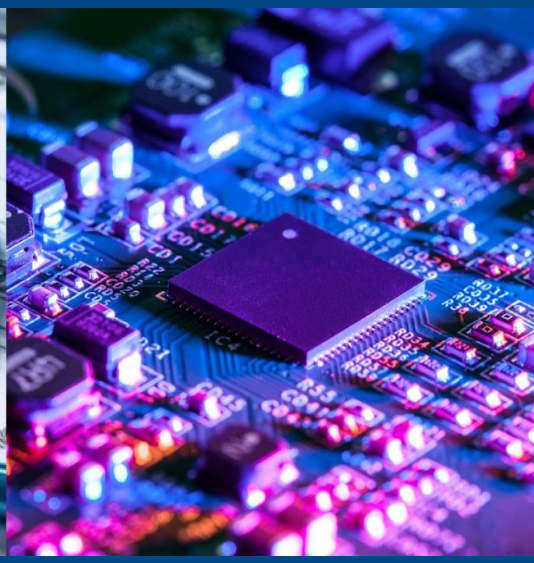
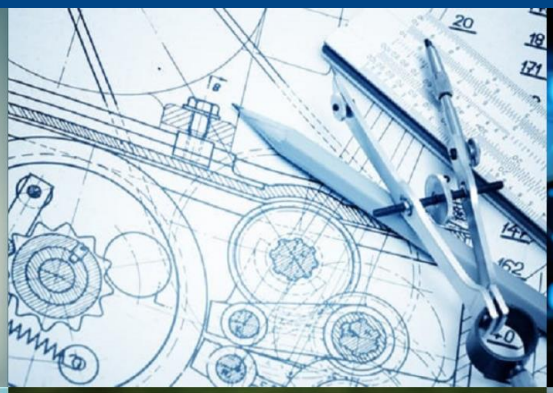
ISSN: 2582-7219

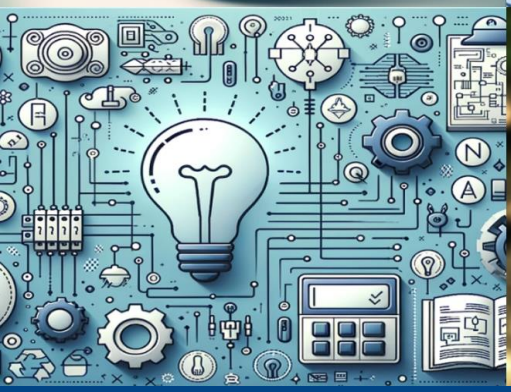




International Journal of Multidisciplinary Research in Science, Engineering and Technology

(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.2025.0804091

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)

Image Based Malware Classification Using Transfer Learning Techniques

Shaik Riyazuddien[1], Sandu Srihari[2], Syed Haasin Sameer[3], Thallam Abhishek[4], Pokuri Mahesh[5]

Associate Professor, Department of Electronics & Communication Engineering, Vasireddy Venkatadri Institute of Technology, Guntur, India[1]

Department of Electronics & Communication Engineering, Vasireddy Venkatadri Institute of Technology, Guntur, India[2,3,4,5]

ABSTRACT: Malware still presents a serious threat to cybersecurity as attackers are using more advanced evasion strategies. Conventional malware detection techniques, such heuristic and signature-based approaches, find it difficult to keep up with newly developed and hidden malware variants. This work uses transfer learning techniques to offer a novel image-based malware classification method that addresses these issues. This method converts malware programs into grayscale pictures, with each pixel standing for a byte value. It is possible to identify distinctive structural patterns that are exclusive to certain malware families thanks to this visual depiction. The approach reduces the need for considerable human feature engineering while achieving excellent classification accuracy by utilizing pre-trained convolutional neural systems (CNNs), like XceptionNet, and deep learning. Additionally, the suggested methodology is incorporated into an online platform that offers a user-friendly malware analysis interface. By enabling users to upload samples of malware and obtain real-time categorization findings, the technology expedites the detection process. Results from experiments show how transfer learning may improve malware classification precision and resilience to changing threats. This study demonstrates the promise of machine learning in cybersecurity by providing a scalable and effective method for malware detection. The suggested solution helps to bolster defences against new cyberthreats by utilizing the capabilities of transfer learning and image-based analysis.

KEYWORDS: Malware classification, Xception, transfer learning, Deep Learning.

I. INTRODUCTION

Malware, which stands for malicious software, is a persistent and constantly changing cybersecurity issue that affects people, companies, and governmental organizations all over the world. Malware is a tool used by cybercriminals to breach networks, steal confidential information, interfere with daily operations, and initiate extensive cyberattacks. Conventional malware detection algorithms mostly use heuristic and signature-based approaches, which work well for known malware samples but frequently have trouble detecting novel and disguised variations. Conventional detection methods are becoming less effective as malware writers continue to use complex evasion techniques including encryption, polymorphism, and metamorphism. Researchers and cybersecurity experts are currently investigating cutting-edge methods that use deep learning and artificial intelligence (AI) to classify malware more effectively and adaptively in order to overcome these difficulties.

Image-based malware classification, which converts malware binaries into pictures and uses deep learning algorithms for classification, has become a promising method in recent years. The discovery that various virus families display distinct structural and textural features when represented as pictures serves as the foundation for this method. Deep learning models, in particular Convolutional Neural Networks (CNNs), can evaluate and extract complex information that could be difficult to discover through conventional static or dynamic analysis by turning files containing malware into grayscale or RGB pictures. Malware categorization is now more precise, scalable, and effective thanks to the switch from traditional feature engineering to automate feature extraction.

By using pre-trained deep learning models that have been trained on extensive datasets, transfer learning has significantly increased the potential of based on pictures malware classification. Transfer learning enables the adaption of well-known architectures like XceptionNet, VGG16, which or ResNet that have already been pre-trained on

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)

enormous picture datasets, as opposed to building a CNN from scratch. By fine-tuning these models using malware picture datasets, classification accuracy is increased, training speed is accelerated, and computing costs are greatly decreased. The problem of data scarcity is also addressed by using pre-trained models, as malware datasets frequently include limited labelled samples or class imbalances.

The ability of image-based malware categorization to withstand common evasion tactics used by hackers is another benefit. Deep learning models trained on picture representations are better at generalizing and can identify malware even if its binary code has been slightly changed, in contrast to traditional detection using signatures, which depends on preset patterns. CNNs can also detect latent patterns in samples of malware that may be hard for traditional rule-based systems to detect since they learn hierarchical characteristics from picture data. Because of its resilience, image-based categorization may be used to improve malware detection in practical cybersecurity applications.

II. LITERATURE REVIEW

Computer programs (code, scripts, and software) intended to impede computer operations or gather data are referred to as malware [1]. Malware results in aggressive actions, unauthorized access to system resources, and privacy loss. 360,000 new malicious files were discovered per day by Kaspersky Lab in 2017 [2]. Given the incentive for malware writers to create such programs for illicit financial benefit, this alarmingly high number is probably going to rise in the future.

For many years, traditional antivirus software has had the largest market share in the security industry and is often quite successful. Nonetheless, the capacity of malware writers to create new strategies and technologies has increased dramatically during decades of development [3]. According to research, conventional antivirus software has seldom ever been able to identify new malware in the last ten years.

These days, the majority of antivirus software providers have employed machine learning to address this issue and have implemented these methods across the industry, such as by utilizing statistical features like N-grams [5] or API calls [4]. For malware analysis and the extraction of features, many of them rely on extensive domain expertise, which can be time-consuming.

Deep convolutional neural networks (CNN) built around the VGG16 architecture have been investigated recently for malware classification [6]. CNN was given the task of classifying images in order to tackle the malware classification challenge. When it comes to picture categorization, the most often used CNN model is the VGG16 model.

In recent years, there is a tendency for models to perform better the deeper they are. The VGG16 model, which had a depth of 23, achieved a top-1 classification error of 28.5% in the 2014 ImageNet competition. The top-1 classification error for the Residual Network (ResNet) model with depth 168 in 2015 was 24.1%. The top-1 classification error for the Inception V3 model with depth 159 in 2016 was 21.8%. Last but not least, the 126-layer Xception model produced a top-1 classification mistake of 21.0% in 2017 [7].

But in the literature, Xception was never employed to address the malware categorization issue.

In this work, we presented a unique approach to malware family classification using the Xception model and deep CNN. The Xception model utilized in this study is an ImageNet pre-trained model that was made available on Keras. Thus, the pre-trained Xception model has been applied to the malware classification problem using the well-liked transfer learning technique, which enhances learning in a new task by transferring information from a similar task that has already been learnt.

The results of the experiments show that, in comparison to the previous victor of the VGG model on malware image classification, the validation accuracy of the model developed by Xception on the Malimg Dataset [8] and the Microsoft malware datasets [9] is greater (99.04% and 99.17%, respectively) with lower log loss. Furthermore, the Global Average Pooling layer in the suggested approach minimizes overfitting by lowering the overall number of parameters in the Xception model, which lowers the likelihood of overfitting when compared to the VGG model [10].

© 2025 IJMRSET | Volume 8, Issue 4, April 2025| DOI: 10.15680/IJMRSET.2025.0804091

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.SYSTEM ANALYSIS

More efficient and sophisticated detection systems have to be developed in light of the growing sophistication of cyber threats and the quick spread of malware. Novel and obfuscated malware versions have proven difficult to identify using conventional malware detection techniques including signature-based and heuristic-based approaches [11]. A more flexible and resilient strategy is needed to combat new threats as cyber criminals are always changing their strategies. A potential remedy is deep learning-based picture classification, which uses neural networks to visually evaluate malware samples and identify hitherto undiscovered malware families [12]. By using transfer learning to prepared Convolutional Neural Networks (CNNs), this study seeks to improve malware categorization and offer a scalable and effective malware detection system as shown in Figure 1.

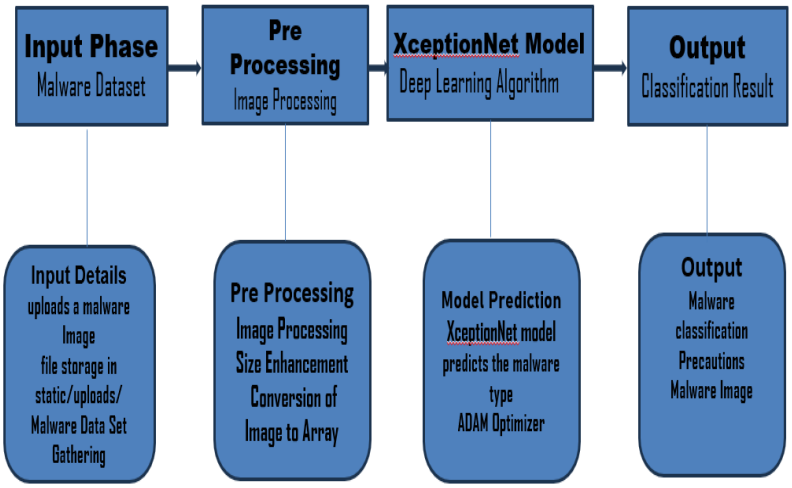


Figure 1: Model architecture

A.EXISTING SYSTEM

The majority of current malware detection methods depend on static and dynamic analysis methods. Static analysis is effective yet useless against opaque or polymorphic malware since it looks at a malware sample's code, file structure, and signatures without actually running the virus [13]. On the other hand, dynamic analysis entails executing malware in a sandbox — a controlled environment—in order to watch how it behaves. Dynamic analysis works well for identifying sophisticated threats, but it requires a lot of resources and is vulnerable to evasion strategies like malware-enabled anti-analysis tools [14]. Furthermore, rule-based methods have trouble generalizing and frequently need to be updated in order to identify newly discovered malware strains. Because of these drawbacks, researchers are now looking at several strategies that use deep learning to automatically classify malware [15].

B.PROPOSED SYSTEM

By converting malware binaries into RGB or grayscale pictures and using transfer learning on CNN architectures that have already been trained for classification, the suggested system presents an image-based method for malware classification. By using deep learning to extract hierarchical characteristics from malware photos, this technique makes it possible to identify intricate patterns that conventional approaches would overlook. The approach increases classification accuracy while reducing the requirement for substantial processing resources and sizable training datasets by employing transfer learning. Using pre-trained models like ResNet, VGG16, or XceptionNet improves detection effectiveness and makes it possible to quickly adjust to new malware strains.

A website-based interface that enables users to upload suspicious malware files for categorization is another feature of the suggested system. After processing the data and turning them into pictures, the system predicts the categorization of the photos using a deep learning model. The online application offers a useful solution for cybersecurity experts and companies by facilitating automated threat analysis and providing real-time data. This

© 2025 IJMRSET | Volume 8, Issue 4, April 2025| DOI: 10.15680/IJMRSET.2025.0804091

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)

method improves malware identification, increases its resistance to evasion strategies, and increases overall cybersecurity resilience by incorporating deep learning-based picture categorization into malware detection workflows.

C . DATASET

This project uses the Malimg Dataset, which was obtained from Kaggle and includes 9,339 grayscale pictures that were extracted from malware binary files. With 25 different malware families represented by these photos, it's a large dataset for malware classification using deep learning. To provide a wide representation of real-world threats, the dataset is divided into several malware kinds, such as worms, Trojan horses, and adware. With 8,405 photos used for learning and 934 images for testing, the dataset has a 90:10 train-test split sample is shown in Figure 2. By classifying each picture according to the virus family it belongs to, supervised learning techniques are made possible. The labelling procedure guarantees that the algorithm using deep learning can efficiently differentiate between various malware kinds and discover significant patterns from them.

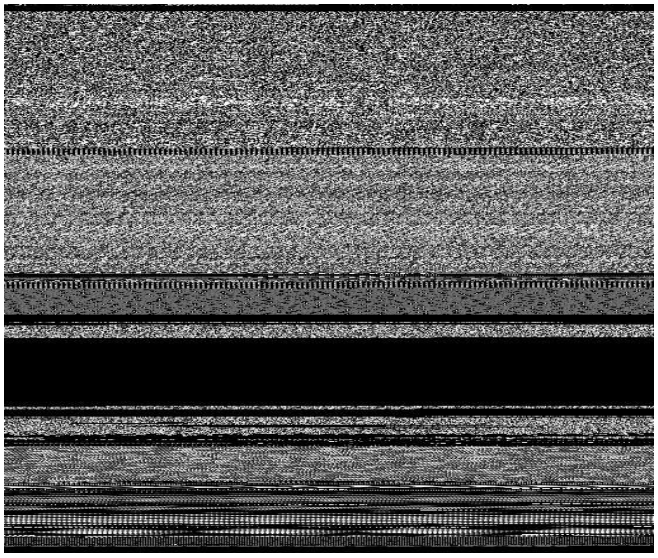


Figure 2: Sample Image from Malimg Dataset.

A number of preparation techniques, including as thresholding, normalizing, and picture scaling, are used to get the dataset ready for deep learning. By normalizing input data and enhancing contrast between various malware samples, these procedures improve the model's feature recognition capabilities. The technique is able to uncover spatial patterns that distinguish distinct malware families by using grayscale photos rather than raw binary data.

D . PROPOSED METHODOLOGY

1. **Data Acquisition:** Data collection is the initial stage of the suggested technique, during which pertinent data is gathered for the malware classification model's training and testing. The Malimg Dataset, which comprises 9,339 grayscale photos representing different malware families, was used in this experiment. By transforming the hexadecimal representations of malware binary files into pixel values, these pictures are produced, enabling visual pattern analysis by deep learning models.

   The dataset was acquired via the popular machine learning dataset portal Kaggle. The dataset contains 25 different malware families, encompassing different kinds like Adware, Trojan horses, and worms, to guarantee a thorough classification effort. 90% of the dataset's 8,405 photos are used for instruction, while the remaining 10%, or 934 images, are utilized for testing. By avoiding overfitting and enhancing generalization, this division guarantees that the model is trained efficiently while maintaining a distinct dataset for assessment.

2. **Data Preprocessing:** A vital step in enhancing the caliber of input data prior to deep learning model training is data preparation. Raw virus photos can have a detrimental effect on model performance since they frequently

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)

contain extraneous noise, irregular sizes, and different brightness levels. This is addressed by applying a number of preprocessing approaches. By guaranteeing that every image has the same proportions, image scaling makes them compatible with the deep learning architecture. Normalization enhances model convergence during training by scaling values of pixels between 0 and 1.

To improve feature extraction, thresholding techniques are also applied to the malware pictures to increase contrast and draw attention to significant patterns. The malware family names are transformed into numerical representations by label encoding, which enables the deep learning model to efficiently handle categorical input. These preprocessing procedures guarantee that the model extracts useful characteristics from the malware pictures, lower noise, and enhance the quality of the input data.

3. **Model Selection and Architecture:** XceptionNet, a potent convolutional neural network, or CNN, that employs separable convolutions for effective feature extraction, is the deep learning model used for this project. Because of its capacity to attain high accuracy while preserving computing economy, XceptionNet was chosen. The architecture of the model is perfect for classification jobs since it can recognize intricate patterns in malware pictures. Using the Malimg Dataset, a pre-trained XceptionNet model is refined through the application of transfer learning. This shortens training time and boosts performance by enabling the model to use previously learnt features.

   The model's ability to differentiate between the 25 malware families is ensured by adapting XceptionNet's final layers to the multi-class classification challenge. The model's fully connected layers categorize malware pictures using learnt representations after many convolutional layers extract hierarchical characteristics. By using depthwise separable convolutions, the model becomes quicker and more effective by using fewer parameters.

4. **Model Training and Optimization:** The training procedure starts with the training set (8,405 photos) when the model architecture is established. The Categorical Cross- entropy loss function, which works well for multi-class classification problems, is used to train the model. Because of its flexible learning rate capabilities, which guarantee quicker convergence and superior generalization, the Adam optimizer was used. To strike a compromise between training stability and computational efficiency, an appropriate batch size is chosen. Image rotation, tumbling, and zooming are examples of data augmentation techniques that are used to artificially expand dataset size and enhance model resilience. During several training epochs, the model gains the ability to recognize patterns and correctly categorize malware pictures.

   Performance parameters including accuracy, precision, recall, and F1-score are tracked during the training process to evaluate model advancements. Techniques like early halting and dropout layers are used to avoid overfitting. The model's capacity to effectively identify and classify malware is enhanced as it continually learns to distinguish between distinct malware families based on visual patterns.

5. **Model Evaluation and Performance Metrics:** Following training, the model's performance in the actual world is assessed using the testing set, which consists of 934 photos. Evaluation is done using a number of important measures. The majority of malware photos are accurately categorized thanks to accuracy, which gauges how accurate the model's predictions are overall. In order to reduce false positives, precision measures the proportion of anticipated malware kinds that are really accurate. Recall evaluates the model's ability to identify all pertinent malware occurrences, guaranteeing that no dangerous threats are missed. The F1-score offers a thorough evaluation statistic by striking a balance between accuracy and recall. The efficacy of the trained XceptionNet model in differentiating distinct malware families is demonstrated by its remarkable accuracy of around 98.5% on test data.

   The model's excellent performance suggests that it may be included into cybersecurity systems to classify malware. Adding more datasets, testing with other deep learning architectures, and implementing the model for real-time malware detection are possible future developments.

E.ALGORITHMS USED:

XceptionNet Architecture: XceptionNet (Extreme Inception) is a deep convolutional neural network designed to improve efficiency and accuracy in image classification tasks. It builds upon the Inception architecture but replaces its standard convolutions with depthwise separable convolutions, significantly reducing the number of parameters and computational complexity while improving performance.

The network is divided into three main flows:

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)

1. Entry Flow – Extracts low-level features from the input image.
2. Middle Flow – Enhances and refines these features through repeated transformations.
3. Exit Flow – Final feature extraction and classification.

The architecture, as seen in the diagram, is designed to maximize learning efficiency while reducing redundancy in convolutional operations. Below is a detailed breakdown of each stage.

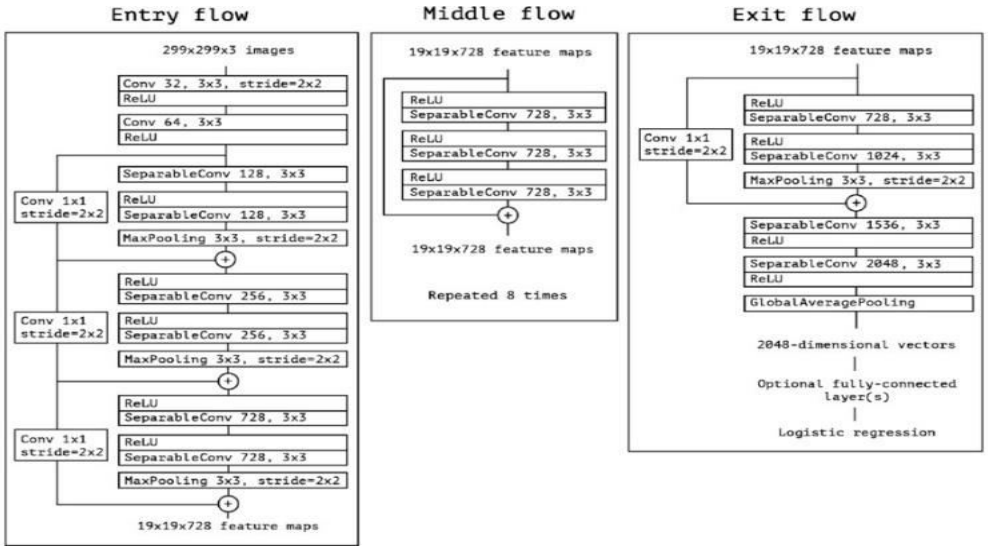


Figure 3: Detailed Architecture of XceptionNet.

1. Entry Flow – Feature Extraction:

The Entry Flow is responsible for extracting low-level features from the input image, progressively reducing its spatial dimensions while increasing its depth. This helps in identifying fundamental patterns like edges and textures.

Step-by-Step Breakdown:

1. Input Layer:
   - The model takes a 299×299×3 RGB image as input.
   - This ensures that the image has enough resolution for learning meaningful features.
2. First Convolutional Layers:
   - Conv (32 filters, 3×3, stride=2) → Extracts the initial image patterns.
   - Conv (64 filters, 3×3) → Further enhances feature extraction.
   - These layers use standard convolutions with ReLU activation.
3. Transition to Depthwise Separable Convolutions:
   - From this point, the model replaces standard convolutions with separable convolutions to reduce computational cost.
   - SeparableConv (128 filters, 3×3) → ReLU
   - SeparableConv (128 filters, 3×3) → ReLU
4. Downsampling and Expansion:
   - The network uses a 1×1 convolution (stride=2) followed by max pooling (3×3, stride=2) to reduce spatial dimensions while increasing depth.
   - This process is repeated, with feature depth increasing at each step:
     - 128 → 256 → 728 filters
   - These deeper layers allow the network to capture complex structures and patterns in the image.

2. Middle Flow – Deep Feature Learning:

The Middle Flow is where the network performs intensive feature extraction. It captures high-level patterns and relationships in the data by applying repeated separable convolutions.

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)

Step-by-Step Breakdown:

1. The feature maps from the Entry Flow (size: 19×19×728) are passed into this phase.
2. The Middle Flow consists of 8 identical blocks, each applying:
   - ReLU activation
   - Separable convolution (728 filters, 3×3)
   - Separable convolution (728 filters, 3×3)
   - Separable convolution (728 filters, 3×3)
3. The output feature map remains the same size (19×19×728) throughout these 8 blocks.
4. Skip connections are used to enhance gradient flow, preventing vanishing gradients in deep networks.

3. Exit Flow – Classification and Output:

The Exit Flow performs the final feature extraction and classification. It condenses the learned representations into a format suitable for decision-making.

Step-by-Step Breakdown:

1. The feature maps from the Middle Flow are passed into the Exit Flow (still 19×19×728).
2. A final separable convolution (728 filters, 3×3) → ReLU is applied.
3. Another 1×1 convolution (stride=2) is used to downsample the feature map.
4. A series of depthwise separable convolutions extract the final feature representations:
   - SeparableConv (1024 filters, 3×3)
   - SeparableConv (1536 filters, 3×3)
   - SeparableConv (2048 filters, 3×3) → ReLU
5. Global Average Pooling (GAP) Layer:
   - Converts the 2048-dimensional feature maps into a 1D feature vector.
   - This reduces overfitting by ensuring that spatial details are averaged instead of densely connected layers.
6. Fully Connected Layer (Optional):
   - If needed, additional fully connected layers can be added before classification.
7. Softmax/Logistic Regression Output Layer:
   - The final layer classifies the image into one of the 25 malware families using softmax activation.

XceptionNet is a powerful and efficient deep learning model that enhances malware classification by utilizing depthwise separable convolutions. The three-phase structure (Entry Flow, Middle Flow, and Exit Flow) allows the network to progressively extract, refine, and classify malware images. By leveraging transfer learning and fine-tuning the model on a specialized malware dataset, XceptionNet achieves high classification accuracy, making it well-suited for cybersecurity applications.

Transfer Learning: Transfer learning is a deep learning approach that uses knowledge from a big dataset to increase efficiency and accuracy by adapting a pre-trained model for a new job. This study uses a dataset of 9,339 grayscale photos representing 25 malware families to refine the XceptionNet model, which was first trained on ImageNet, for malware classification. Even with a small dataset, the model's learnt features are reused rather than being trained from scratch, enabling faster convergence and improved performance. This method drastically cuts down on training time and computational expenses while improving the model's capacity to identify malware patterns.

IV.IMPLEMENTATION

System Setup and Model Loading: Setting up the environment and loading the deep learning model that has been built is the first stage in the implementation process. The web application, which functions as the interface for malware picture categorization, is developed using the Flask framework. XceptionNet, a deep learning architecture trained on the Malimg Dataset, is the model utilized to categorize malware photos into 25 distinct families. TensorFlow's load_model() method is used to load the pre-trained model, which is saved as a.h5 file. To guarantee that it can process photos and produce predictions, this model is assembled with the required settings. In order to manage image preparation, deep learning processes, and web interactions, necessary libraries like OpenCV, NumPy, TensorFlow, Keras, and Flask are also imported.

© 2025 IJMRSET | Volume 8, Issue 4, April 2025| DOI: 10.15680/IJMRSET.2025.0804091

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)

Image Upload and Preprocessing: Users must upload an image file using the Flask web application in order to categorize a malware picture. The program specifies an upload folder in which the pictures are momentarily kept. Secure_filename() is used to sanitize the uploaded file's name in order to guarantee security and appropriate file processing. The only acceptable picture formats are PNG, JPG, and JPEG. The image is preprocessed after upload and then entered into the model. The image is resized to 75x75 pixels, converted to an array format, and normalized using tf.keras.applications.xception.preprocess_input() as part of the preparation processes. By taking these precautions, the model is guaranteed to receive input data in a consistent format, increasing accuracy and lowering noise.

Malware Classification Using Deep Learning: Following preprocessing, the model forecasts the uploaded image's malware class. The image path is sent to the predict_image_class() function, which loads the image, transforms it into an array, and enlarges its size to match the required input format of the model. Following image processing, the model generates likelihood ratings for every malware family. The anticipated malware family is determined by selecting the class index with the greatest probability using np.argmax(). Users can view the malware name found in their uploaded file by the system mapping the index to the appropriate virus kind from a prepared list.

Web Interface and Result Display: The categorization system's user-friendly interface is made with Flask. To handle various pages, including the homepage, about, offerings, and contact, many routes are developed. The /resultp route, where visitors submit an image for categorization, contains the primary functionality. When a prediction is successful, the program displays the submitted image, the malware family that was found, and the forecast's confidence level on the resultp.html page. An error notice directing the user to submit a valid image file is shown by the system if the picture format is wrong. To give real-time feedback on effective or unsuccessful procedures, flash messages are utilized.

Deployment and API Optimization: The Flask application must be deployed and optimized for practical use as the last stage. In debug mode, the program operates on a Flask server, making testing and changes simple. Using @after_request, caching techniques are turned off for API endpoints to increase speed and guarantee that new data is handled at all times. Furthermore, the application's efficient and lightweight architecture enables it to process numerous picture categorization requests with negligible delay. For increased accuracy and improved classification performance, future improvements can involve increasing the malware collection, enhancing the user interface, and incorporating a cloud-based deployment.

V.RESULTS

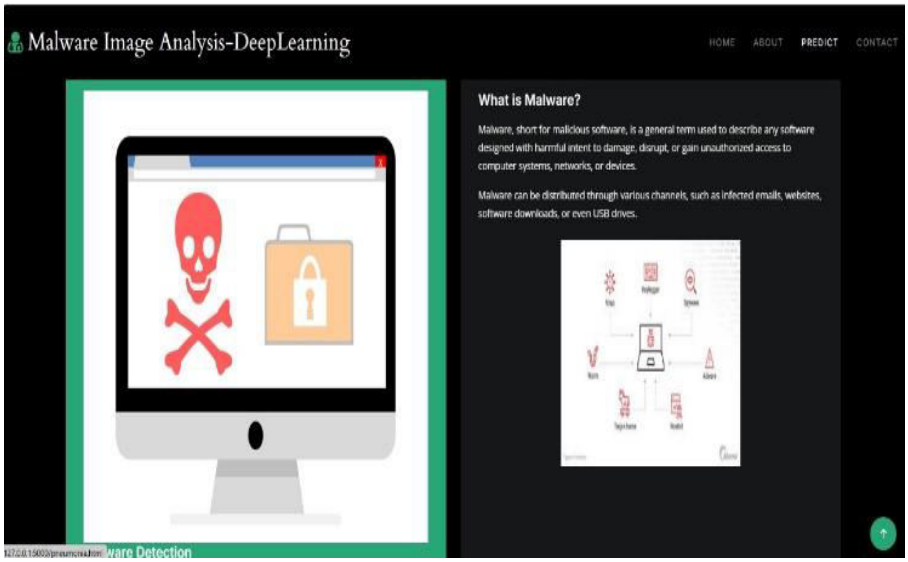


Figure 4: Malware Analysis Information Page

© 2025 IJMRSET | Volume 8, Issue 4, April 2025| DOI: 10.15680/IJMRSET.2025.0804091

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)

Figure 4 showcases an informational page explaining what malware is and how it spreads. The page contains a brief yet detailed explanation of malware, describing it as malicious software designed to harm or disrupt computer systems. Various types of malware, such as viruses, worms, Trojans, spyware, and adware, are illustrated through an informative diagram.The left side of the screen displays an image of a computer with a red skull icon and a lock, symbolizing a security threat. This visual representation serves as an alert, emphasizing the dangers of malware infections. The right side of the screen provides textual information, educating users about different malware types and their attack vectors, such as infected email attachments, compromised software, and malicious websites.

Additionally, the navigation bar at the top includes options like Home, About, Predict, and Contact, suggesting that this is a web-based application where users can learn about malware, submit files for prediction, and contact the developers or support team if needed.

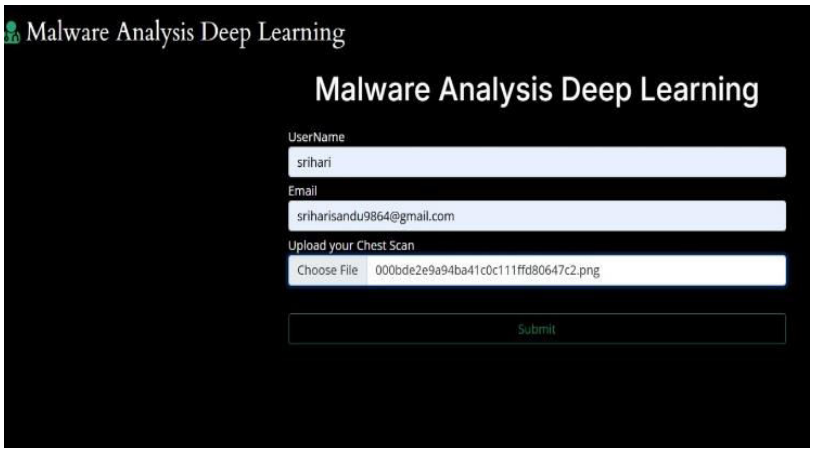


Figure 5: User Input Interface for Malware Analysis

Figure 5 represents the user interface where users input their details and upload a file for malware analysis. The form consists of fields for entering a username, email, and an option to upload an image or file. Once the user provides the necessary details and selects a file, they can submit it for analysis. This interface is designed with a simple and user-friendly layout, ensuring that users can easily navigate and interact with the system.

The title of the application, **"Malware Analysis Deep Learning,"** indicates that the system leverages deep learning models to analyze malware. The uploaded file is likely converted into an image format for processing, which helps in identifying patterns and characteristics of malware through computer vision techniques.

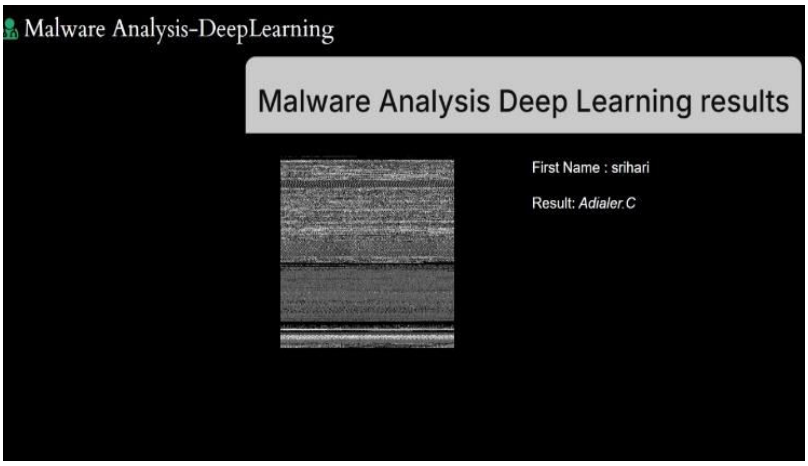


Figure 6: Malware Detection Result

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)

Figure 6 displays the results page of the malware analysis system. After a user submits a file, the system processes it using deep learning algorithms and displays the detected malware type. In this example, the user "Srihari" uploaded a file, which the system analyzed and identified as "Adialer.C." This name corresponds to a known type of malware, specifically a dialer Trojan, which is designed to make unauthorized premium-rate phone calls or send messages without user consent.

The background of the results section features a grayscale pattern, which is likely a visual representation of the malware file in image format. Many deep learning-based malware detection systems convert executable files into grayscale images before classifying them using Convolutional Neural Networks (CNNs). The pattern reflects unique byte-level structures of the malware, helping the model differentiate between benign and malicious files.
This result page plays a crucial role in informing users about potential threats in their uploaded files, allowing cybersecurity teams to take necessary precautions.

VI.CONCLUSION

In In this research, we used deep learning techniques to construct a malware classification system, especially utilizing the XceptionNet model using transfer learning. We detected and categorized malware into 25 different families by transforming malware binary data into grayscale photos and applying image classification techniques. Accuracy was greatly increased while training time and computational expenses were decreased by using a pre-trained XceptionNet model. The effectiveness of deep learning in protective applications is demonstrated by the suggested system's excellent accuracy (~98.5% on test data) in malware classification. The Malimg dataset, which had 9,339 grayscale photos, was used to train the model. To improve its prediction power, the model was pre-processed using techniques including thresholding, normalization, and scaling.

The system offers a user-friendly online interface for malware categorization with Flask integration, allowing users to input photos of suspected malware and get real-time predictions. This useful program fortifies cybersecurity defences and improves malware detection. All things considered, the experiment demonstrates how deep learning can be used to classify malware, providing a scalable and automated method of spotting online dangers. To give further insights into malware behaviour, future developments may involve growing the dataset, adding new deep learning architectures, and improving interpretability methodologies.

VII. FUTURE WORK

Future improvements can focus on expanding the dataset with more malware families, integrating real-time malware detection, and enhancing model efficiency for faster predictions. Additionally, exploring other deep learning architectures and hybrid models may further improve accuracy and adaptability to evolving malware threats.

REFERENCES

[1] S. M. Robert Moir, “Defining Malware.” [Online]. Available: https://technet.microsoft.com/en-us/library/dd632948.aspx. [Accessed: 01-Oct-2003].
[2] Kaspersky, “Kaspersky Lab detects 360,000 new malicious files daily – up 11.5% from 2016,” 2017. [Online]. Available: https://www.kaspersky.com/about/pressreleases/2017_kaspersky-lab-detects-360000-new-malicious files-daily. [Accessed: 29-Aug-2018.
[3] The Cylance Team, “How Traditional Antivirus Works,” 2017. [Online]. Available: https://threatvector.cylance.com/en_us/home/how-traditionalantivirus-works.html. [Accessed: 29-Aug-2018].
[4] K.-S. Han, I.-K. Kim, and E. G. Im, “Malware Classification Methods Using API Sequence Characteristics,” Springer, Dordrecht, 2012, pp. 613–626.
[5] C. Liangboonprakong and O. Sornil, “Classification of malware families based on N-grams sequential pattern features,” in 2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA), 2013, pp. 777–782.
[6] E. K. Kabanga and C. H. Kim, “Malware Images Classification Using Convolutional Neural Network,” J. Comput. Commun., vol. 06, no. 01, pp. 153–158, Dec. 2018.
[7] The Keras Team, “Documentation for individual models,” 2018. [Online]. Available: https://keras.io/applications/. [Accessed: 01- Sep-2018].

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)

[8] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware Images: Visualization and Automatic Classification." 23-Feb-2016.
[9] M. Kalash, M. Rochan, N. Mohammed, N. D. B. Bruce, Y. Wang, and F. Iqbal, "Malware Classification with Deep Convolutional Neural Networks," in 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), 2018, pp. 1–5.
[10] M. Lin, Q. Chen, and S. Yan, "Network In Network," 2013.
[11] A. Makandar and A. Patrot, "Malware Image Analysis and Classification using Support Vector Machine," Int. J. Adv. Trends Comput. Sci. Eng., vol. 4, no. 5, pp. 1-03, 2015.
[12] Kaggle Team, "Microsoft Malware Winners' Interview: 1st place, 'NO to overfitting! - Confusion matrix' | No Free Hunch," 2015. [Online]. Available:http://blog.kaggle.com/2015/05/26/microsoft-malware-winners interview-1st-place-no-to-overfitting/. [Accessed: 05-Sep-2018].
[13] F.-F. Li, J. Johnson, and S. Yeung, "CS231 Lecture 09: CNN Architectures," Cs 231, pp. 1–101, 2017.
[14] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," 2016.
[15] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, "Microsoft Malware Classification Challenge," Feb. 2018.
[16] Microsoft, "Microsoft Malware Classification Challenge (BIG 2015) | Kaggle," 2015. [Online]. Available: https://www.kaggle.com/c/malware-classification#evaluation. [Accessed: 29-Aug-2018].

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