Derek Peng
Troy High School

Visual Malware Classification Using a Vision Transformer

Faculty Advisor:
Dr. Mohammad Husain
Cal Poly Pomona

Problem

Malware continues to pose a significant threat to cybersecurity, with the frequency and sophistication of attacks increasing annually. Traditional malware classification methods, including static and dynamic analysis (YARA Rules), often require substantial computational resources and time. Static analysis relies on **detecting known malware signatures**, making it vulnerable to obfuscation techniques. Dynamic analysis, while effective against obfuscation, involves executing malware in a sandbox environment, which is both time-consuming and resource-intensive. These methods may not keep pace with the rapid evolution of malware, highlighting the **need for more efficient and robust classification techniques.**

Approach

To address the limitations of conventional malware analysis, this paper proposes a novel approach utilizing **vision transformers (ViT) for malware classification**. By converting malware binaries into grayscale images, the proposed method leverages image processing techniques to classify malware based on visual patterns. Traditionally, machine learning techniques for image classification would utilize convolutional neural networks. However, the recent advances in using vision transformers have shown that they can be **more resistant** to **data/concept drift**, and can be **interpretable** through analysis of their attention maps. In this research, a baseline CNN and a ViT model were developed to analyze the Malimg dataset (25 families of Windows malware), shown in Fig. 1. Their architectures are shown in Fig. 4. The **goal** of this research was to test whether vision transformers have potential in delivering **real-time, interpretable, and accurate analysis in malware classificati**

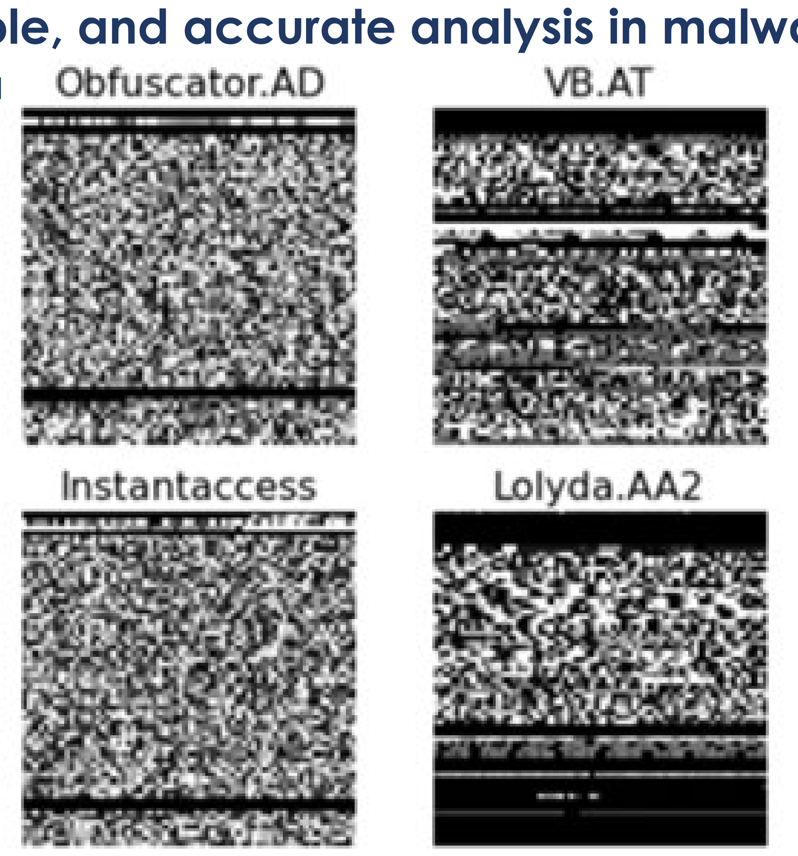

Fig. 1: Malimg Dataset (4 Families Shown)

Research Question 1: Is it possible to develop and train an **accurate** vision transformer model to classify binary image files of malicious malware samples?
Research Question 2: What is the trade-off between different machine learning models in accuracy and speed?

Methodology

A region-based Vision Transformer (RegionViT)[1] for image classification was implemented using PyTorch. Opposed to a vanilla ViT, **RegionViT achieves a similar accuracy while training less**. Additionally, it uses regional-to-local tokens to have context on its neighboring sections. The model was configured with four stages of encoders to analyze different regions of the binary image. The model was trained using an Adam optimizer with a learning rate of 0.003 and cross-entropy loss as the criterion. Training was performed on an AWS EC2 instance (ml.g4dn.xlarge), with a Nvidia Tesla T4 GPU.

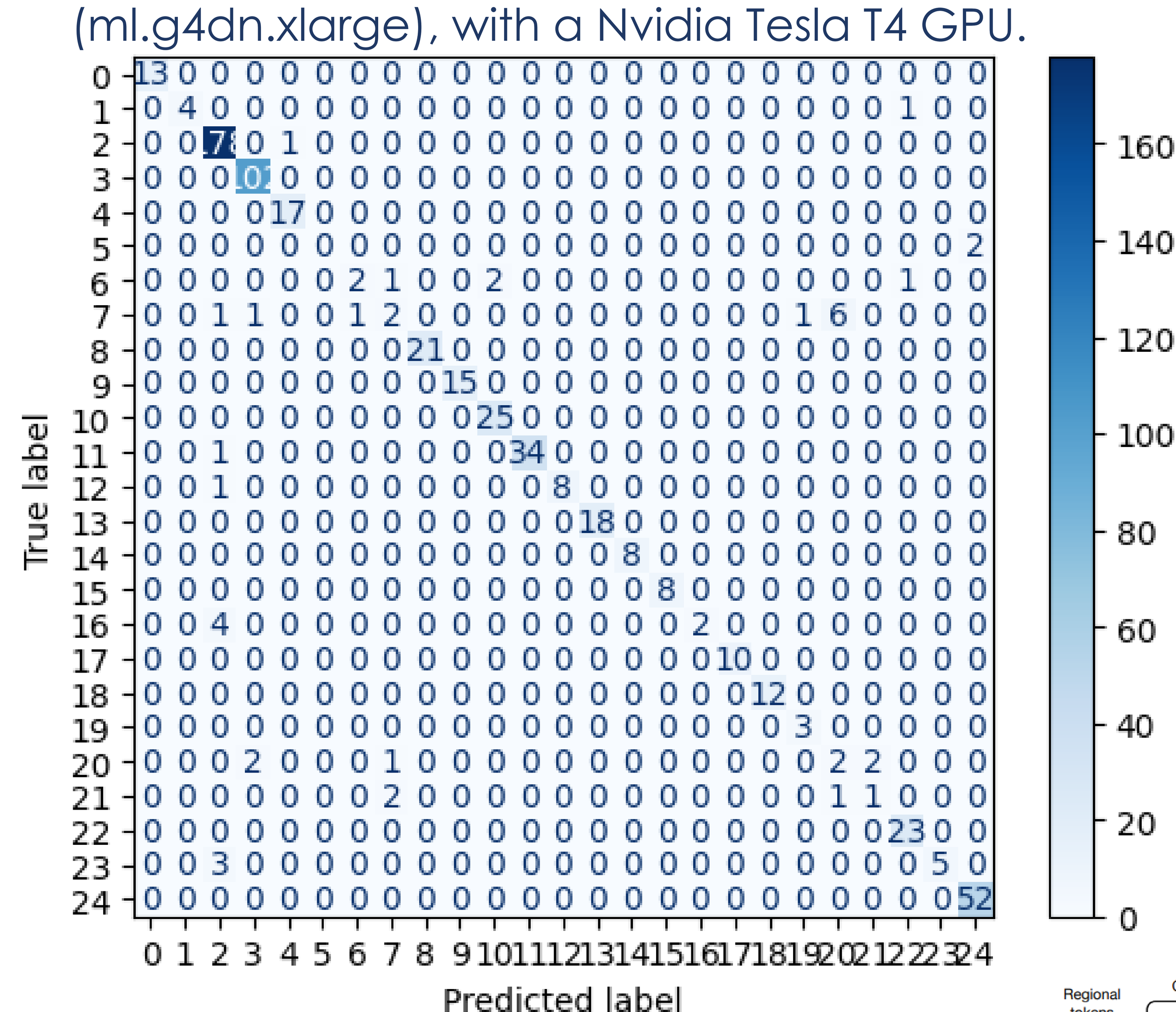

Fig. 2: ViT Model Training Statistics

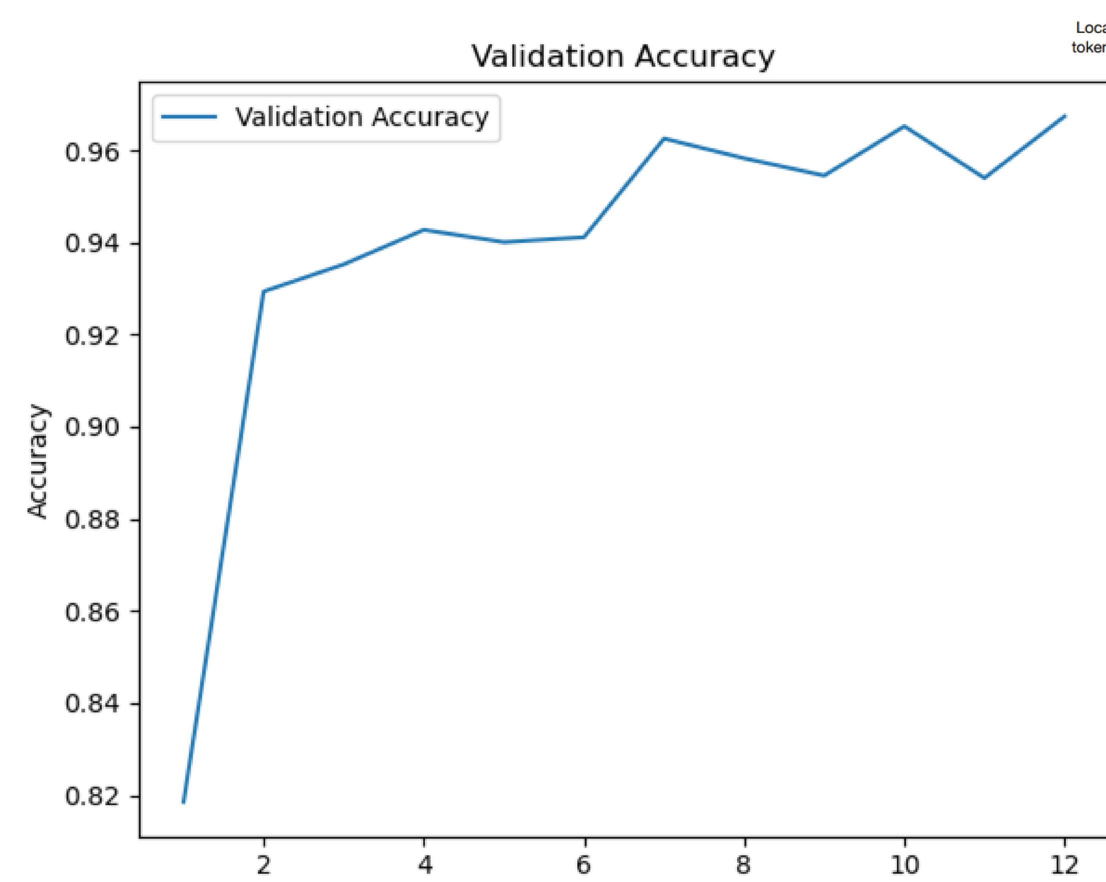

Fig. 3: ViT Model Training Statistics

Results

Overall, the model I designed had an accuracy of **97.8%** on the testing split of the dataset. For a large majority of the classes, the model performed well, producing an average accuracy of 0.97 across all classes, and many having perfect recall. However, some classes, such as class 20 corresponding to TDownloader in Fig. 2, were **particularly tricky** to classify, despite retraining the model in attempts to improve on the accuracy. I generally got F1, recall, and precision scores of about 97% on average. However, computing power was a limitation. RegionViT is designed to be more efficient in training than a traditional ViT, but still took hours to train. As a result, a weaker model than what was possible was developed. In Fig. 3, the Train Loss and Validation Accuracy has not flattened out after 12 epochs, indicating a **higher possible accuracy is able to be achieved**. After training, it contained **57.10M parameters**. In total, to classify the dataset of 9339 images, it took **101.58 seconds.** In comparison, a similar CNN baseline model achieved an accuracy of 97.6%, with 1.8M parameters, able to classify the dataset in 6.58 seconds. CNNs are generally faster, but the ViT model with 57.10M parameters was able to produce results in a reasonable amount of time.

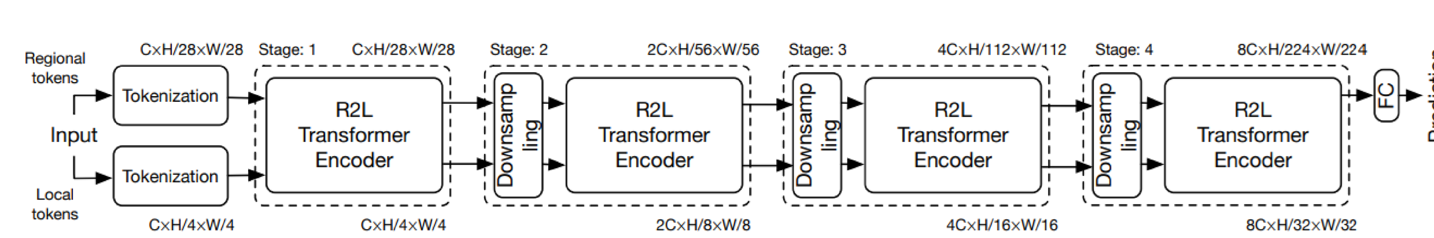

Fig. 4a: ViT Model Architecture

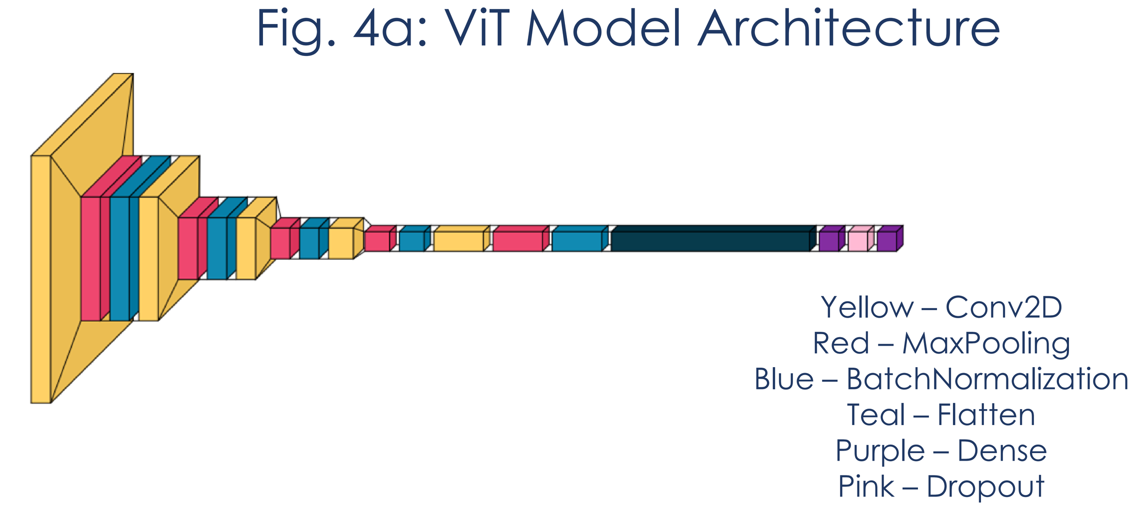

Fig. 4b: CNN Baseline Model Architecture

Conclusion

By analyzing grayscale images of malware binaries, the model **effectively** identifies and classifies malware into distinct families, showcasing the potential of image-based techniques for cybersecurity. Despite some challenges, including data preprocessing, model complexity, and class imbalance, the results indicate that vision transformers offer a viable and cost-effective alternative to traditional malware analysis methods. It **demonstrates the feasibility** of using vision transformers for future development of tools to classify malware to supplant existing methods and other machine learning methods(CNNs).

Future Works

Future work could incorporate larger datasets as vision transformers are **generally intended for datasets with 10M+ members**. Additionally, the resistance of vision transformers in data drift compared to other ML techniques could be explored as a method to combat concept drift. **Google's CoAtNet** (combining convolutional and transformer architecture) could be explored. Additionally, **interpretability** with the **attention map** of the vision transformer would greatly help experts identify reasoning for a classification.

References

[1] Chun-Fu Chen, Rameswar Panda, and Quanfu Fan. "RegionViT: Regional-to-Local Attention for Vision Transformers". In: International Conference on Learning Representations. 2022. URL: https://openreview.net/forum?id=T V3uLix7V.
[2] L. Nataraj et al. "Malware images: visualization and automatic classification". In: Proceedings of the 8th International Symposium on Visualization for Cyber Security. VizSec '11. Pittsburgh, Pennsylvania, USA:Association for Computing Machinery, 2011. ISBN:9781450306799. DOI: 10.1145/2016904.2016908. URL:https://doi.org/10.1145/2016904.2016908.

Acknowledgements

I would like to thank all those who helped me develop the idea for this project. Firstly, **Professor Husain** for giving me this opportunity, and the **TAs** for guiding me through issues I faced. Furthermore, I would like to thank my friends who helped me edit and suggested new ideas and innovations.