DYNAMIC MALWARE DETECTION USING GENERATIVE AI

SUPERVISOR

Dr. Sufian Hameed

CO-SUPERVISOR

Dr. Muhammad Rafi

GROUP MEMBERS:

Muhammad Hatif Mujahid (20K-0218)

Wahaj Javed Alam (20K-0208)

Agha Maarij Amir (20K-0160)

ABSTRACT

Malware Detection and prevention is critical for the security of a system.

Malware Obfuscation and encryption are common techniques used to avoid static detection

We propose a Deep Learning method for Malware Classification and Detection

It leverages GANs alongside Deep Autoencoders and CNNs to acheived the proposed task.

PROBLEM STATEMENT

Malware poses a significant threat to computer and mobile device security, with the potential to gain unauthorized access

Traditional signature-based detection methods, although effective to some extent, face scalability limitations, restricting their utility in identifying new and diverse malware threats.

Many of these approaches relied on hand-crafted features, requiring expert knowledge and limited generalizability.

As a result, there is a need for innovative approaches capable of addressing these issues and enhancing cybersecurity against evolving malware threats.

LITERATURE REVIEW

Malware Detection with Malware Images using Deep Learning Techniques [1]

Premise:

- Used CNN and CNN combined with SVM for classification
- Dataset used was Malimg Dataset

Solution/Results

- CNN accuracy: 97.58%
- CNN with SVM accuracy: 89%

Challenges

- Implementing SPP was impractical due to memory constraints.
- The study found that greyscale imaging was effective against redundant API injection, indicating a specific challenge in malware detection

HYDRA: A multimodal deep learning framework for malware classification [2]

Premise:

- Traditional methods rely on expert-designed features.
- HYDRA combines various feature types to address these issues.

Solution/Results

- Merges hand-engineered and end-to-end components.
- Evaluated on Microsoft Malware Classification Challenge.
- Achieved 99.75% accuracy and 99.51% Macro F1 score.

Challenges

- Integrating multiple feature types to have a consistent result
- Memory issues when dealing with many frameworks and features.

Malware Classification Using Static Disassembly and Machine Learning [3]

Premise:

Extracted 7 Features and used Auto Sklearn with RF, SVM and KNNs

Solution/Results

- Big 2015 Dataset was used.
- An accuracy of 99.48% was achieved.
- Used File size, API 4-grams, OPCODE 4-grams, import libraries, PE section size and permissions, content complexity as features

Challenges

- Code obfuscation and encryption
- Name mangling
- Lazy loading

Zero-day Malware Detection using Transferred Generative Adversarial Networks based on Deep Autoencoders [4]

Premise:

Transfer Learning with AutoEncoders, Convolutional Networks and Generative Adversarial Networks

Solution/Results

- Used the Big 2015 Dataset
- Achieved 95.74% accuracy
- converted malware codes to images, then fed to the architecture

Challenges

- Data scarcity
- Adversarial instability
- Evasion Attacks

A novel malware classification and augmentation model based on convolutional neural network [5]

Premise The proposed approach used Convolutional Neural Networks to classify malware

Solution/Results

- Utilizes convolutional neural networks (CNNs) to learn malware features.
- Addresses class imbalance issues through data augmentation using CycleGAN.
- 99.86% accuracy on the BIG2015 dataset and 99.60% accuracy on the DumpWare10 dataset

Challenges

- Suffers against adversarial malware samples

Emulating malware authors for proactive protection using GANs over a distributed image visualization of dynamic file behavior [6]

Premise:

Wassertein GANs

Solution/Results

- API calls converted to images using a distributed tranformation approach that allows the images to be decoded back
- A WGAN is fed these images so that it learns to create similar realistic images
- WGAN is more stable and requires less hyperparamter tuning
- The WGAN was able to generate realistic images which could be used to train classifiers to improve detection accuracy

Challenges

- Difficult to train GANs
- Data Scarcity

RESEARCH GAP

Malware RGB Images

Code not open-source

Failure Analysis of work

RESEARCH GAP
IMPLEMENTATION

RGB IMAGES

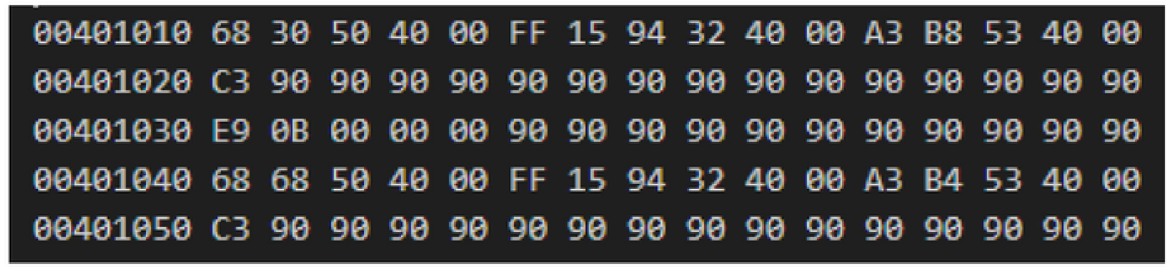
According Ke HE et al. to Converting into RGB images helps CNNs:

- find more complex patterns in the image
- shortens the distance between each pair of bytes because of increase in volume, which increases accuracy.

There were 2 steps used to create the images:

- Bytes code was converted to grey-scale images
- Then the grey-scale images were turned into RGB using intensity of pixels
    - 255 was divided into 3 ranges. If a pixel value was in the red division it was given a red color and so on.

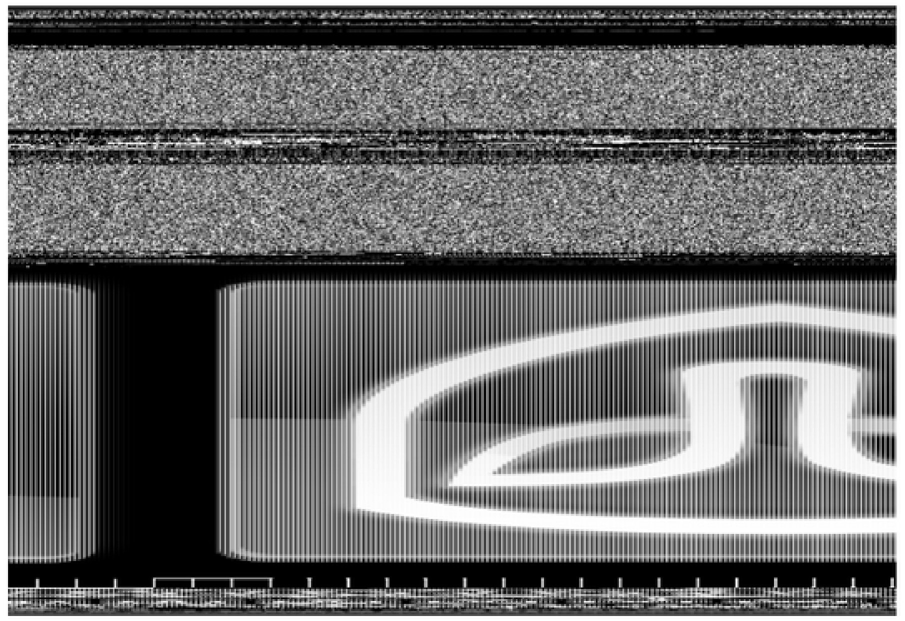
Input Image Pre-Processing



```
00401010 68 30 50 40 00 FF 15 94 32 40 00 A3 B8 53 40 00
00401020 C3 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
00401030 E9 0B 00 00 00 90 90 90 90 90 90 90 90 90 90 90
00401040 68 68 50 40 00 FF 15 94 32 40 00 A3 B4 53 40 00
00401050 C3 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
```

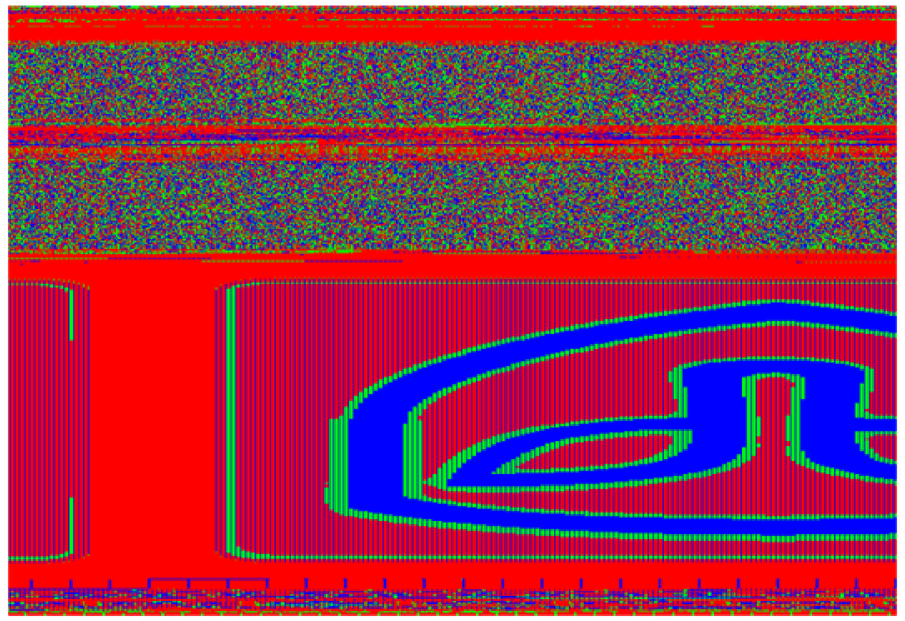
.bytes code

$\rightarrow$



Gray-scale
Image

$\rightarrow$



RGB
Image

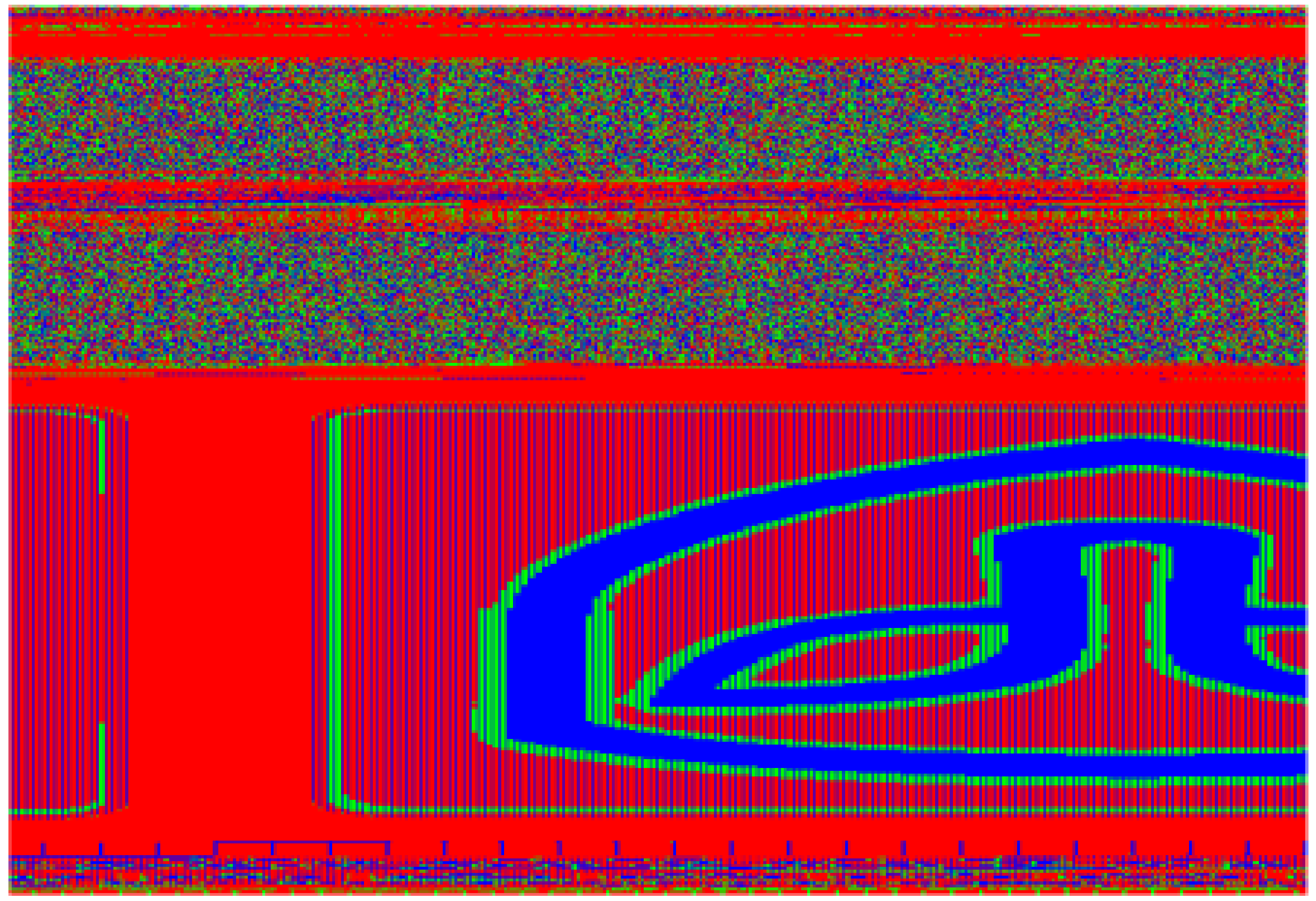
INPUT IMAGES TO CNN

- CNNs are designed to process fixed-size input images.
- This necessitates standardizing malware images to a consistent size before feeding them into the network.

Challenges of Standardization

- Cropping leads to loss of spatial context which may be crucial for identifying patterns
- Cropping might remove important information at image corners
- Resizing can also discard important information related to malware features because of distortion

Image Cropping


1024 x 496

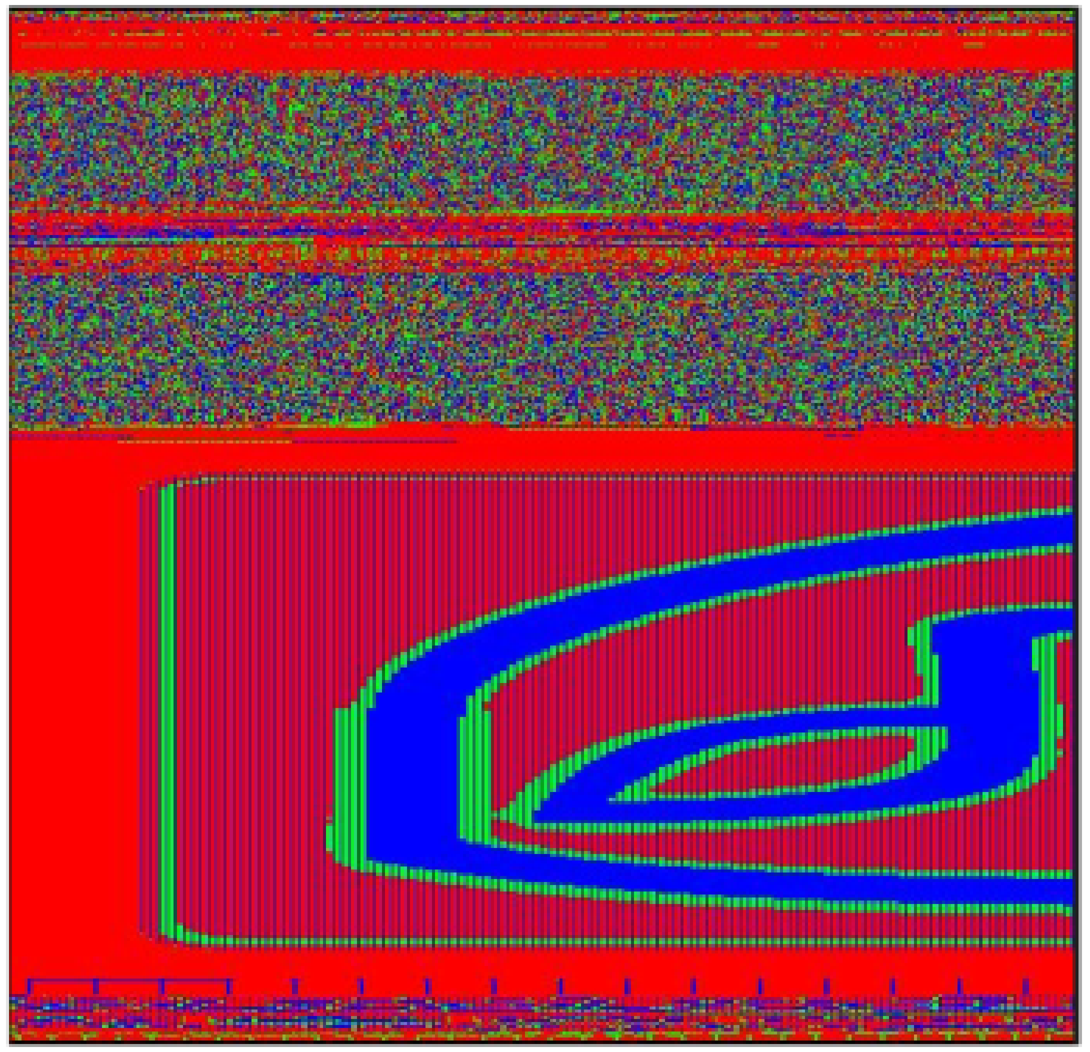
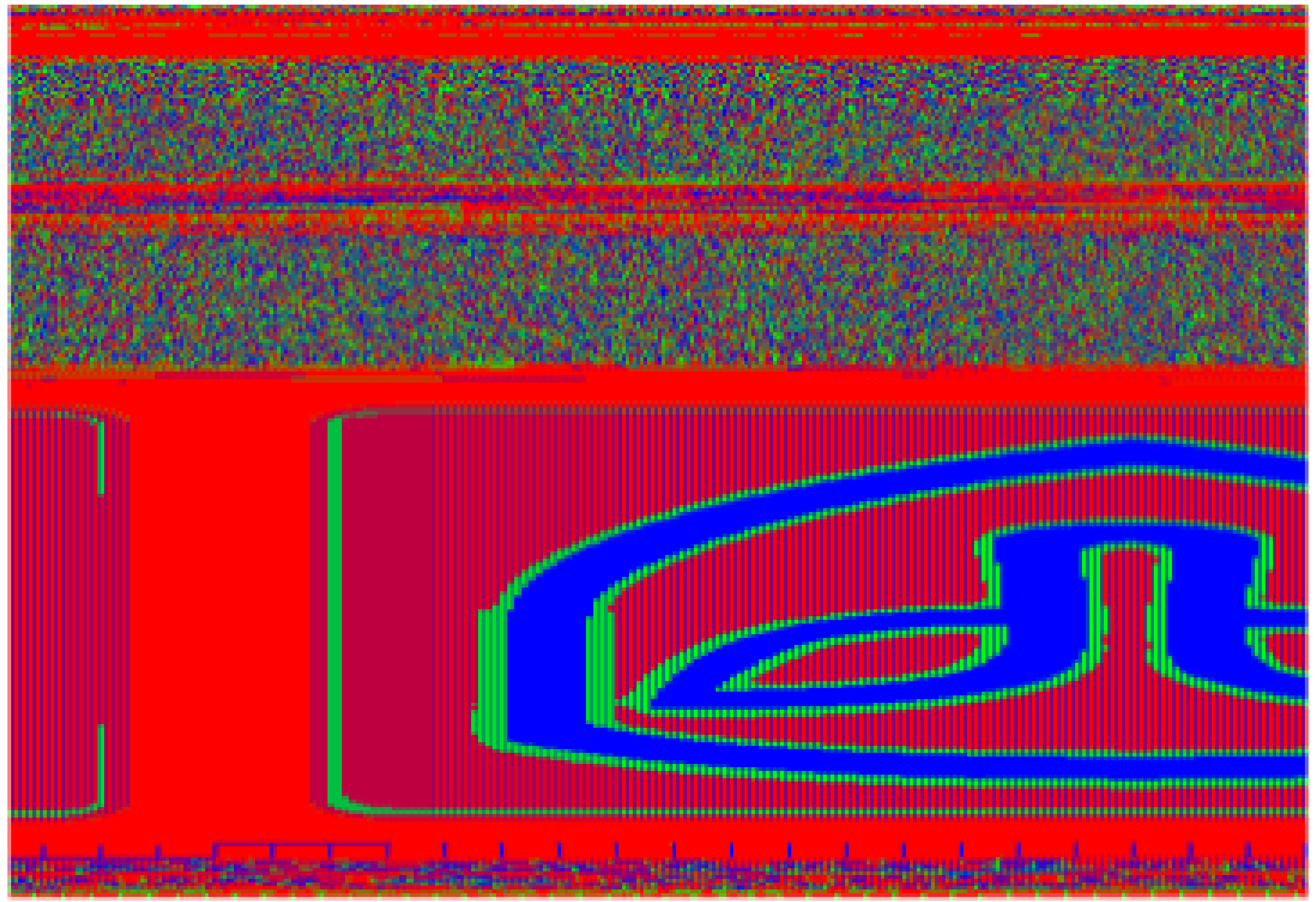

512 x 512

Image Resizing


1024 x 496

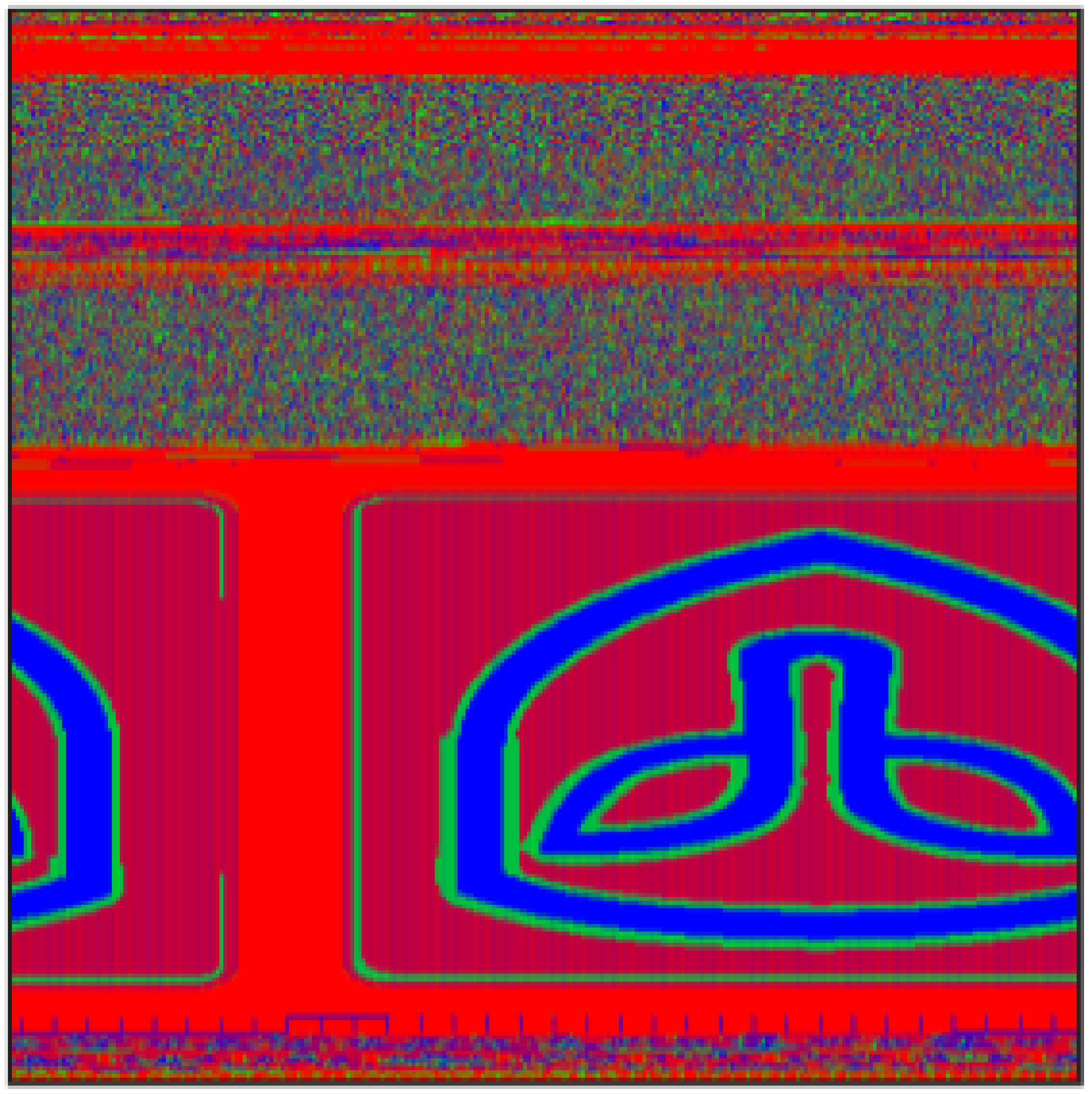

512 x 512

IMAGE PREPROCESSING

RESULTS

There were issues with input image preprocessing:

- Image cropping meant that crucial data was lost.
- Image scaling was a good approach.
- So we scaled the images to 3 different dimensions:
    - 128x128 pixels
    - 256x256 pixels
    - 512x512 pixels

POOLING TECHNIQUES

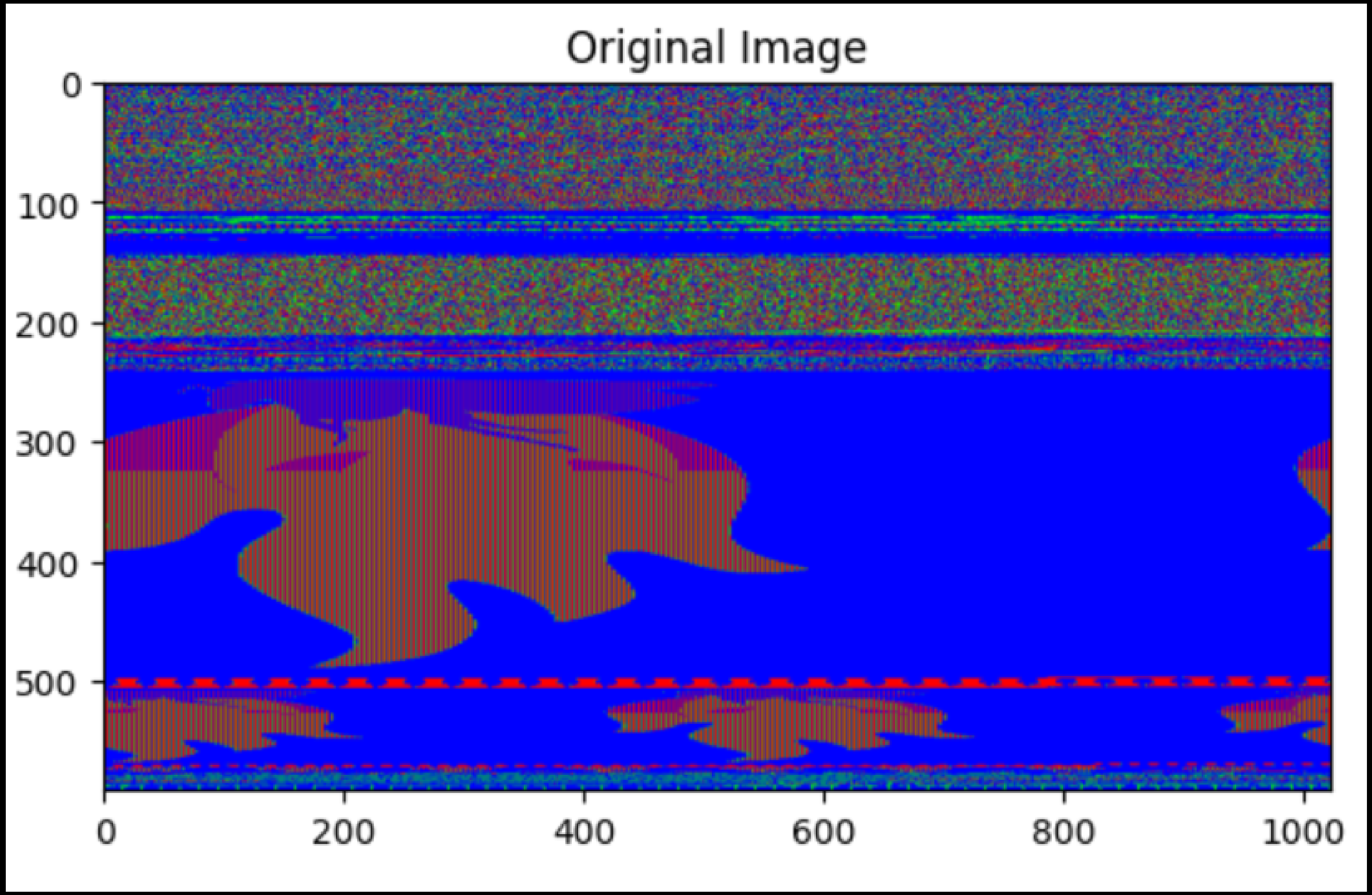
Pooling techniques are used in NN for spatial downsampling, reducing dimensionality and capturing key features

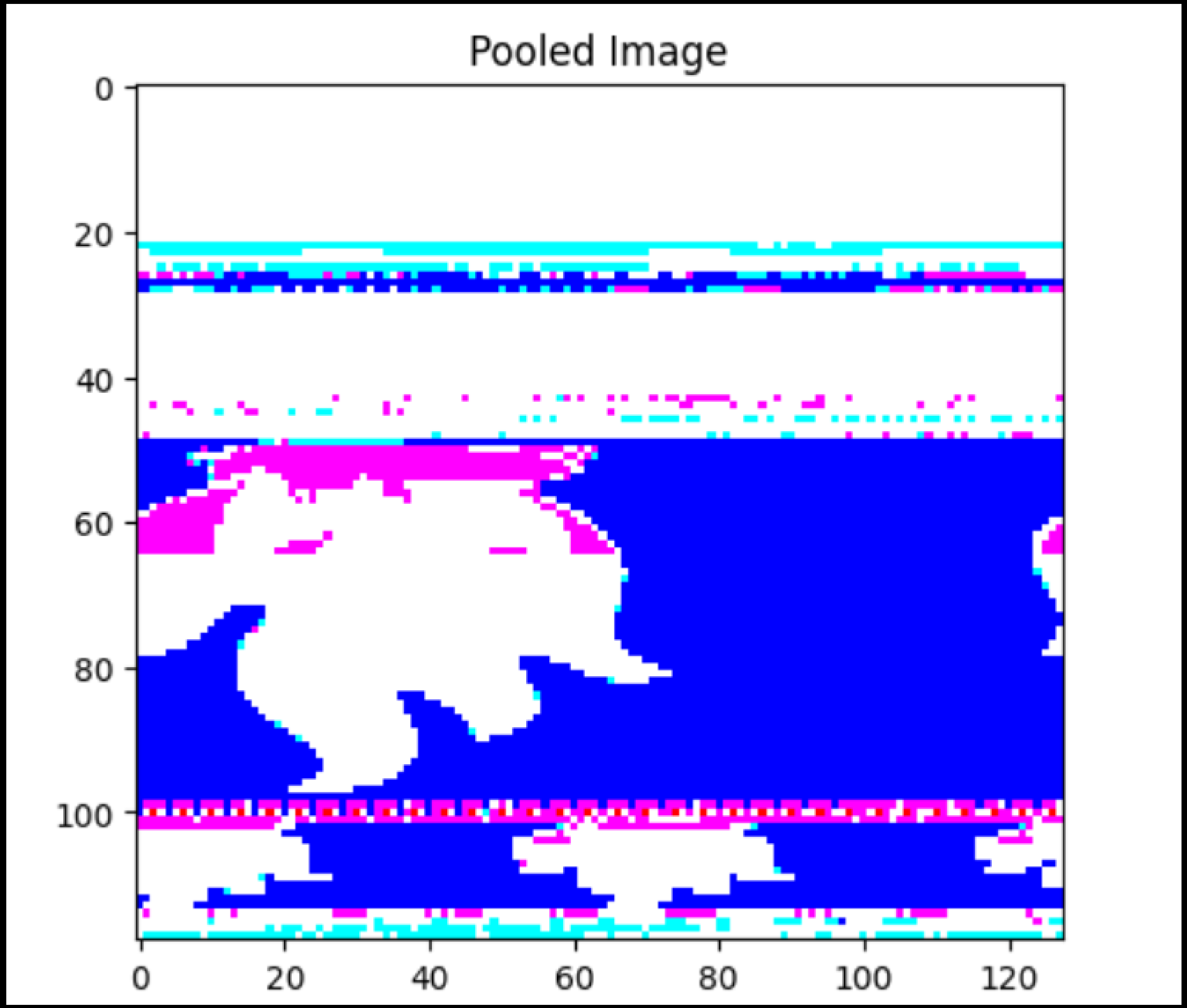
Spatial Pyramid Pooling

- SPP allows variable sized inputs to be fed into the CNN
- SPP divides the input into fixed-size regions at different levels and pooling is applied in each of these regions.
- The pooled output from each level is concatenated to produce a fixed size output

RESULTS OF FINDINGS

- Malware images are very complex and contain a lot of data
- Using pooling on the malware images leads to loss of important information
- The network will not be able to capture and learn features from pooled images leading to degraded performance


Original Image
0
100
200
300
400
500
0
200
400
600
800
1000


Pooled Image
0
20
40
60
80
100
0
20
40
60
80
100
120

DAE & GAN IMPLEMENTATION

DAE Consists of two parts:

- An encoder that compresses the input into a lower-dimensional, A decoder that reconstructs the original input from this compressed representation.

A GAN consists of two parts:

- type of artificial neural network architecture consisting of two distinct networks, a generator and a discriminator, that are trained simultaneously through adversarial training

DAE & GAN ISSUES

IMPLEMENTATION HAD SEVERAL ISSUES:

- Input image size is not defined in the paper, this resulted in high val_loss and lower accuracy.
- RGB implementation was dubious in the paper.
- We tested out several dimensions of images all had little success:
    - 128x128 pixels
    - 256x256 pixels
    - 512x512 pixels
- Achieved only 15% accuracy on 1000 epochs.

DATASET

Data Composition:

It is taken from **Microsoft Malware Classification Dataset on Kaggle**.

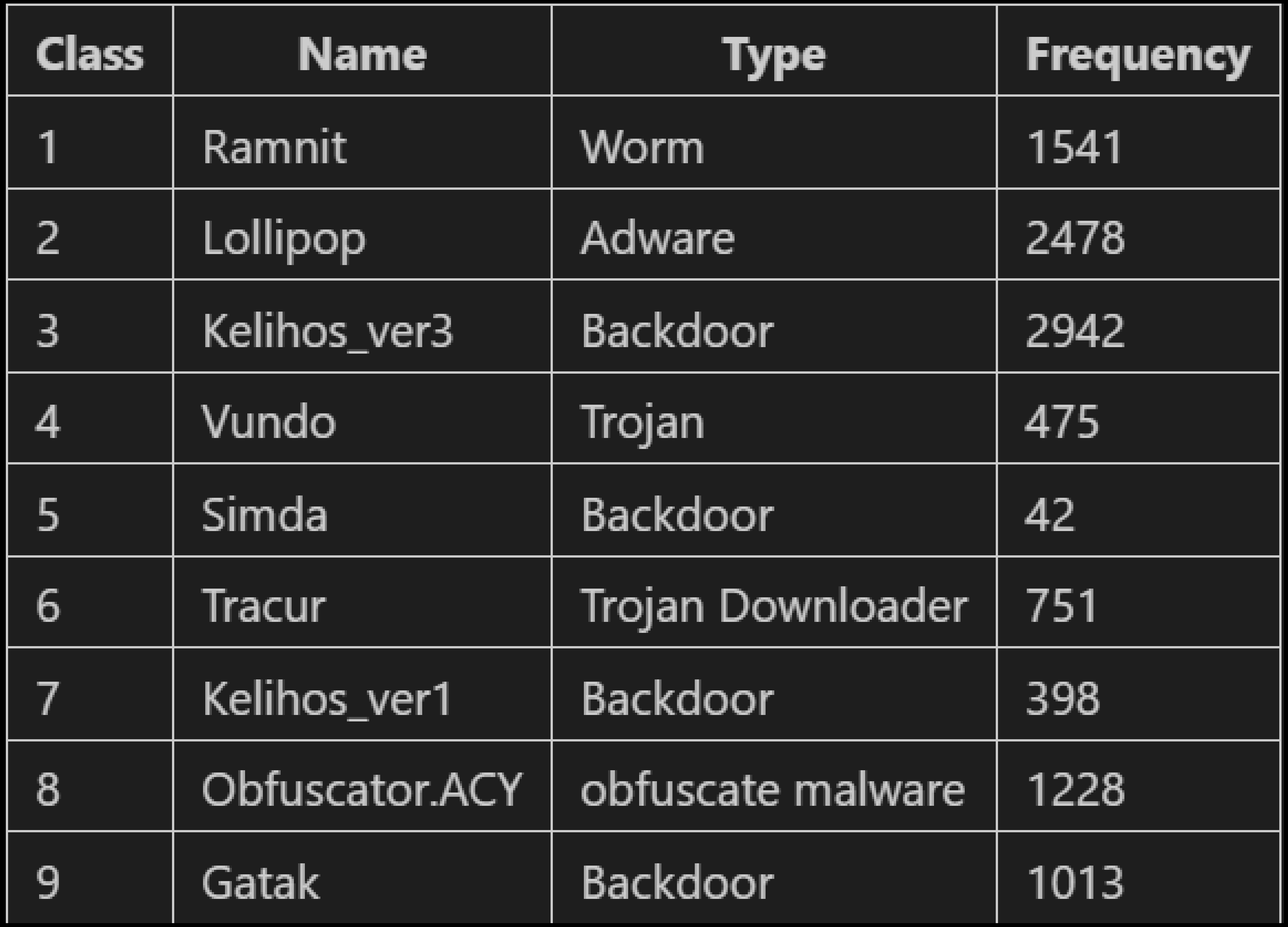
Raw Data

The hexadecimal representation of binary malware files, excluding the PE header for sterility.

Metadata Manifest

Insights into the binary files' structure and behavior.

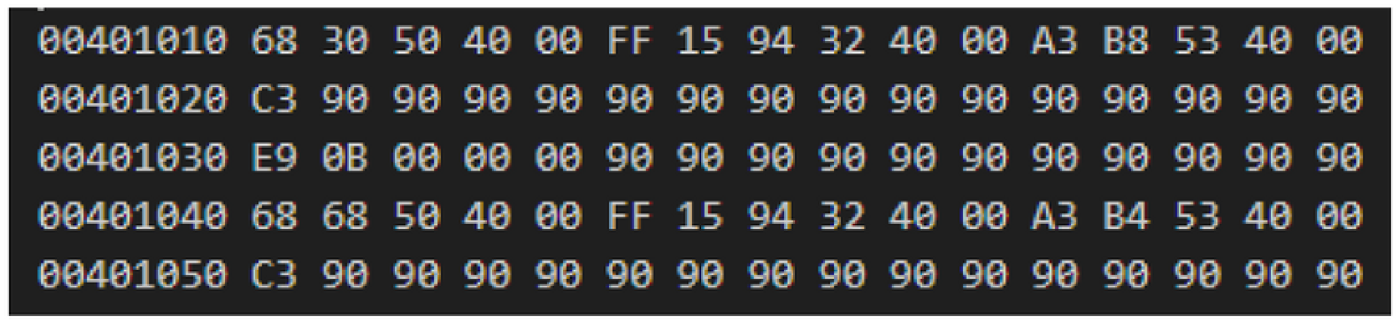
DATASET DESCRIPTION



Class	Name	Type	Frequency
1	Ramnit	Worm	1541
2	Lollipop	Adware	2478
3	Kelihos_ver3	Backdoor	2942
4	Vundo	Trojan	475
5	Simda	Backdoor	42
6	Tracur	Trojan Downloader	751
7	Kelihos_ver1	Backdoor	398
8	Obfuscator.ACY	obfuscate malware	1228
9	Gatak	Backdoor	1013

UPDATED PROPOSED SOLUTION

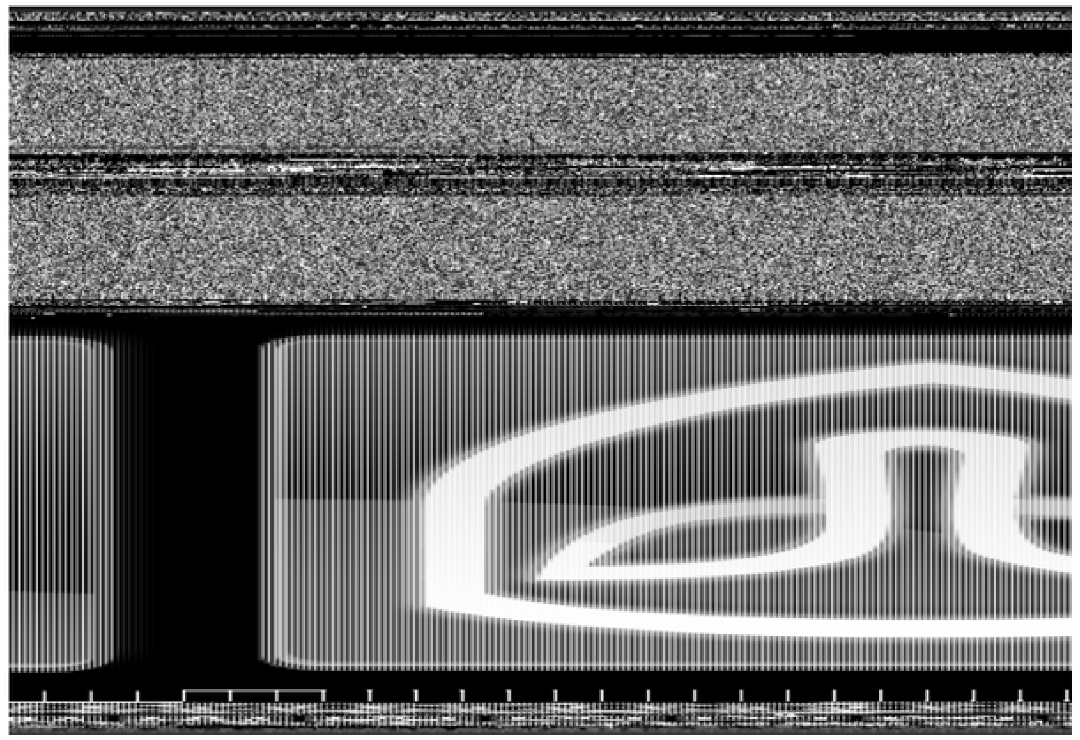
Input Image Pre-Processing



```
00401010 68 30 50 40 00 FF 15 94 32 40 00 A3 B8 53 40 00
00401020 C3 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
00401030 E9 0B 00 00 00 90 90 90 90 90 90 90 90 90 90 90
00401040 68 68 50 40 00 FF 15 94 32 40 00 A3 B4 53 40 00
00401050 C3 90 90 90 90 90 90 90 90 90 90 90 90 90 90 90
```

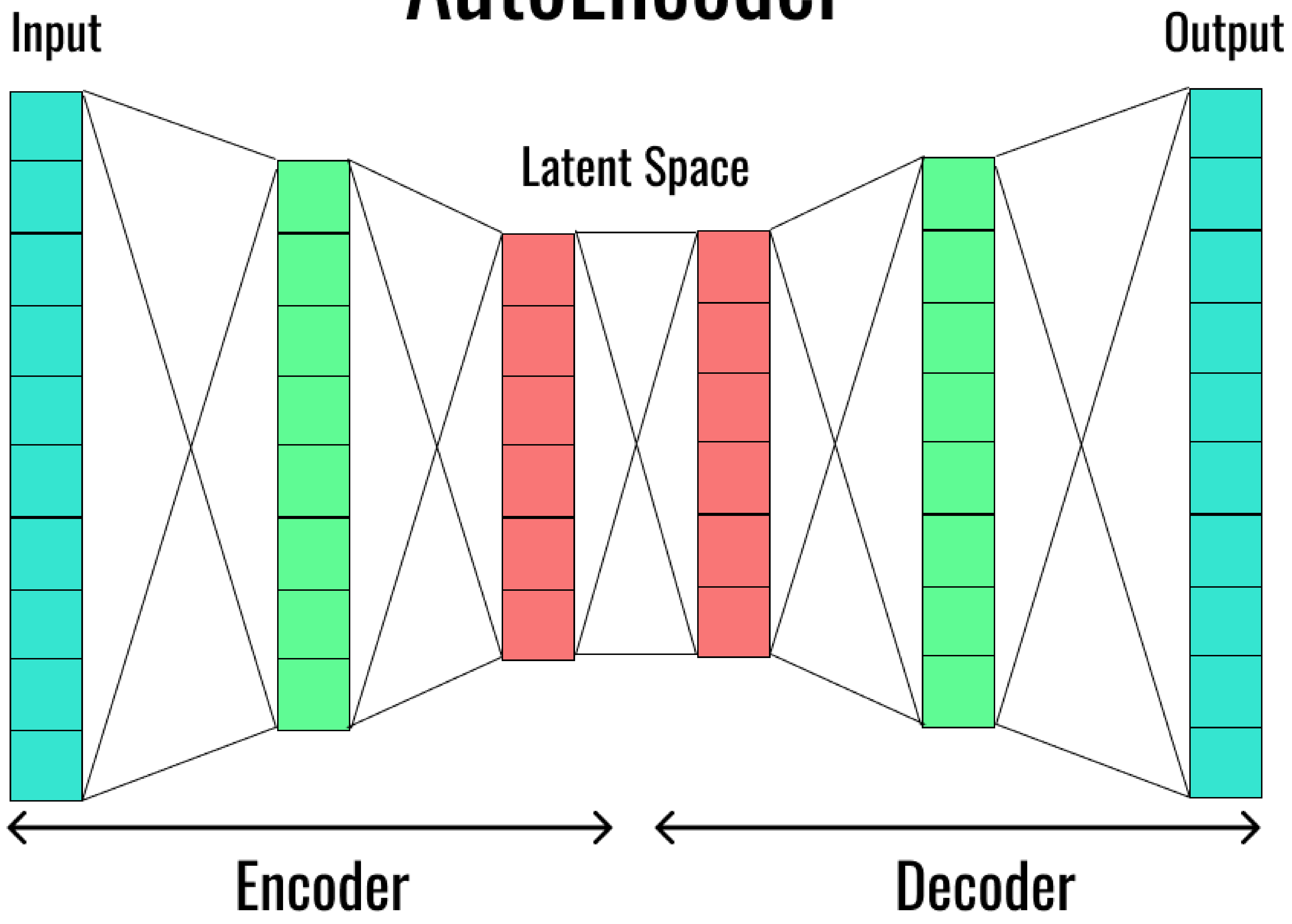
.bytes code

$\rightarrow$

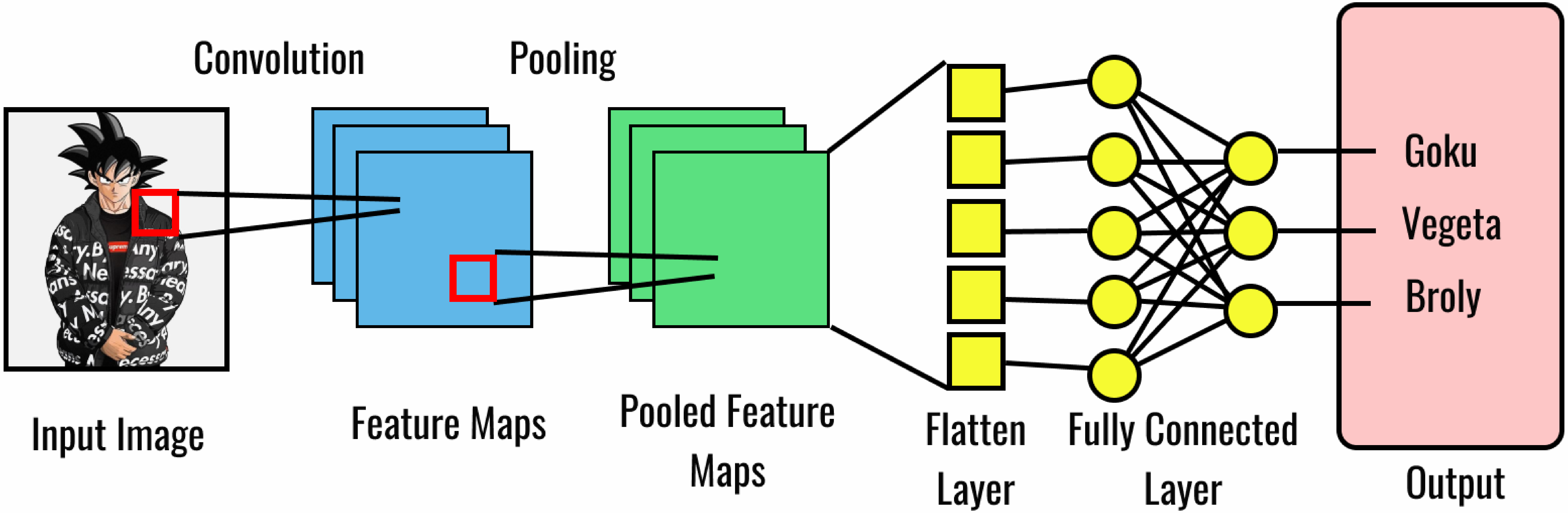


Gray-scale
Image

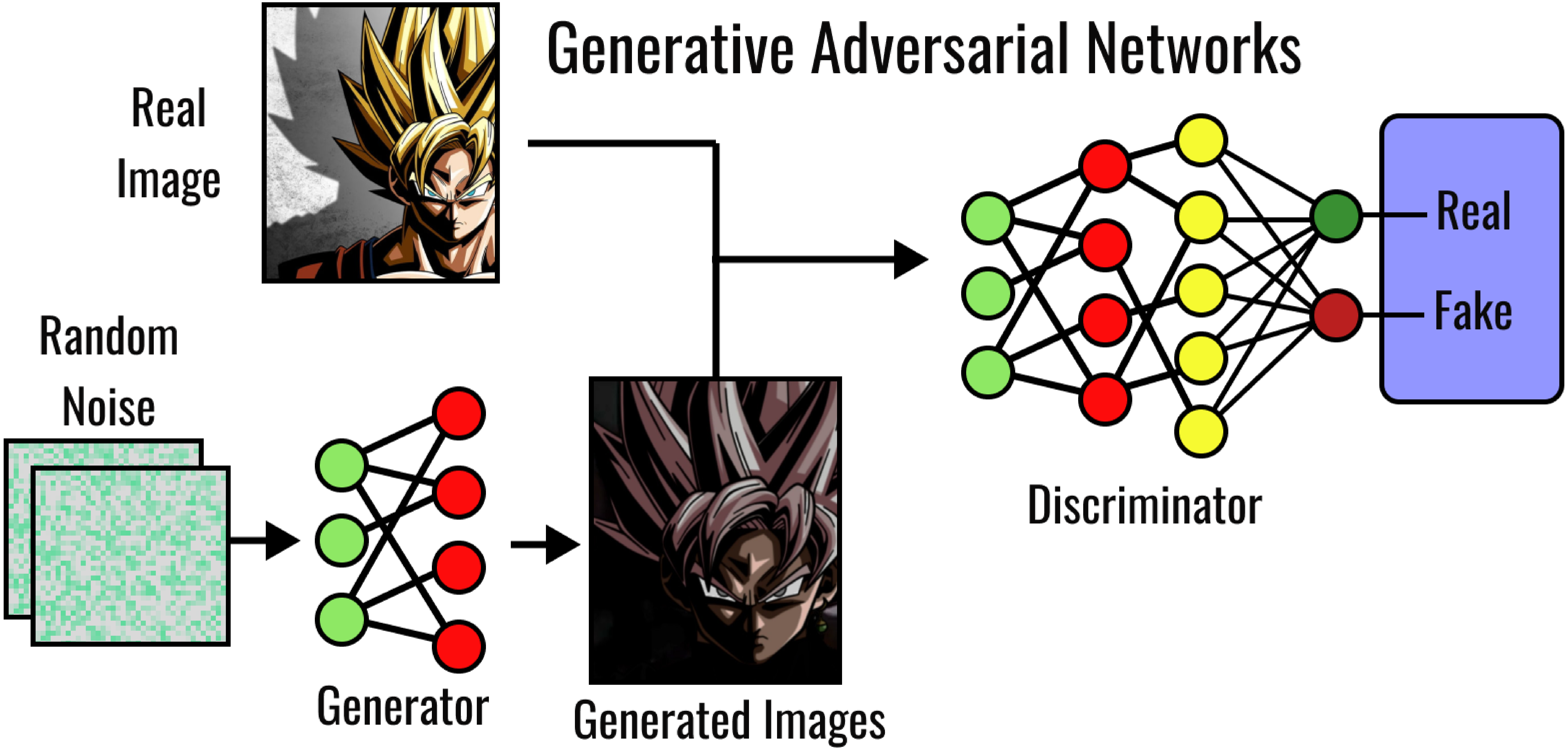
AutoEncoder



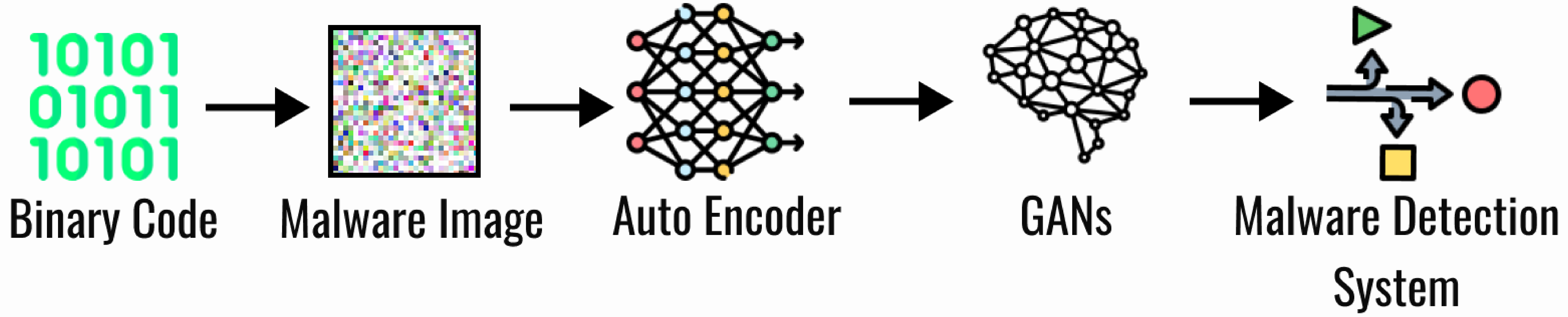
Convolutional Neural Networks



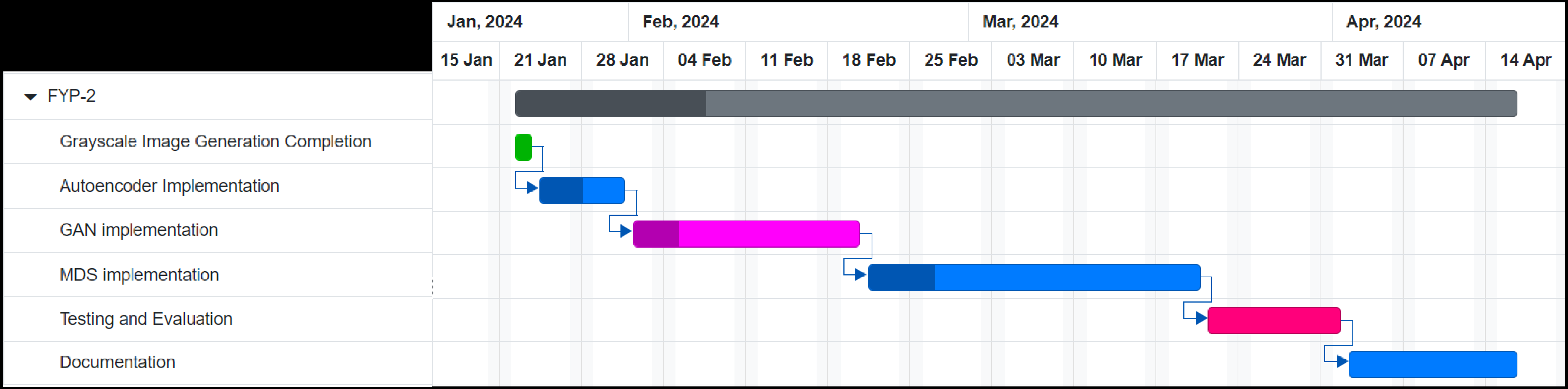
Generative Adversarial Networks



Work Flow


10101
01011
10101
Binary Code
Malware Image
Auto Encoder
GANs
Malware Detection System

TIMELINE



Jan, 2024
Feb, 2024
Mar, 2024
Apr, 2024
15 Jan
21 Jan
28 Jan
04 Feb
11 Feb
18 Feb
25 Feb
03 Mar
10 Mar
17 Mar
24 Mar
31 Mar
07 Apr
14 Apr
FYP-2
Grayscale Image Generation Completion
Autoencoder Implementation
GAN implementation
MDS implementation
Testing and Evaluation
Documentation

REC ●

THANK YOU

REFERENCES

[1] K. He and D. -S. Kim, "Malware Detection with Malware Images using Deep Learning Techniques," 2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), Rotorua, New Zealand, 2019, pp. 95-102, doi: 10.1109/TrustCom/BigDataSE.2019.00022.
[2] D. Gibert, C. Mateu, and J. Planes, “HYDRA: A multimodal deep learning framework for malware classification,” Computers & Security, vol. 95, p. 101873, Aug. 2020, doi: 10.1016/j.cose.2020.101873.

[5] A. Tekerek and M. M. Yapıcı, “A novel malware classification and augmentation model based on convolutional neural network,” Computers & Security, vol. 112, p. 102515, Jan. 2022, doi: 10.1016/j.cose.2021.102515.
[6] V. S. Bhaskara and D. Bhattacharyya, “Emulating malware authors for proactive protection using GANs over a distributed image visualization of dynamic file behavior,” arXiv (Cornell University), Jul. 2018, doi: 10.48550/arxiv.1807.07525.

REFERENCES

[4] J. Y. Kim, S.-J. Bu, and S. B. Cho, “Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders,” Information Sciences, vol. 460–461, pp. 83–102, Sep. 2018, doi: 10.1016/j.ins.2018.04.092.

[3] Z. Chen, E. Brophy, and T. E. Ward, “Malware classification using static disassembly and machine learning,” arXiv (Cornell University), Dec. 2021, doi: 10.48550/arxiv.2201.07649.