CAPSTONE PROJECT DEMO

Static Ransomware Detection

Via PE Header Analysis & Convolutional Neural Networks

Ducournau Ethan

Guechtouli Alexandre

Based on Manavi & Hamzeh (2020)

⚠ The Latency Problem

The Reality: Modern ransomware (like Locky, Ryuk) uses AES-256 encryption. Once executed, files are lost in seconds.

✕Dynamic Analysis Fails

Sandboxes are too slow. Malware "sleeps" to detect virtual environments. By the time behavior is flagged, data is encrypted.

✕Static Signatures Fail

Polymorphism and "Packing" change the file hash and code structure daily. Traditional AV cannot keep up.

```
48 8b b5 58 ff ff ff          movq   -168(%rbp), %rsi
8b bd 8c fd ff ff             movl   -628(%rbp), %edi
ba 00 00 10 00                movl   $1048576, %edx
e8 de f4 ff ff                callq  0x4009d8 <read@plt>
48 89 85 20 ff ff ff          movq   %rax, -224(%rbp)
48 83 bd 20 ff ff ff ff       cmpq   $-1, -224(%rbp)
75 1e                         jne    0x401529 <encrypt_simple+0x1e3>
be 01 00 00 00                movl   $1, %esi
bf a2 84 40 00                movl   $4228258, %edi
e8 e0 f8 ff ff                callq  0x400dfa <print_error>
c7 85 3c fd ff ff 03 00 00 00 movl   $3, -708(%rbp)
e9 1d 03 00 00                jmp    0x401846 <encrypt_simple+0x500>
48 8b 8d 20 ff ff ff          movq   -224(%rbp), %rcx
48 8b 95 58 ff ff ff          movq   -168(%rbp), %rdx
48 8b b5 58 ff ff ff          movq   -168(%rbp), %rsi
48 8d bd 60 ff ff ff          leaq   -160(%rbp), %rdi
e8 19 6b 00 00                callq  0x408063 <sosemanuk_encrypt>
48 8b 85 20 ff ff ff          movq   -224(%rbp), %rax
48 f7 d8                      negq   %rax
48 89 c6                      movq   %rax, %rsi
8b bd 8c fd ff ff             movl   -628(%rbp), %edi
ba 01 00 00 00                movl   $1, %edx
e8 c1 f4 ff ff                callq  0x400a28 <lseek@plt>
48 83 f8 ff                   cmpq   $-1, %rax
75 1e                         jne    0x40158b <encrypt_simple+0x245>
be 01 00 00 00                movl   $1, %esi
bf bd 84 40 00                movl   $4228285, %edi
e8 7e f8 ff ff                callq  0x400dfa <print_error>
```

The Solution: Static Analysis 2.0

We treat the binary file not as code, but as an **image**.

1. PE Header

Extract only the first **1024 bytes**. This contains the "structural DNA" of the file (Signature, Sections).

2. Transformation

Convert bytes to a **32×32 Image** using a Zigzag scan pattern to preserve spatial relationships.

3. Deep Learning

Classify the image using a **CNN** (Convolutional Neural Network) to detect "malicious textures".

Step 1: Feature Extraction

The Target: 1024 Bytes

We do not need the whole file (which can be GBs). We only need the header.

> **MZ Header:** Legacy DOS compatibility.

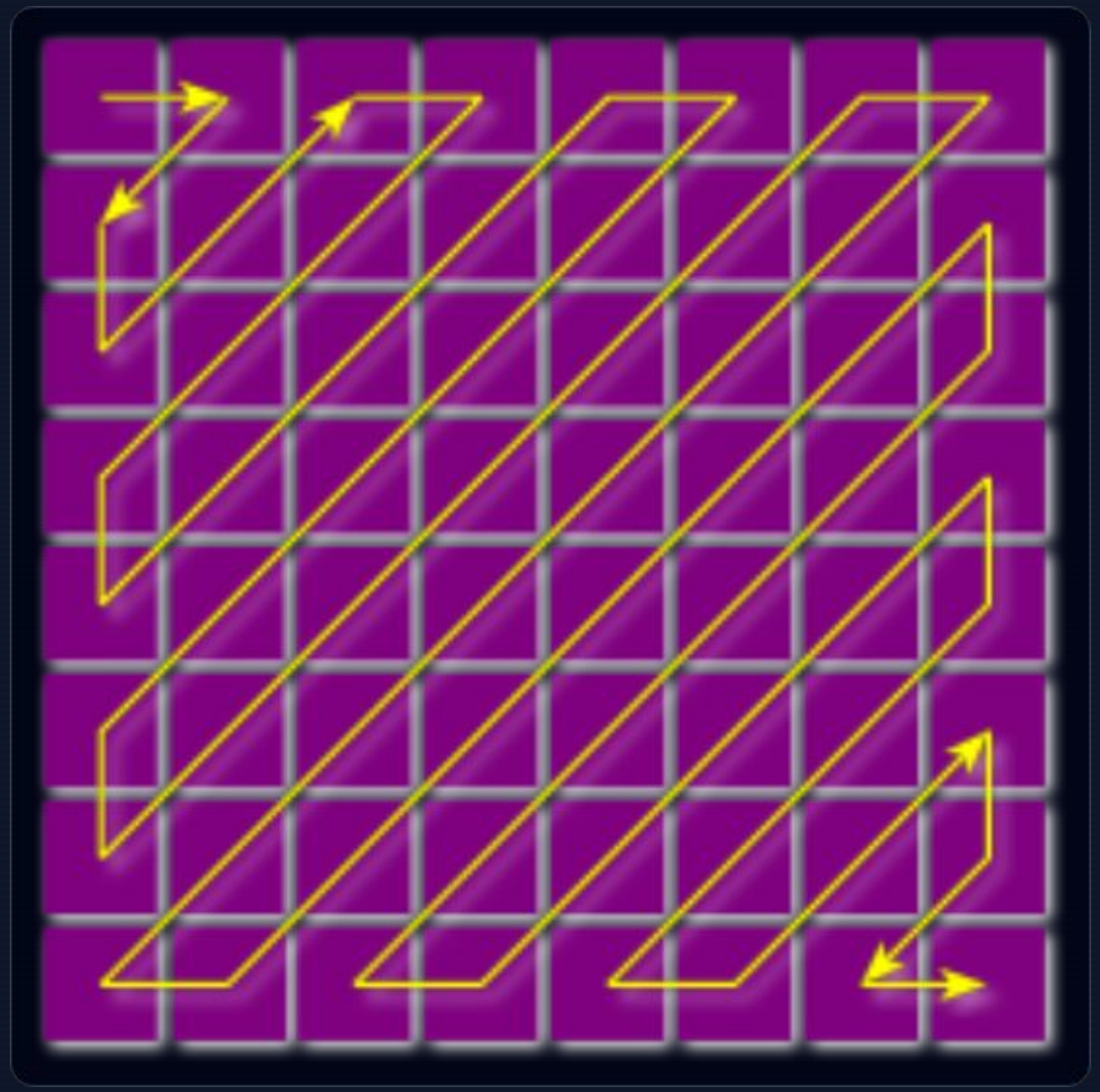
> **PE Signature:** Identifies the file format.

> **Section Table:** Defines code vs. data.
> *Ransomware often has packed/anomalous sections here.

Offset 00 01 02 03 04 05 06 07 00000000 4D 5A 90 00 03 00 00 00 MZ..... 00000008 04 00 00 00 FF FF 00 00 .... 000003F8 00 00 00 00 00 00 00 00 ..... # Total: 1024 Bytes extracted # Output Vector Shape: (1024,)

```
# Total: 1024 Bytes extracted
# Output Vector Shape: (1024,)
```

Step 2: The Zigzag Transform



Why Zigzag?

Unlike a linear scan (row-by-row), the **Zigzag pattern** (borrowed from JPEG compression) fills the matrix diagonally.

Spatial Locality

Key header fields (like the MZ signature) are clustered in the top-left corner. The Zigzag pattern ensures that logically related bytes remain **spatially adjacent** in the 2D matrix.

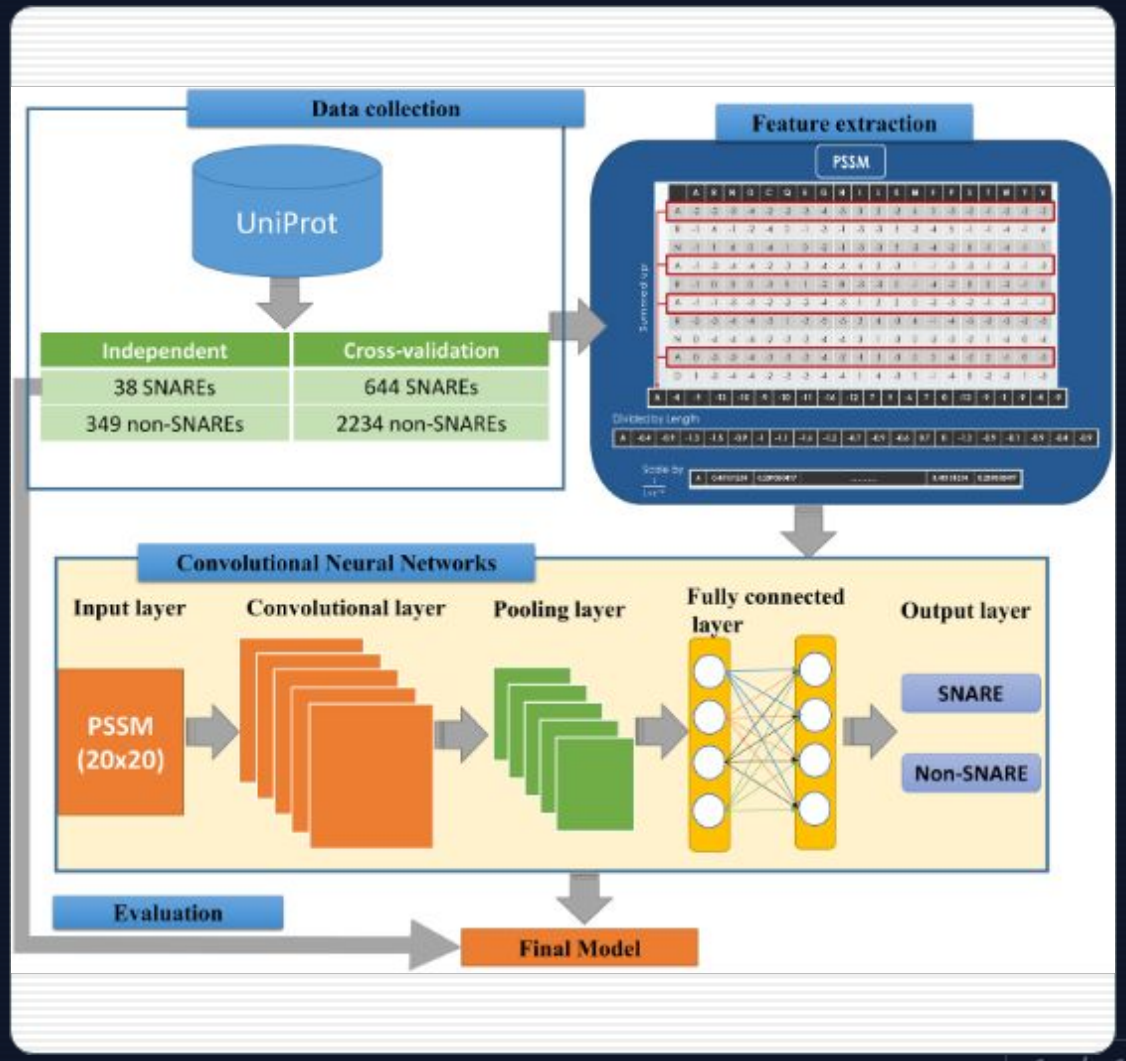
```
Input: [v1, v2, ... v1024]
 Output: 32x32 Grayscale Matrix
```

Step 3: CNN Architecture

Lightweight Model Design

Designed to prevent overfitting on the small dataset (2000 samples).

- `Input:` 32x32x1 (Grayscale)
- `Conv2D:` 64 Filters (3x3) + ReLU
- `Pool:` MaxPooling (2x2)
- `Dropout:` 0.3 (Regularization)
- `Conv2D:` 128 Filters (3x3) + ReLU
- `Pool:` MaxPooling (2x2)
- `Dense:` 16 Neurons + Softmax (2)



Dataset Engineering: Overcoming Critical Class Imbalance

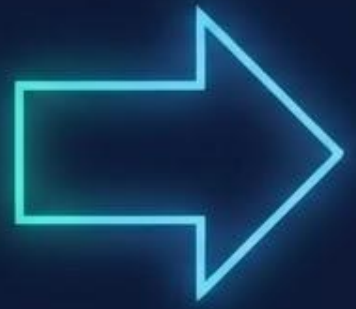
The Problem (The Data Trap)

Real-World Imbalance:

- **~49,400 Ransomware** (96%)
- **~2,500 Benign** (4%)

- **Key Metric:** 96% Malware vs. 4% Benign.
- **The Issue:** The raw "Oliveira" dataset is unusable for deep learning.

Consequence: A model trained on this would suffer from extreme Bias. It would learn to predict 'Ransomware' by default, resulting in a massive False Positive rate on safe files.



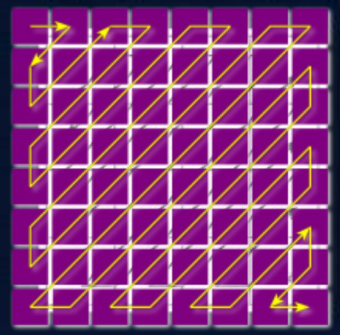
The Solution (Benign Augmentation)

- **Action:** 'Harvesting System Artifacts'
- **Methodology:** To fix the deficit, we developed a script to extract PE Headers from trusted system files (.dll and .exe) directly from the C:\Windows directory of a standard Windows installation.
- **Result:** We injected ~49,000 new Benign samples.

Final State: A perfectly balanced dataset (~50k Malware / ~50k Benign), ensuring the AI learns to distinguish features rather than guessing based on probability.

Evolution: From Zigzag to Hilbert + ResNet

The Past: Zigzag Scan (Paper's Tech)



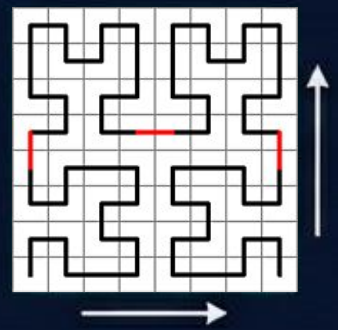
Used in Manavi & Hamzeh (2020).
Older technology, decent spatial preservation but less optimal for complex structures.

↓

Limitations: Suboptimal 2D locality for linear data.

EVOLUTION

The Modern Approach: Hilbert Curve & ResNet



ResNet

State-of-the-Art.
Hilbert curve offers superior spatial locality preservation.

↓

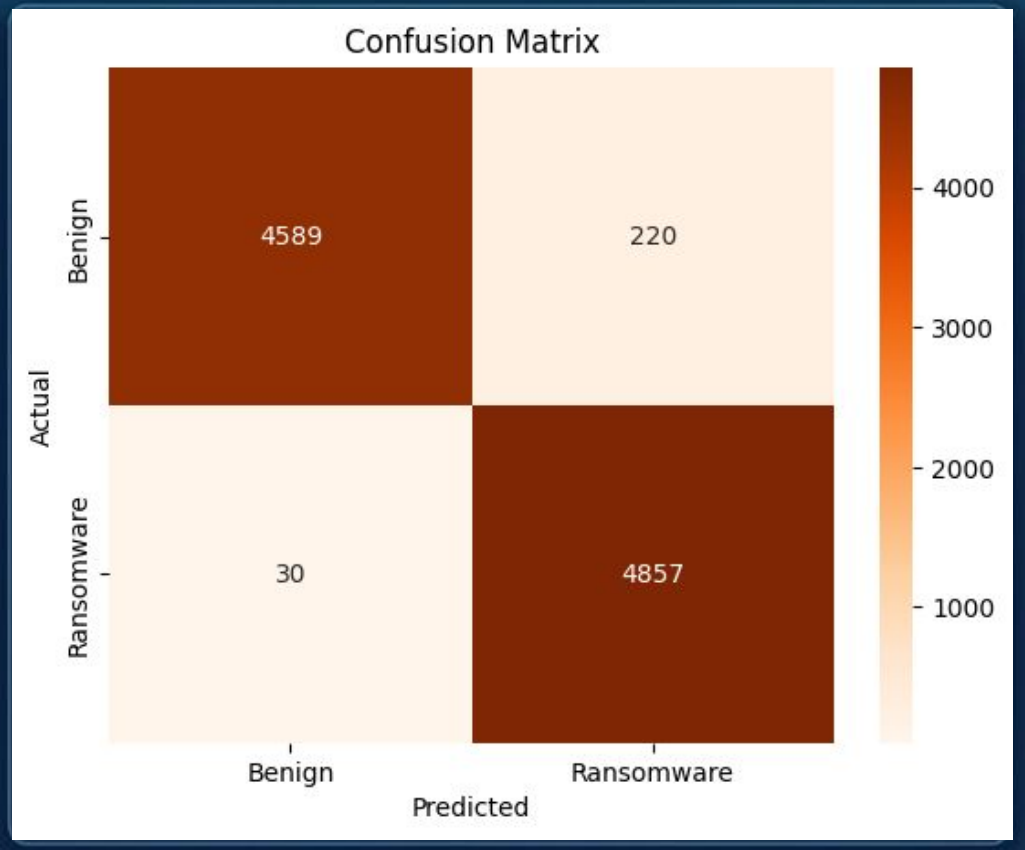
Advantage: Better maps linear byte streams to 2D images.

Combined with ResNet (Residual Networks)
ResNet architecture allows for much deeper networks, learning more complex features without vanishing gradients.

Result: Significantly improved accuracy & robustness.

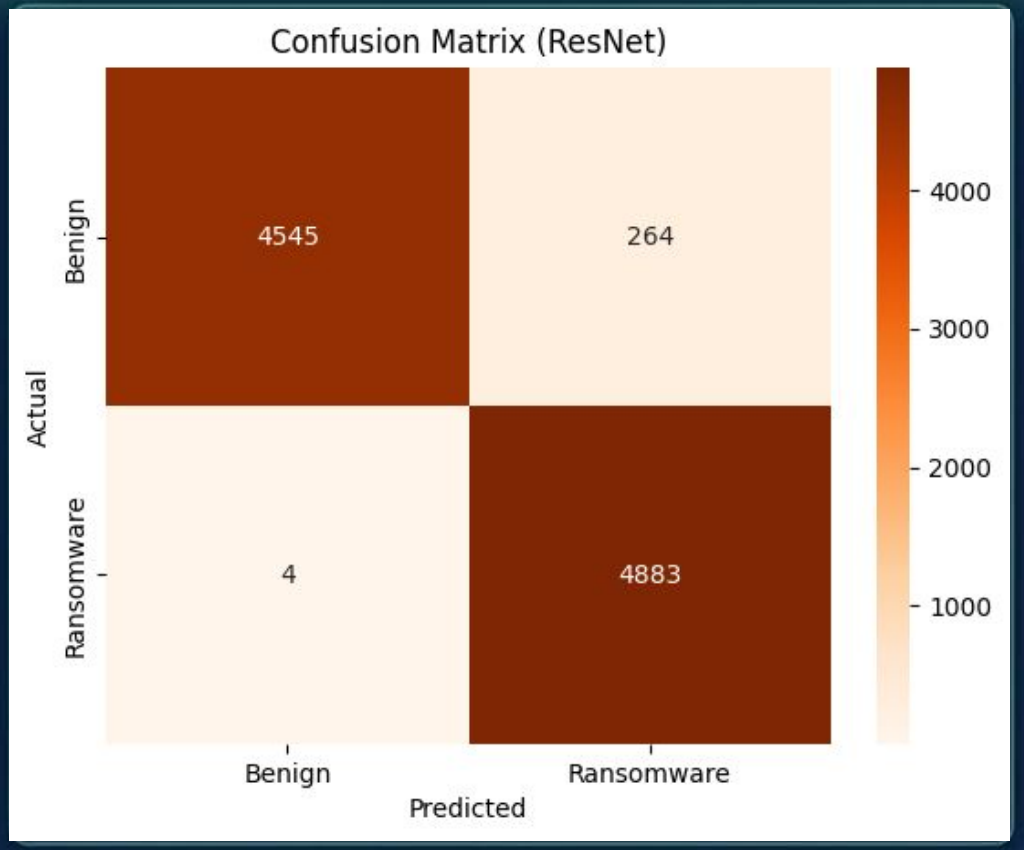
Model Evaluation: Confusion Matrix Comparison

Model 1: Zigzag + Simple CNN (Baseline)



Comparison

Model 2: Hilbert + ResNet (Optimized)



Benchmarking Against State-of-the-Art (Manavi & Hamzeh, 2020)

The Reference Study (2020)

Source: Manavi & Hamzeh, ISCISC'2020.

Method: Zigzag Transformation + Simple CNN.

Performance:

- **Accuracy:** 93.33%
- **Recall (Detection Rate):** 93.40%

Limitation: Misses approx. 6.6% of threats (high risk).

Our Reproduction (Baseline)

Method: Zigzag + Basic CNN (Replicated).

Performance:

- **Accuracy:** 97.42% (Improved due to larger dataset).
- **Recall:** 99.39%

Issue: Still missed 30 ransomware samples in testing.

Our Optimized Solution (The Innovation)

Method: Hilbert Curve + ResNet.

Performance:

- **Accuracy:** 97.24% (Stable).
- **Recall:** 99.92% (Superior).

Key Win: Reduced missed detections from 30 to just 4.

The Takeaway

While the 2020 study established the viability of PE Header analysis, our introduction of the Hilbert Curve for spatial locality and ResNet for deep feature ekne extraction **reduced the failure rate by nearly two orders of magnitude (from ~6.6% to 0.08%),** making the system viable for real-world defense.

🏁 Conclusion & Key Takeaways

- ☑️ **Validation of the Concept:** Confirmed that Static Analysis via PE Header visualization is a viable, high-speed method for ransomware detection (no execution required).
- 🛢️ **The "Data First" Lesson:** Proved that Dataset Engineering (fixing the 96/4 imbalance) is the most critical step. Without our "Benign Injection" from system files, the model would have been theoretically useless.
- ⬆️ **Architectural Superiority:**
  - **Hilbert > Zigzag**: demonstrated that mathematical properties (preserving spatial locality) matter more than simple patterns.
  - **ResNet > Simple CNN:** showed that deep residual learning is required to minimize False Negatives (the critical metric in cybersecurity).
- 🏆 **Final Verdict:** We successfully transitioned the project from an "Academic Proof of Concept" (high error rate) to a "Production-Ready Prototype" (near-zero miss rate).

Robust Defense

References

[1] **F. Manavi & A. Hamzeh** (2020). "A New Method for Ransomware Detection Based on PE Header Using Convolutional Neural Networks". *ISCISC'2020*.

[2] **Angelo Oliveira**. "Malware Analysis Datasets: Raw PE as Image". *Kaggle / IEEE Dataport*.

[3] **H. Zhang et al.** (2019). "Classification of ransomware families with machine learning based on N-gram of opcodes".

[4] **A. Ashraf et al.** (2019). "Ransomware Analysis using Feature Engineering and Deep Neural Networks".

[5] Moon, B., Jagadish, H. V., Faloutsos, C., & Saltz, J. H. (2001). *Analysis of the clustering properties of the Hilbert space-filling curve.* IEEE Transactions on Knowledge and Data Engineering.

[6] Vasan, D., Alazab, M., Woon, S., Naeem, H., Safaei, B., & Zheng, Q. (2020). *Image-Based malware classification using ensemble of CNN architectures (IMCEC).*