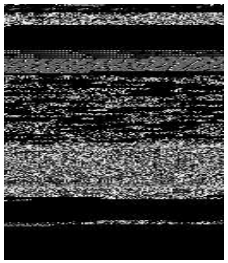
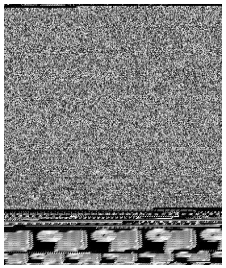
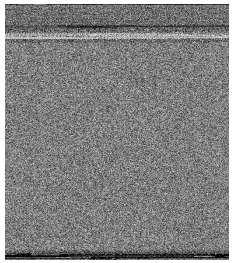
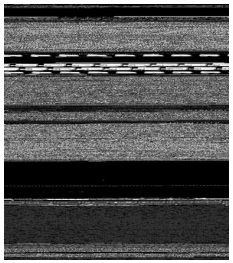
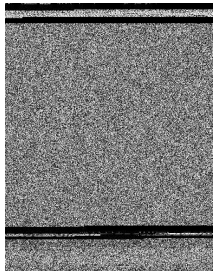
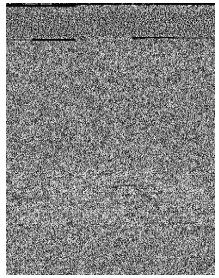
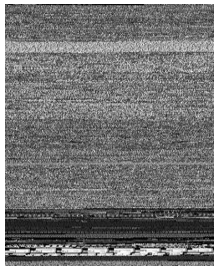
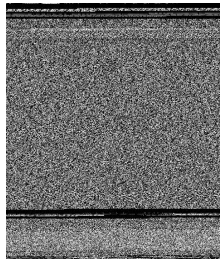
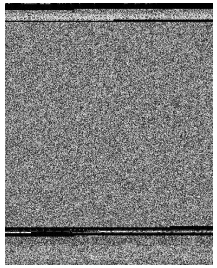
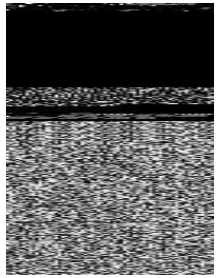
Malware Images: Visualization and Automatic Classification

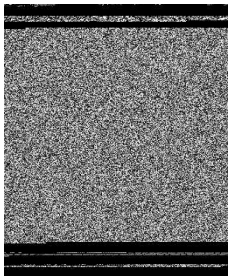
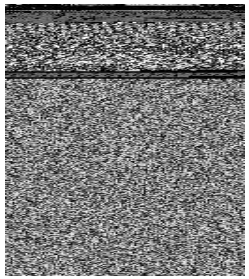
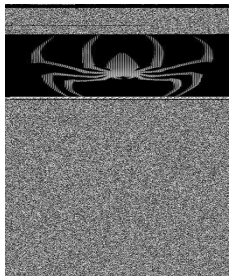
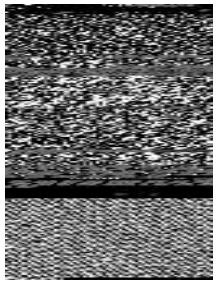
Lakshmanan Nataraj
Vision Research Lab
University of California, Santa Barbara

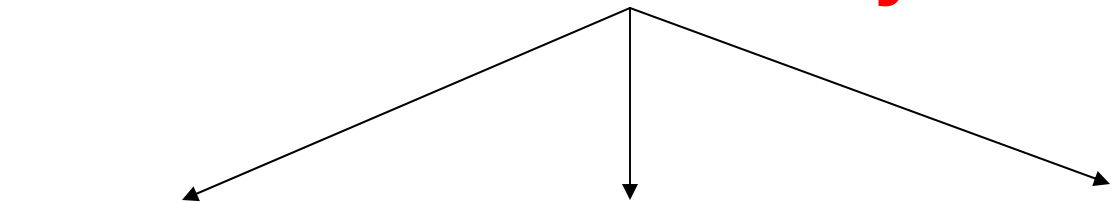






Malware Images

http://vision.ece.ucsb.edu/~lakshman/Malware%20Images/album/index.html

Malware Analysis



Static Analysis

Analyze the code and build control flow graphs

Suffers from code Obfuscation

Dynamic Analysis

Execute the malware in a virtual environment and analyze its execution trace (behavior analysis)

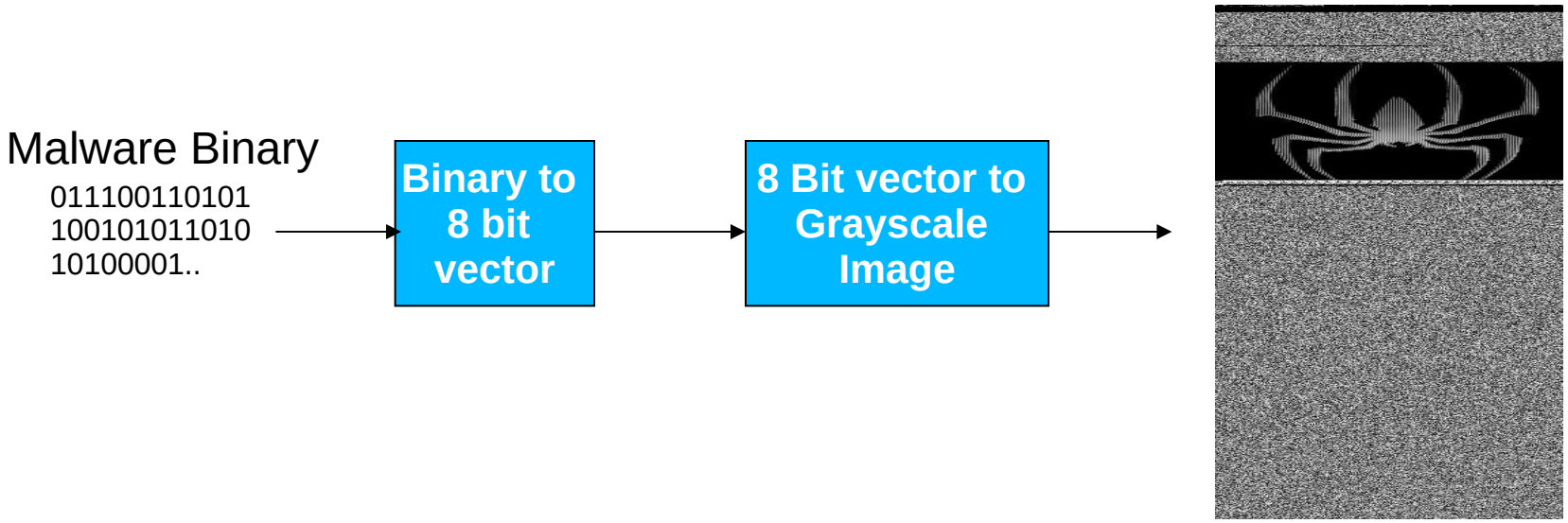
Promising but complex and time consuming (few seconds to several mins!)

Alternative Ways

Analyze Raw binaries and build signature based on n-grams

Doesn't give much information

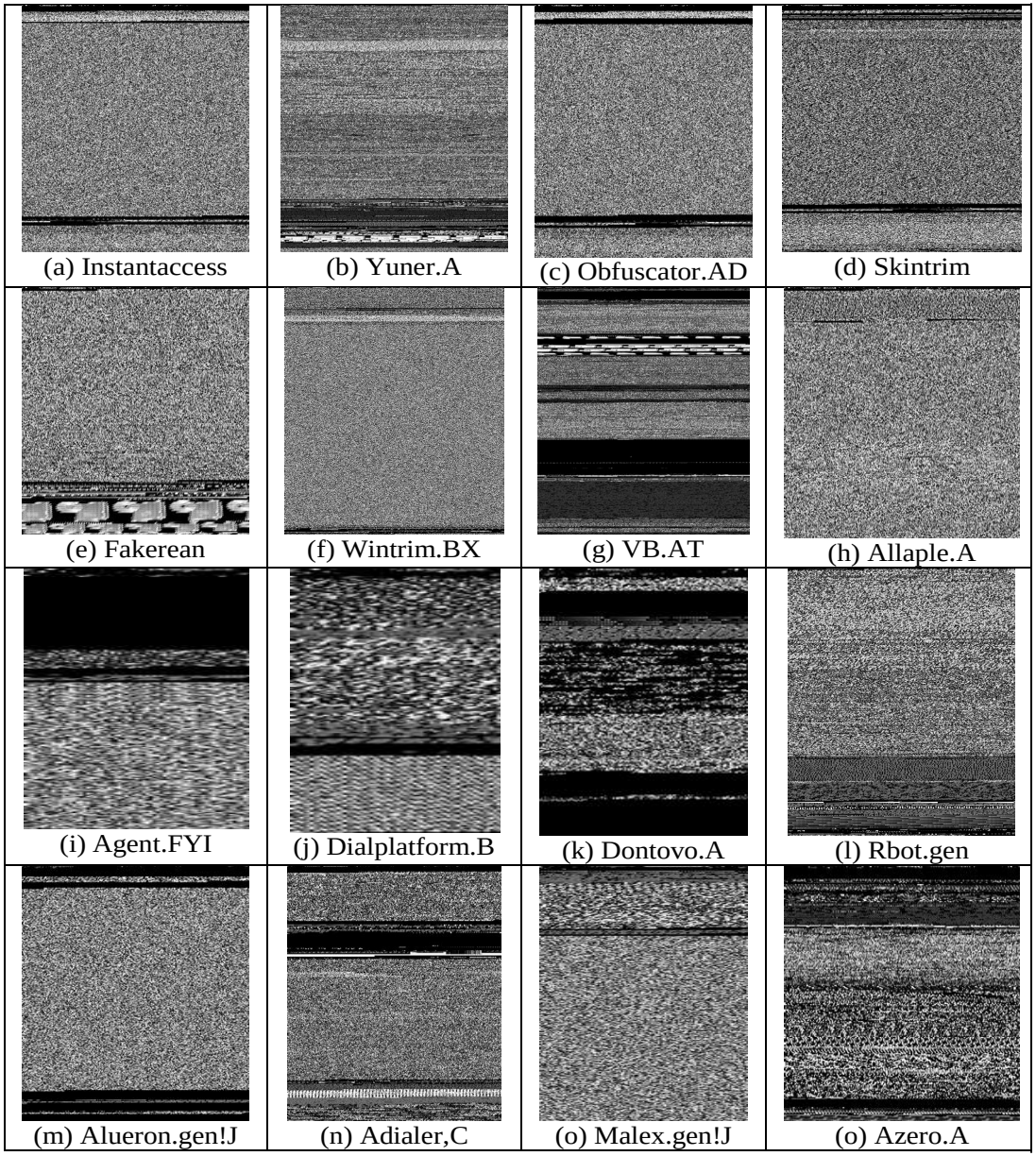
Malware Images: The Next Alternative



Why Images?

- Different sections of a binary can be easily seen when viewed as an image
  - » **VISUALIZATION**
- Malware coders change small parts of the original source code to produce a new variant.
- Images can capture small changes yet retain the global structure.
- Hence, malware variants belonging to the same family appear very similar as images. These images are also distinct from images of other malware families.
  - » **CLASSIFICATION / CLUSTERING using Image Processing Features**

Malware Images of Various Families

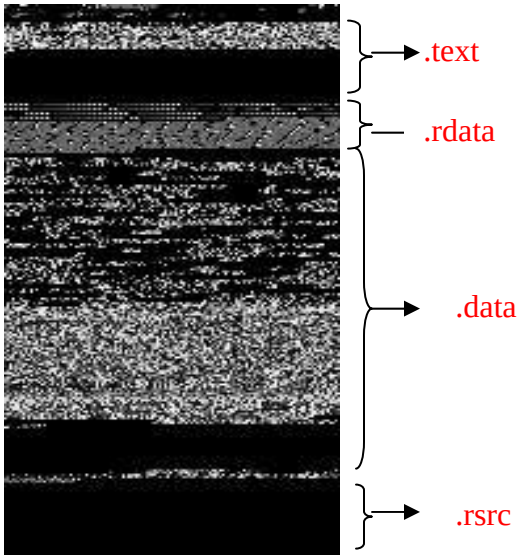


http://vision.ece.ucsb.edu/~lakshman/Malware%20Images/album/index.html

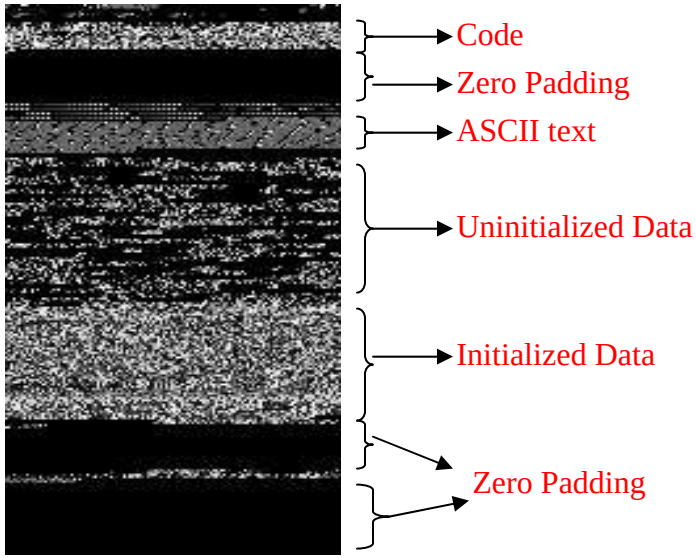
Information from Images

Images give more information about the structure of the malware. We can see that various subsections have different texture. The entire structural layout can also be seen.

Sections obtained from *pefile**



Information that we can obtain from images



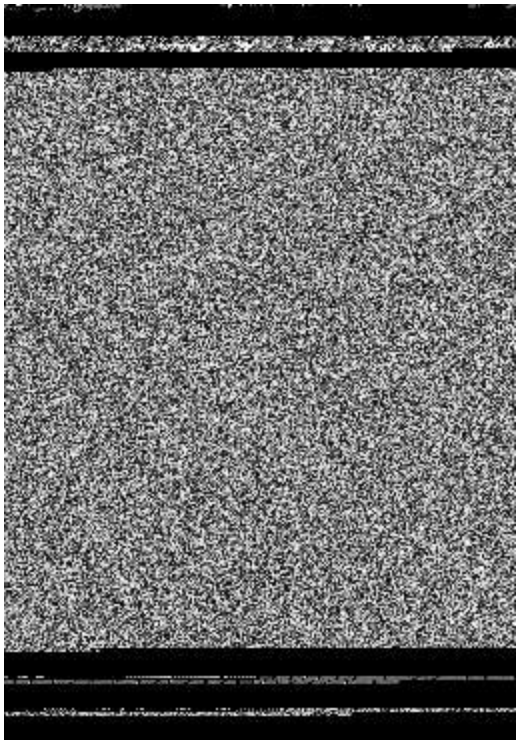
**code.google.com/p/pefile/*

How to choose Image width?

- Width of the image is according to the file size based on visual experiments.
- Height of the image varies depending on the file size.

File Size Range	Image Width
<10 kB	32
10 kB – 30 kB	64
30 kB – 60 kB	128
60 kB – 100 kB	256
100 kB – 200 kB	384
200 kB – 500 kB	512
500 kB – 1000 kB	768
>1000 kB	1024

Example: Variant1



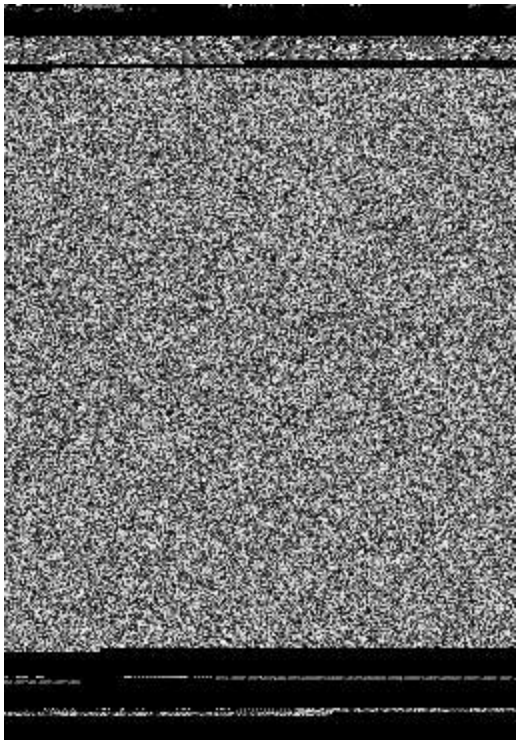
Alueron.gen!J

Dialplatform.B

Agent.FYI

Lolyda.AT

Variant2



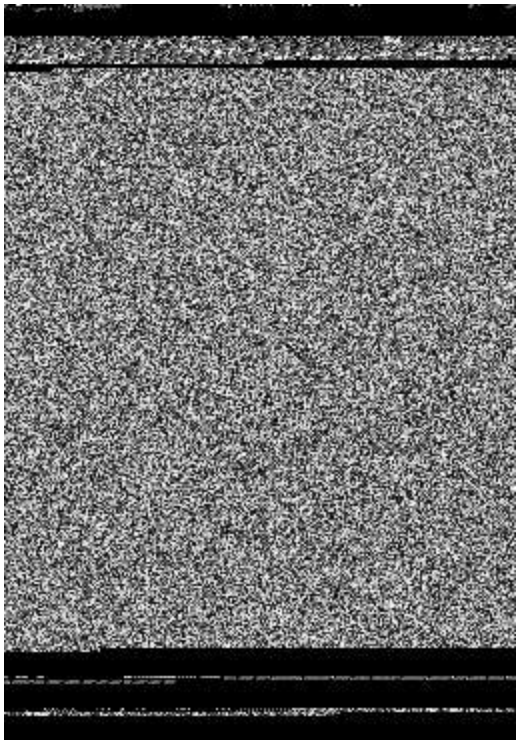
Alueron.gen!J

Dialplatform.B

Agent.FYI

Lolyda.AT

Variant3



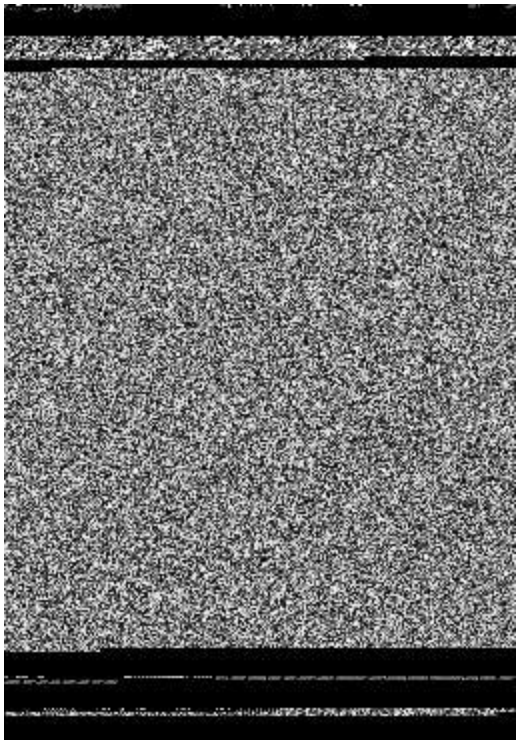
Alueron.gen!J

Dialplatform.B

Agent.FYI

Lolyda.AT

Variant4



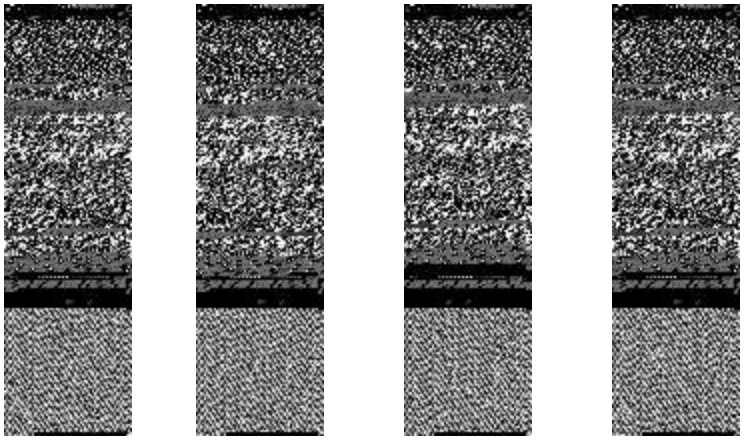
Alueron.gen!J

Dialplatform.B

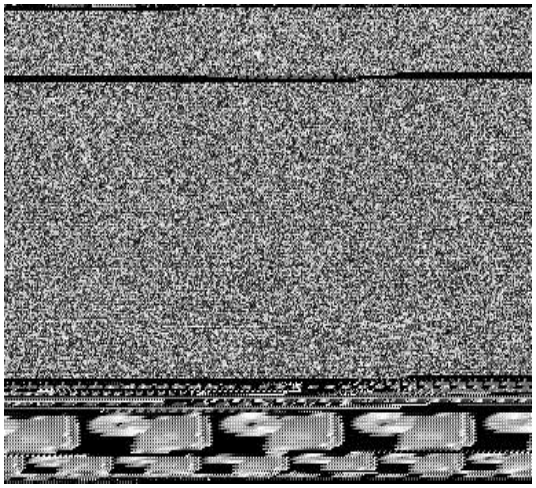
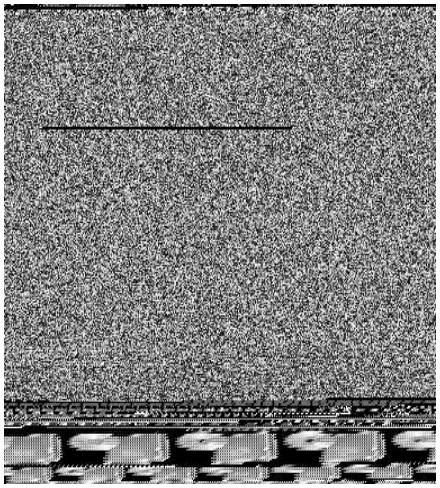
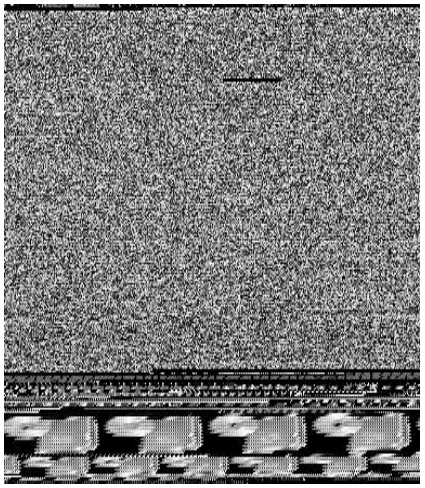
Agent.FYI

Lolyda.AT

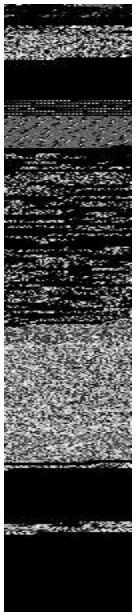
All Variants of Dialplatform.B



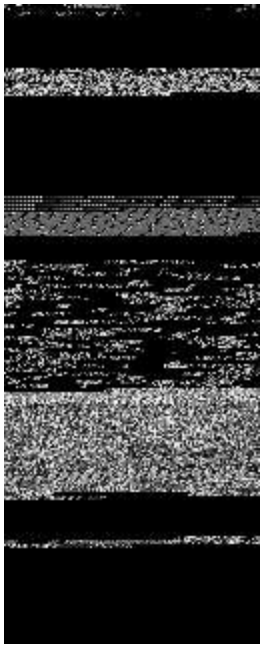
More Examples of Malware Images

Rogue: FakeRean



Although the file size varies, the overall structure is visible from the images

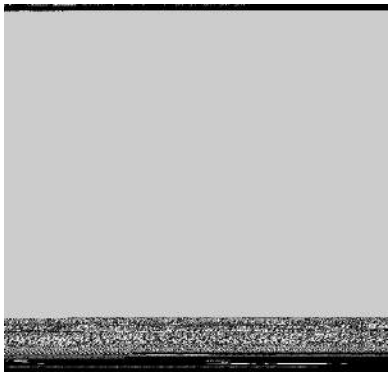



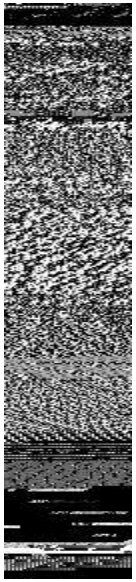
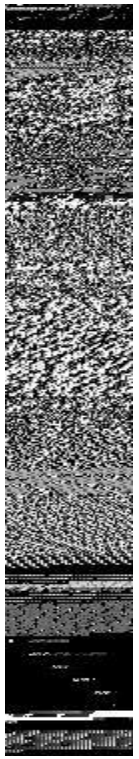
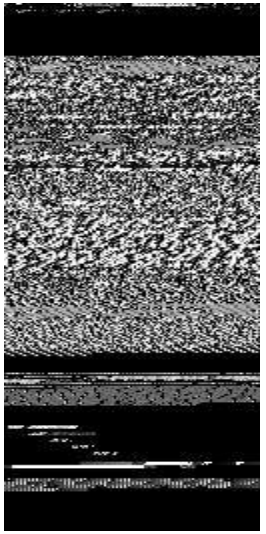
TrojanDownloader: Dontovo.A

New Naming Schemes

The following instances of malware were named by *Microsoft Security Essentials* as Lolyda.AA. But clearly, they can be subdivided into 3 sub-categories based on image properties





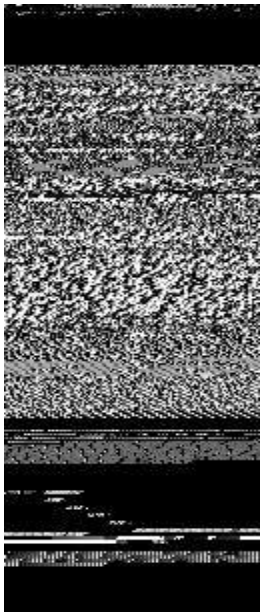


Image Analysis for Similarity

- **Once the malware is converted to an image representation, image based features can be computed to characterize a malware.**

- **We use a feature based on image texture which is commonly used in scene category classification such as coast, mountain, forest, street, etc.**

- **Here, instead of scene categories, we have malware families.**

Texture Features

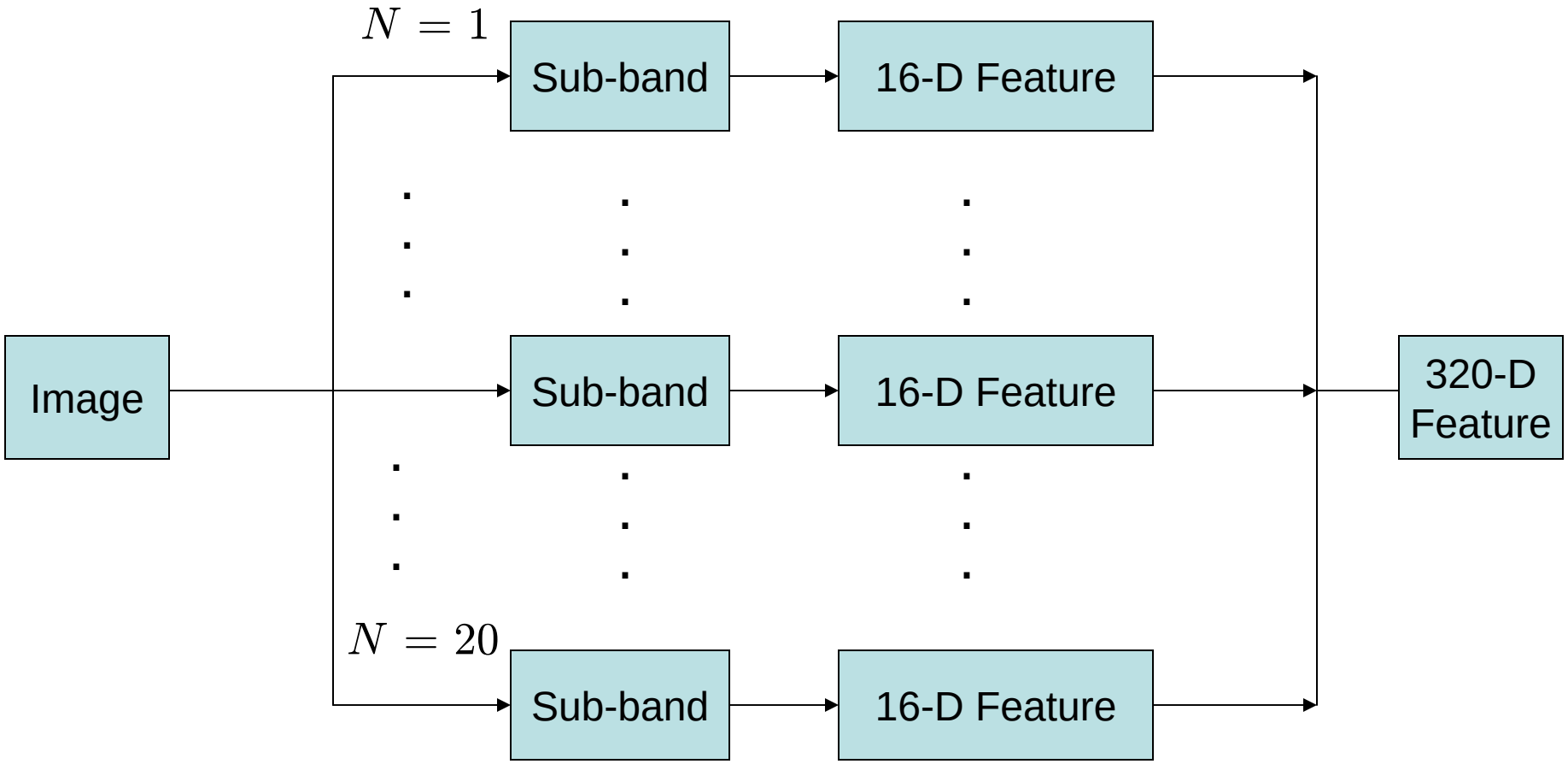
- **Every image location is represented by the output of filters tuned to different orientations and scales.**
- **A steerable pyramid of 4 scales and 8 orientations is used.**
- **The local representation of the image is then given by:**

$$v^L(x) = \{v_k(x)\}_{k=1,N}$$

where N is the number of sub-bands.

- **The global features are then averaged:** $m(x) = \sum_{x'} |v(x')| w(x'-x)$
- **Then they are down-sampled to a 4x4 resolution.**

GIST Feature Computation



Classifier

- **Classification: k-nearest neighbors (k-nn)**
  - A test sample is classified as belonging to Family i if it has k nearest neighbors in the feature space belonging to Family i.

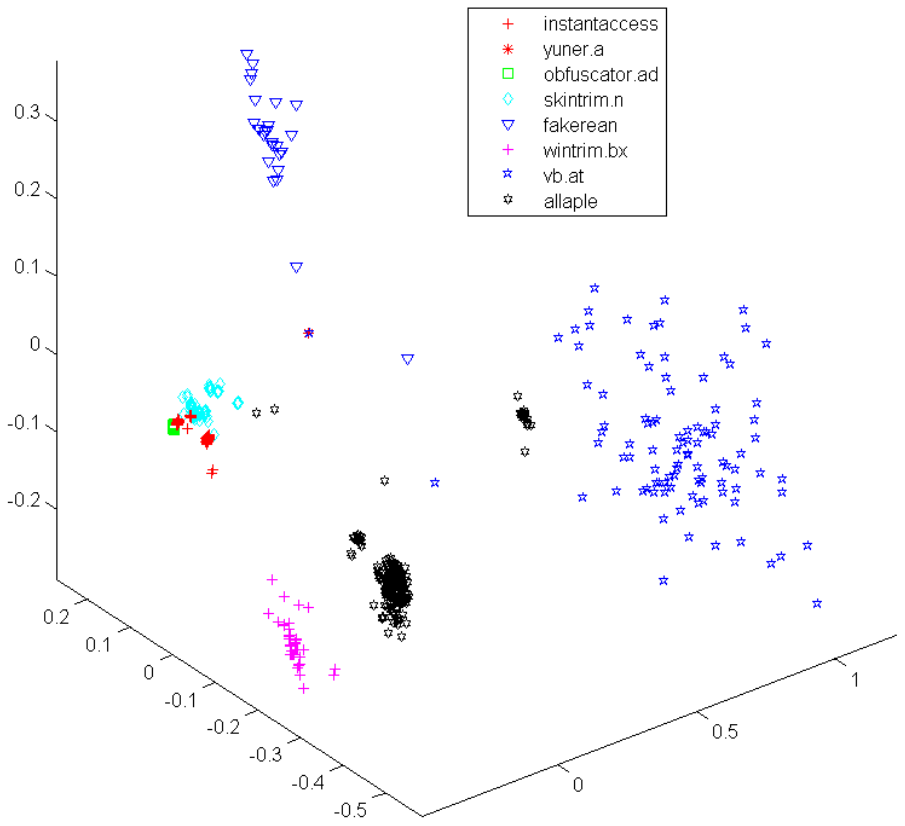
- **Distance Measure: Euclidean distance**
  - To measure the distance in the feature space, we use Euclidean Distance as the distance measure.

- **10-Fold cross validation.**

Preliminary Classification Results on Image Based Signatures

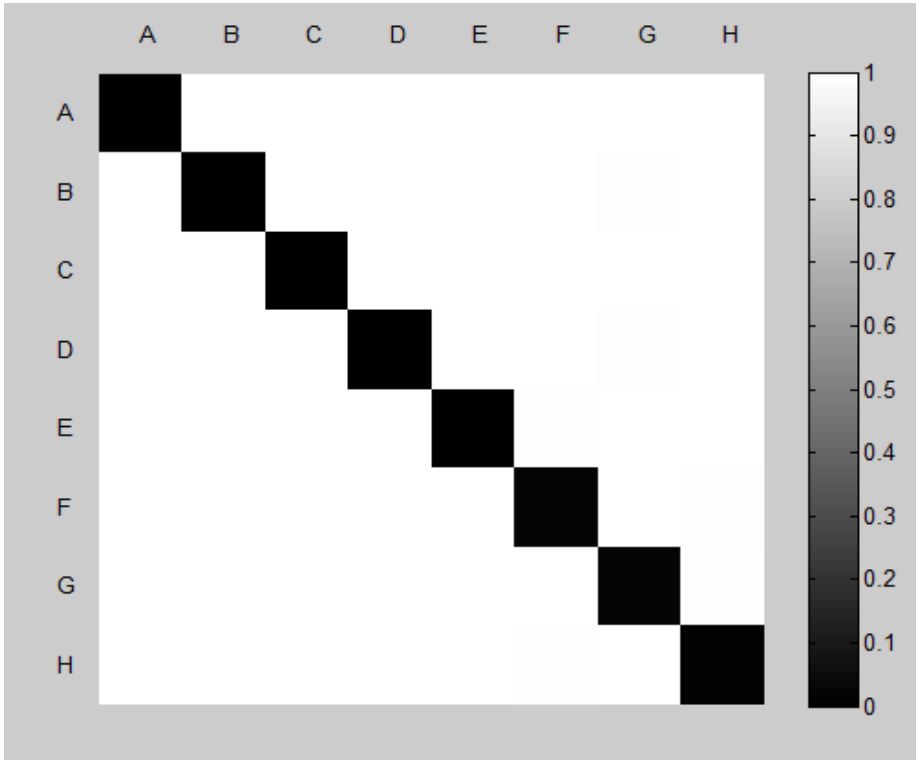
- 2000 malware comprising 8 malware families were converted to digital images[1].
- Image Texture based Features (320 dims) were computed on the images.
- k-nn classifier (k=3) yielded a classification accuracy of 98%.

Low Dimensional Mapping of Image based Features on 8 Malware Families

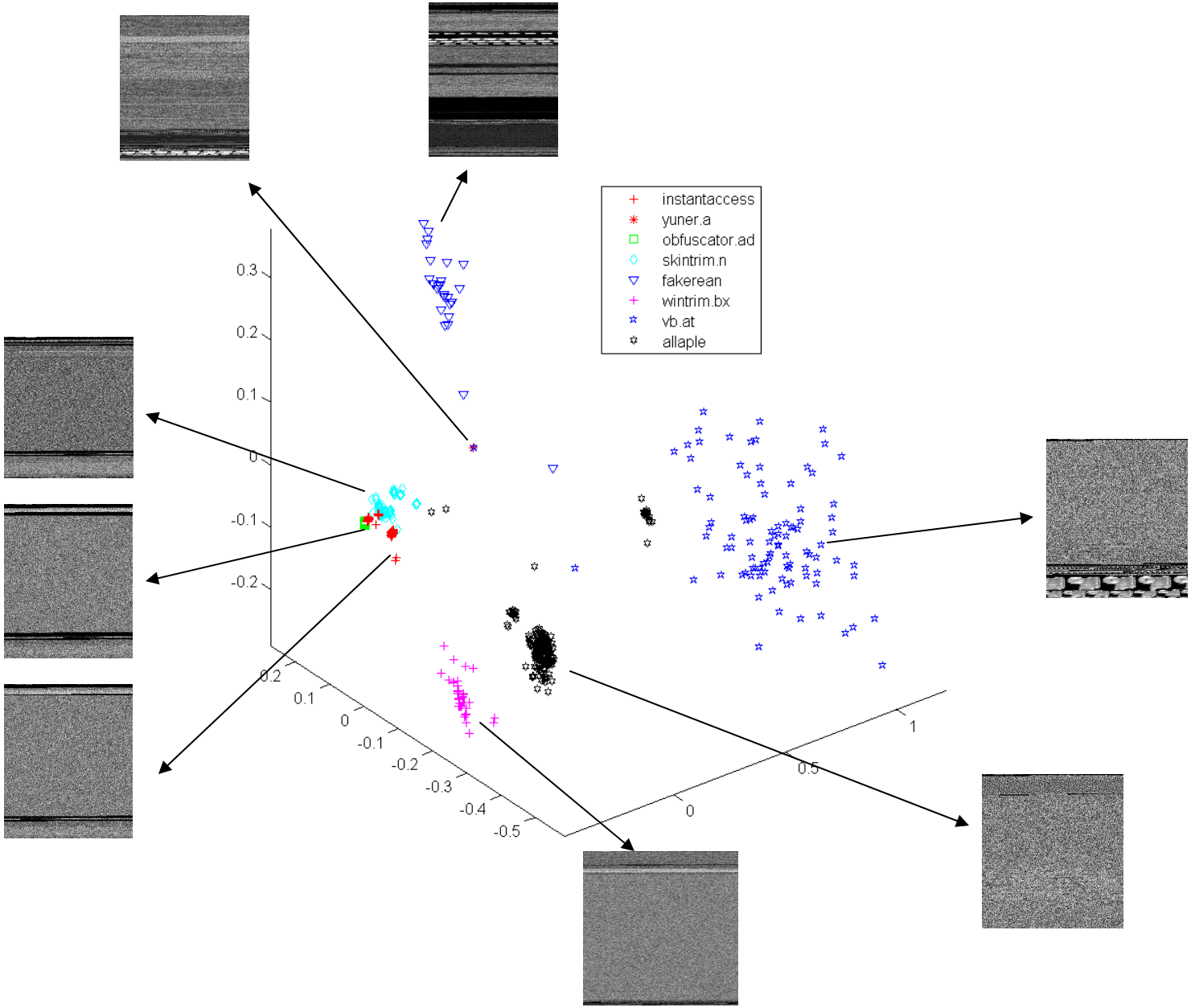


[1]Malware obtained from Anubis (anubis.iseclab.org) and named using Microsoft Security Essentials

Confusion Matrix – No Confusion (almost)



A Closer Look



What about Packing?

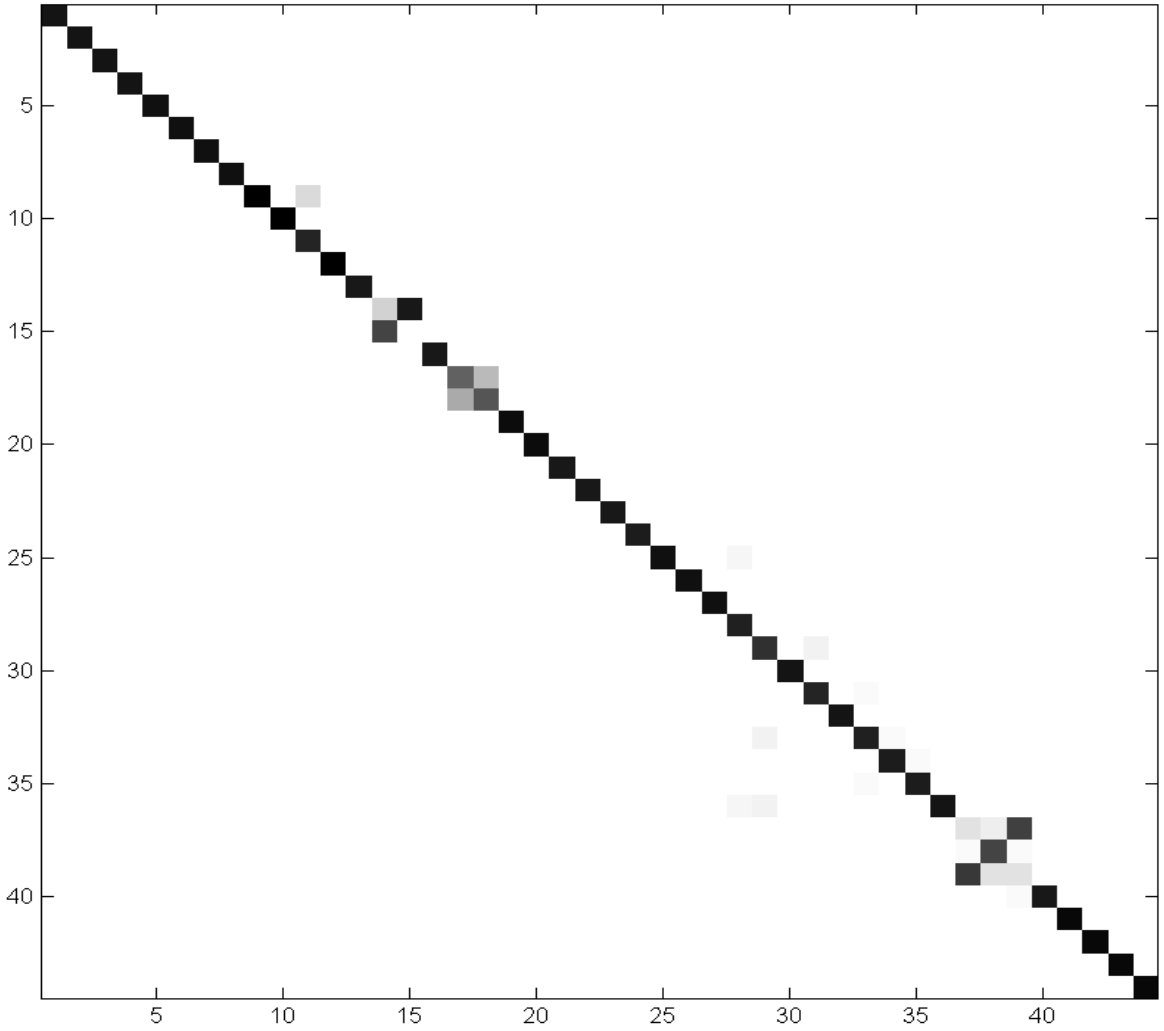
- **Packing transforms a binary to a completely different form.**
- **Hence, the image after packing “usually” appears completely different.**
- **A common misconception is that if two binaries belonging to different families are packed using the same packer, they will appear the same.**
- **However, this is not the case. We did a test to verify this.**

Test with Packed Executables

- **Unpacked malware from 11 families packed with UPX, Winupack and PeCompact.**
- **The packed malware were treated as new families.**
- **The total number of families were now 44 (including unpacked).**
- **The classification experiments were run again.**

Adialer.C
Adpclient
Agent.dz
Browsermodifier.cnnicc
Dontovo.A
Lolyda.AA
Lowsones.gen!B
Rbot.gen
Rootkit.gen!C
Vb.at
Yuner.A

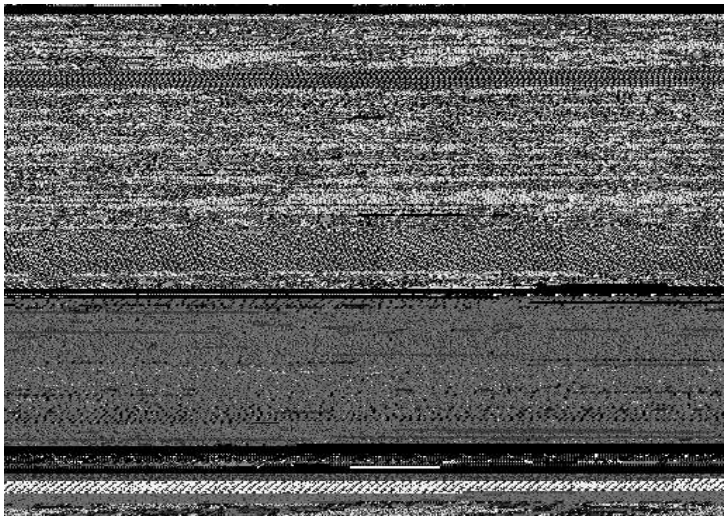
Confusion Matrix for Packing Test



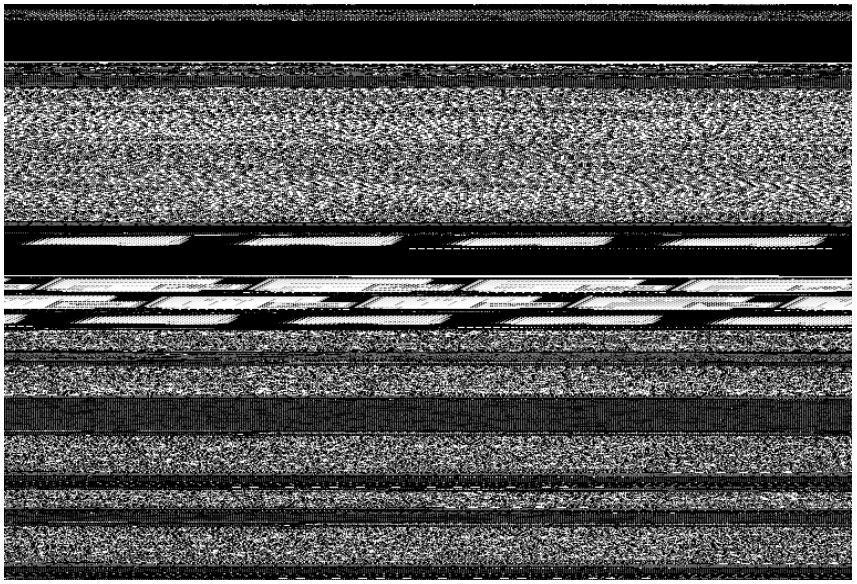
Confusion only within families, that too for malware whose compression ratio is less

Effect of Packing

Before Packing



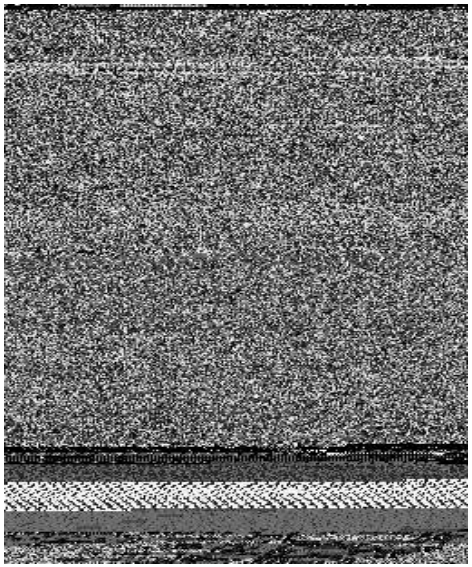
Adialer.C

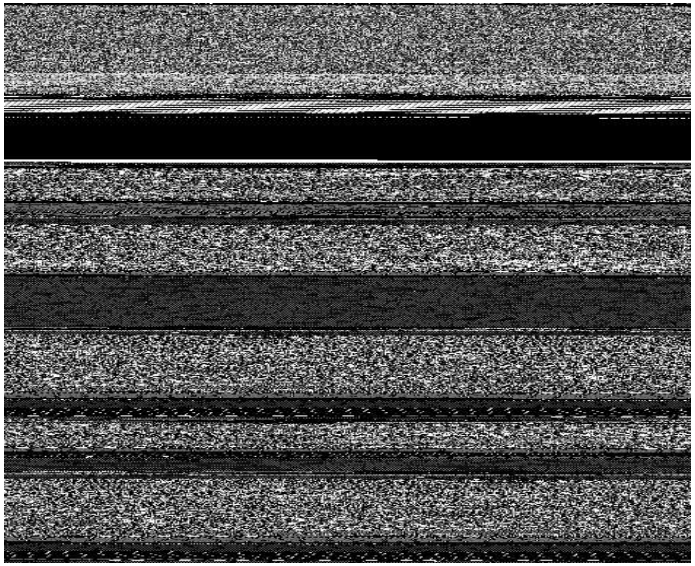


VB.AT

The relationships between a packed malware and an unpacked malware can be analyzed.

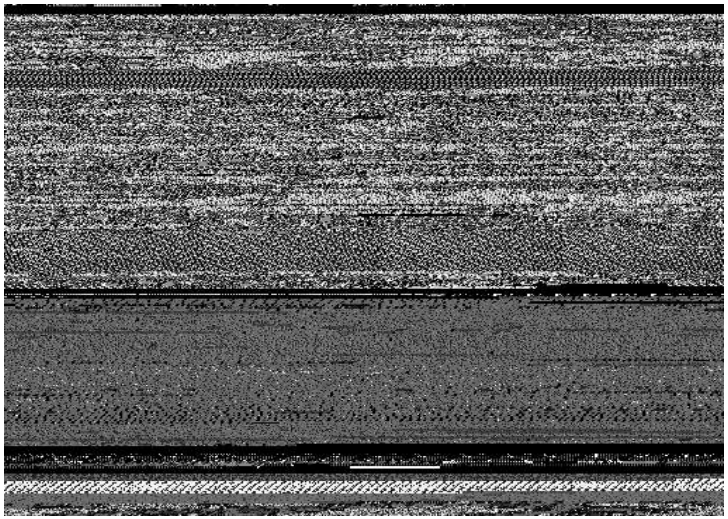
After Packing (UPX)





Effect of Packing

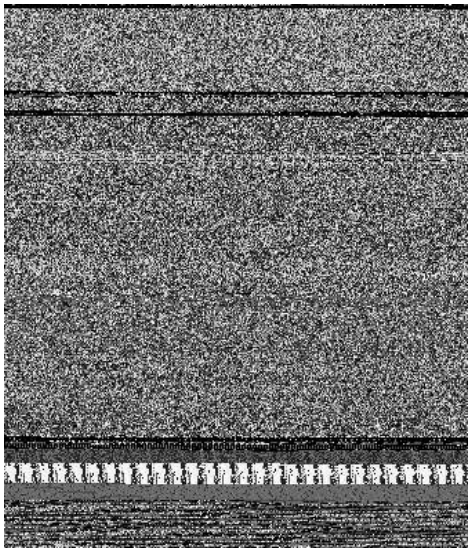
Before Packing

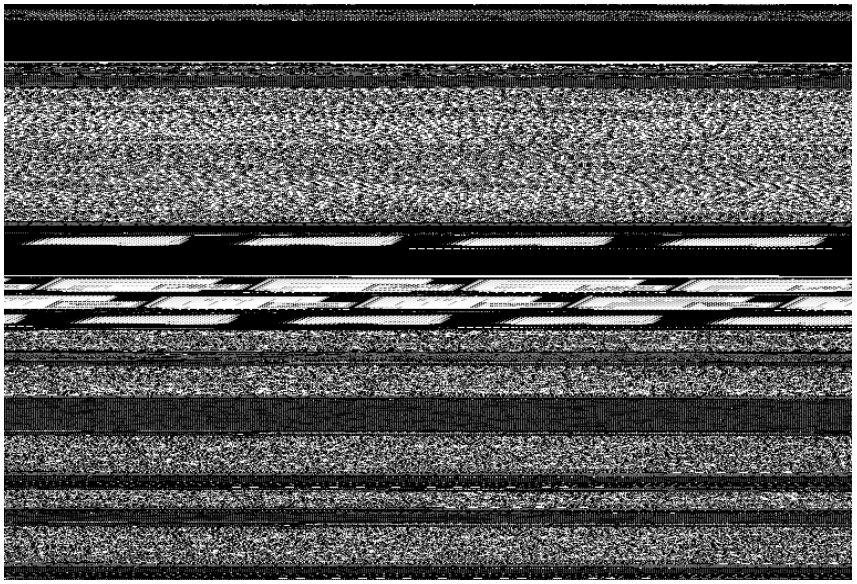


Adialer.C

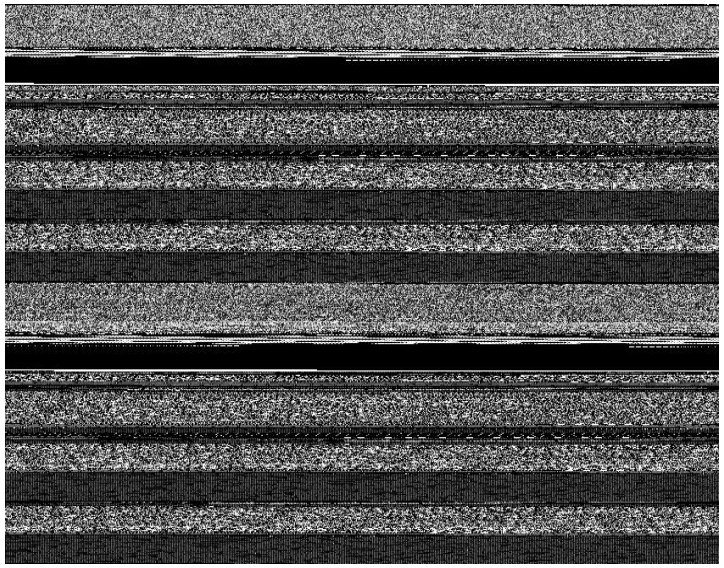
The relationships between a packed malware and an unpacked malware can be analyzed.

After Packing (PeC)



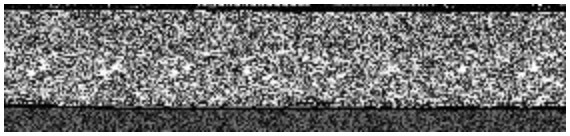


VB.AT

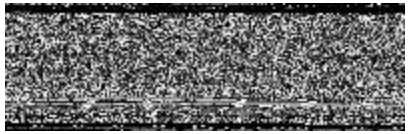


Dontovo.A after UPX

Agent.DZ after UPX



Lolyda.AA after UPX



Analysis on Packed Executables

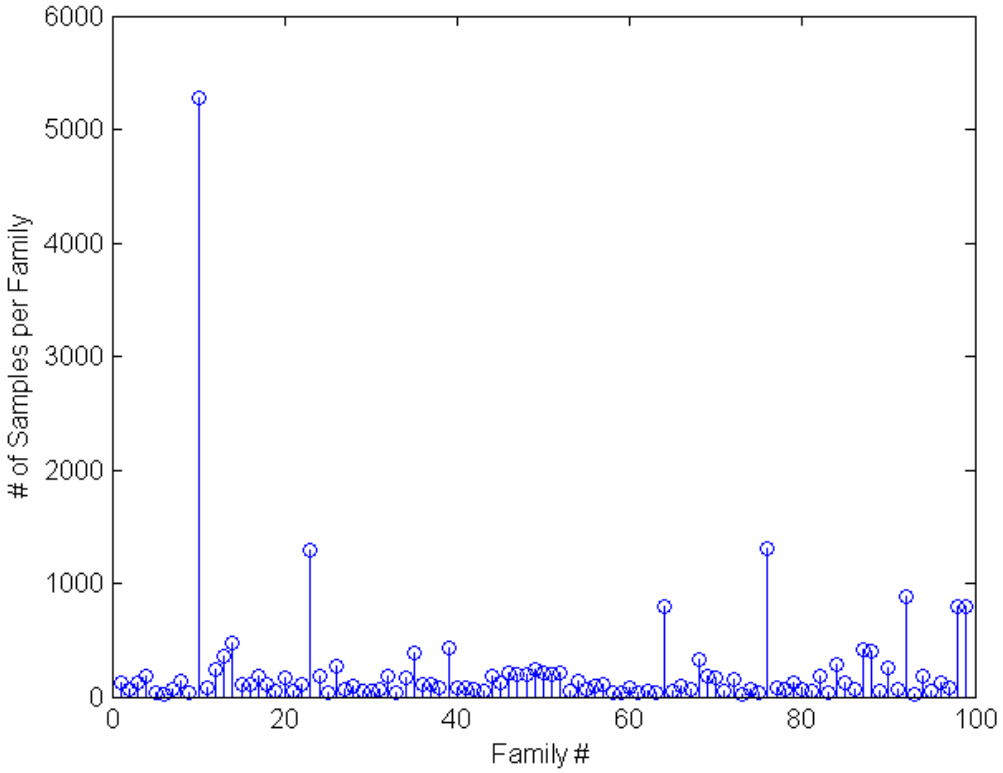
- From preliminary analysis, we observed that:
  - When an unpacked malware family with several similar variants are packed with a specific packer, then the images of the newly packed malware (of same family) are also similar.
  - They are similar “within themselves” if the compression ratio is high.
  - If the compression ratio is low, then they are similar to the original unpacked malware family.
- We are currently doing a more thorough analysis to support our claim.

Large Scale Experiments

- 25k malware from Anubis and VxHeavens Dataset.
- Families labeled using Microsoft Security Essentials
- Top 100 families chosen.

Some Dataset Logistics

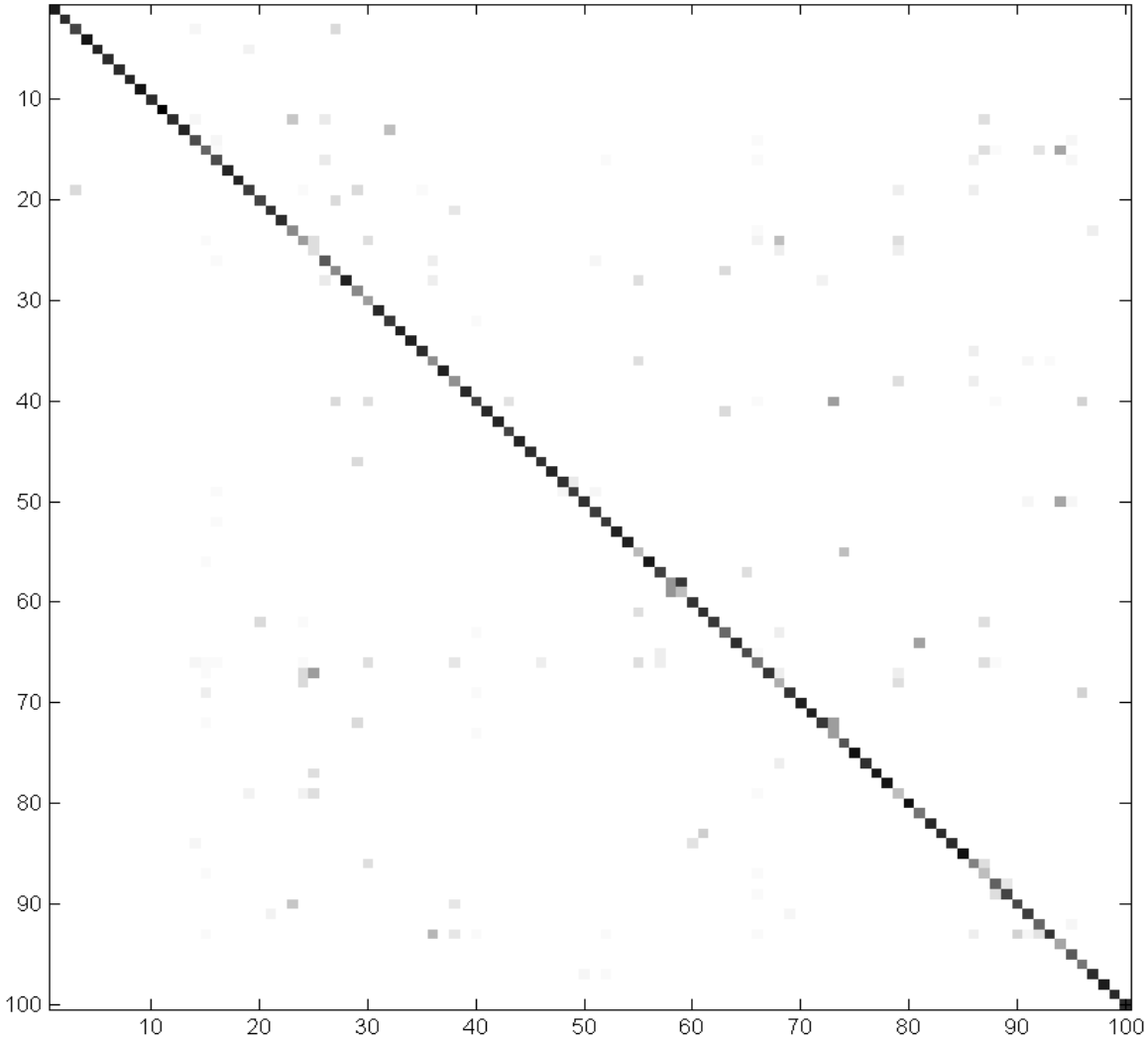
of samples per family



Top 11 Families

Allaple.A
Alueron.gen!j
Browsermodifier.cnnic
Instantaccess
Pcclient.bx
Seimon.D
VB.AT
VB.AT UPX
Vundo.gen!r
Yuner.A
Yuner.A UPX

Confusion Matrix for classification on 100 families



k-nn = 3, 100 families

Families with High Accuracy

Family Name	*No. of samples*
Instantaccess	431
Adialer.C	63
Adialer.G	40
Adpclient	29
Agent.Dz	63
Agent.Fyi	140
Agent.Wx (FSG)	41
Cnnic	1287
Dontovo.A	162
Hupigon.gen!A	114

Accuracy does not depend on number of samples per family

Screenshot of a family with high accuracy

Browsermodifier.cnnicc

The images are rotated 90 deg

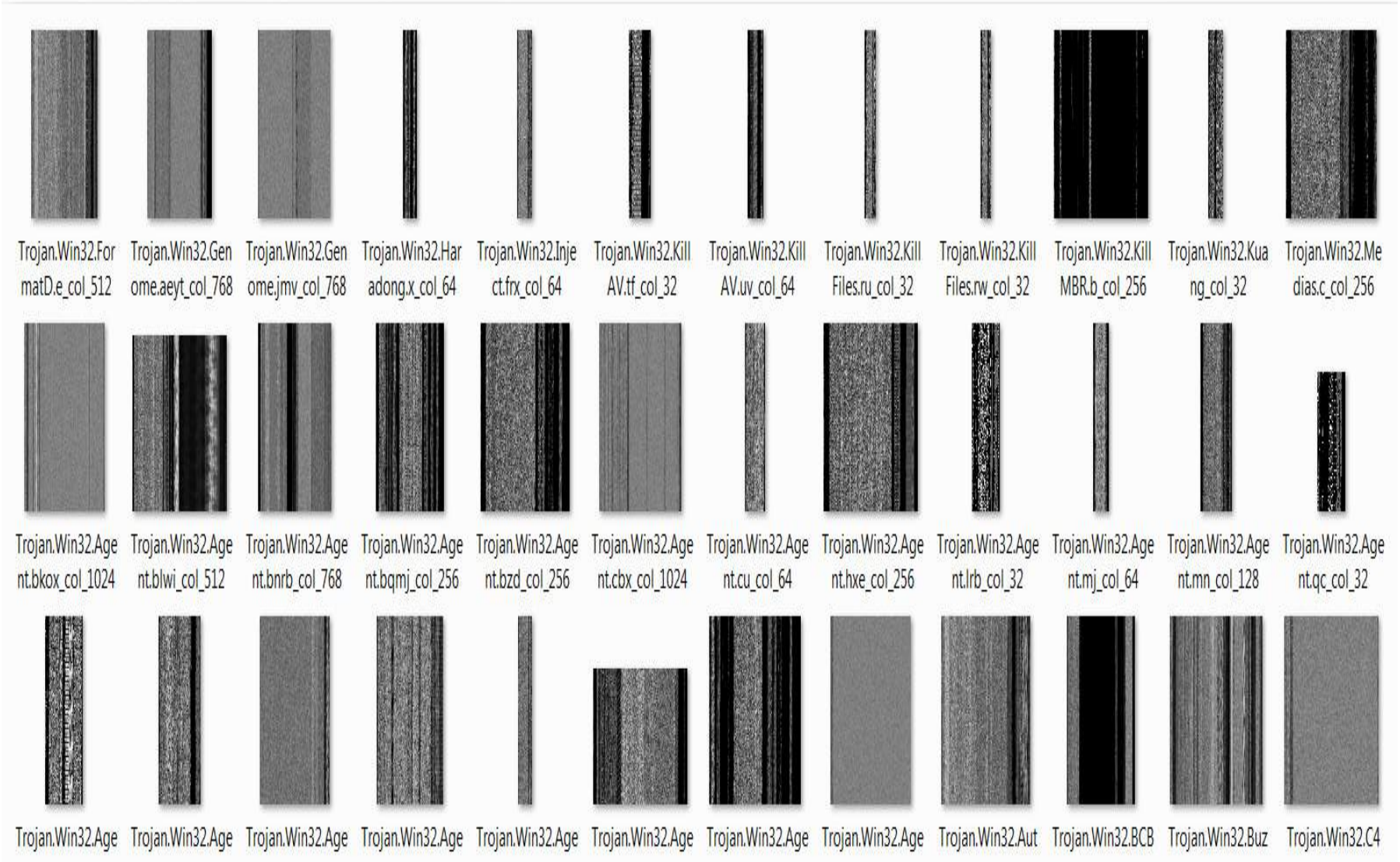
Families with Low Accuracy

Family Name	*No. of samples*
Orsam!rts	56
Malex.gen!j	215
Bumat!rts	188
Backdoor.Agent	189
Pakes	37
Swizzor.gen!k	127
Poison.G	59
C2lop.O	64
Ceeinject.gen!j	54
Trufip!rts	117

Screenshot of a family with low accuracy

Orsam!rts

The images are rotated 90 deg

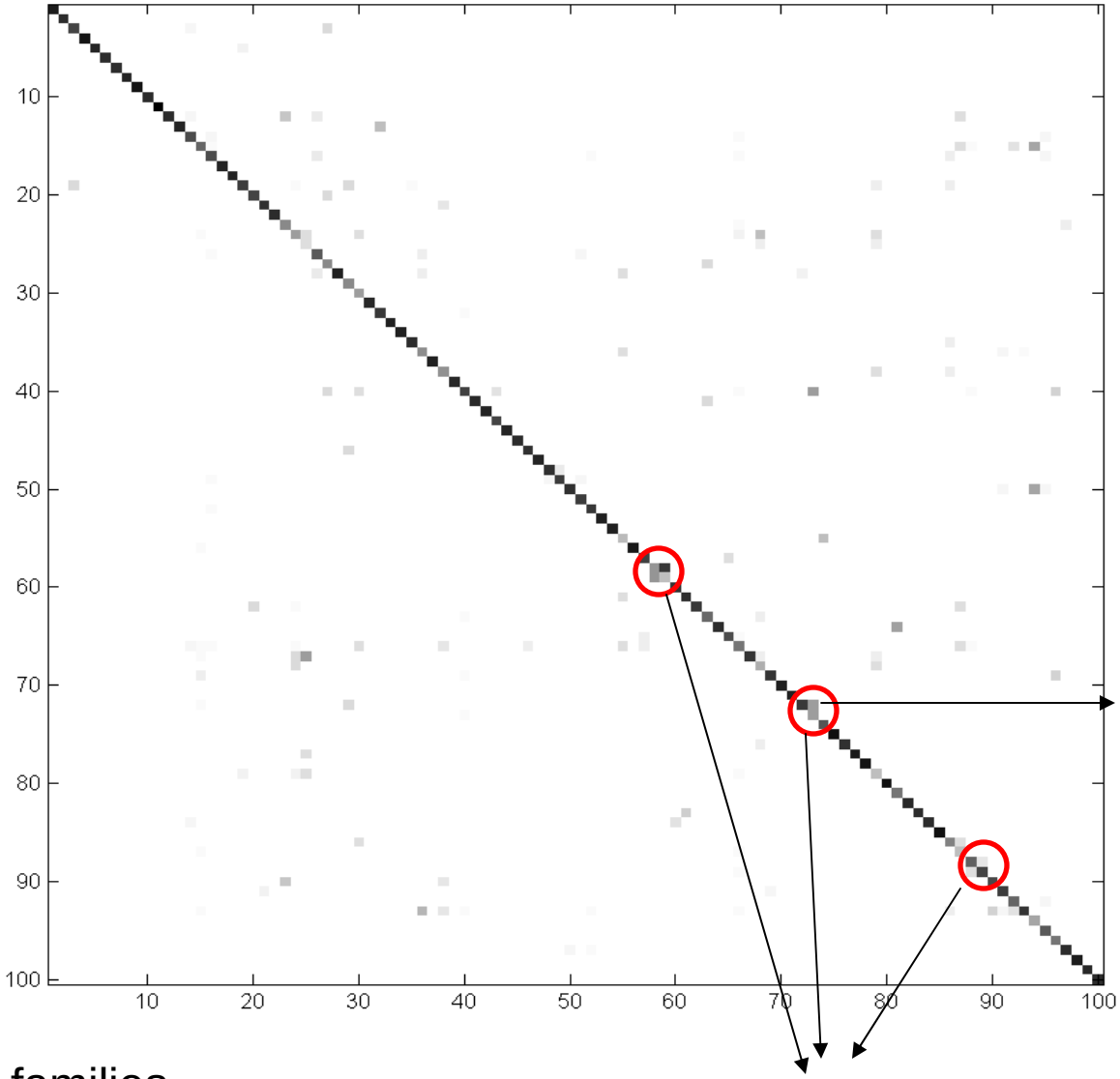


The disparity among the malware images could be due to the AV Software.

Stats on Orsam!rts - MIXED

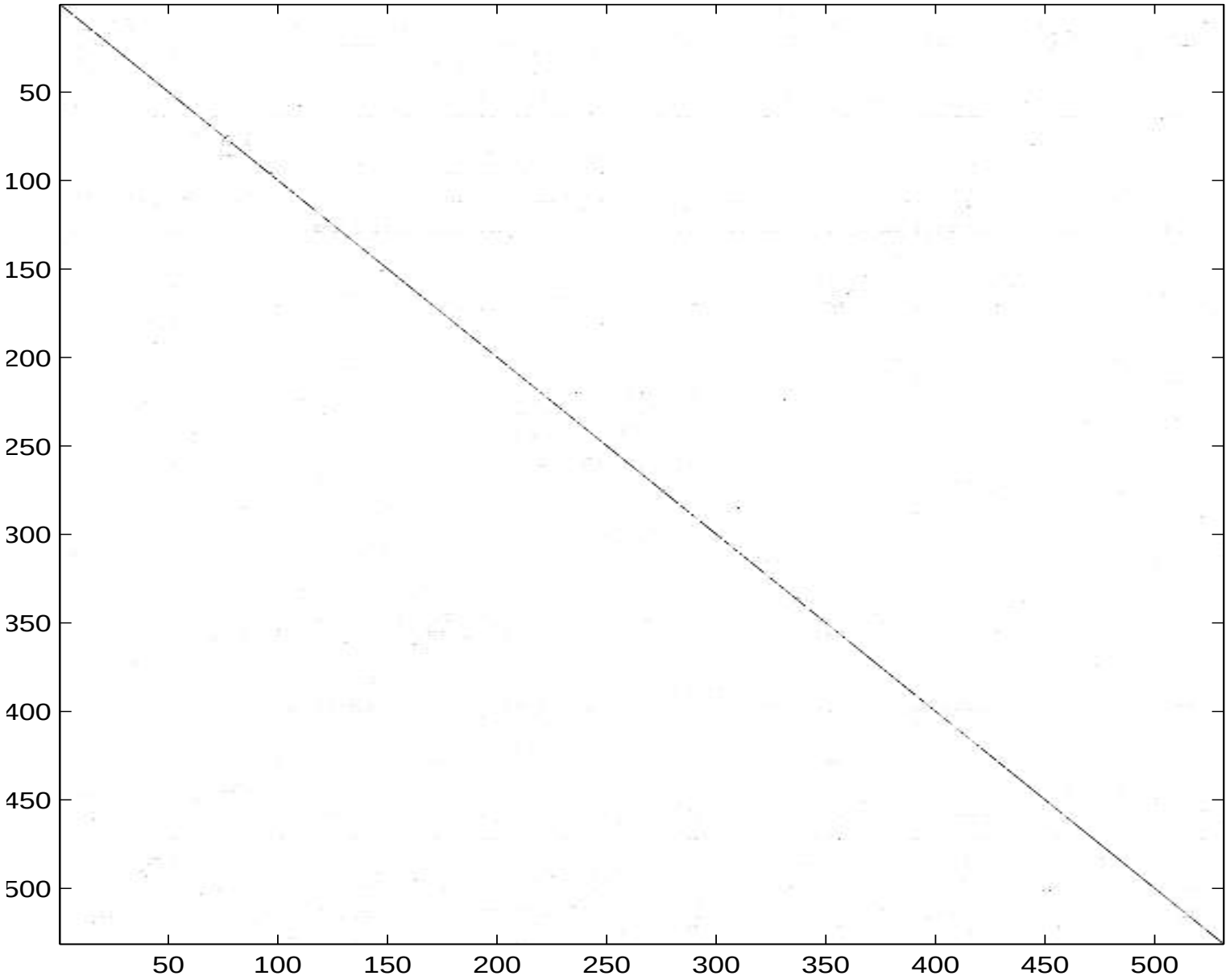
Nothing Found	15
Microsoft VC ++	13
Microsoft Visual Basic	2
Borland Delphi	8
UPX	7
Themida, Aspack	1
Nspack	2
PeCompact, LCC	1

A Closer Look



k-nn = 3, 100 families

64k malware, 531 families



Advantages of Image based Malware Analysis

- Fast (Feature computation time = 50 ms approx)
- No execution or disassembly.
- Images give more information about the structure of the malware.
- Visual Appeal: Develop new naming schemes based on similar malware images.
- Novel. Leverage techniques from Image Processing and Computer Vision community for Malware Analysis.

Limitations of Image based Malware Analysis

- **Data Driven**: Analysis based on existing malware. Hence, difficult to prevent a zero day attack.

- **Characterization**: At present, the characterization of malware as images does not give much information about the actual behavior of the malware other than the label given by AV software. Also, we do not look for actual malware signatures.

Thank You