Kötücül Yazılımların Tanınmasında Evrişimsel Sinir Ağlarının Kullanımı ve Karşılaştırılması

Utilization and Comparision of Convolutional Neural Networks in Malware Recognition

Ahmet Selman Bozkir, Ahmet Ogulcan Cankaya, Murat Aydos
Hacettepe Üniversitesi Bilgisayar Mühendisliği Bölümü, Ankara, Türkiye
E-mail: selman@cs.hacettepe.edu.tr

***Özetçe*—Endüstri 4.0, IoT ve mobil sistemlerdeki gelişmeler beraberinde bu tür sistemleri hedef alan kötücül yazılım tehditlerinde artışa neden olmaktadır. Yapılan araştırmalar kötücül yazılım dosyalarından byte düzeyinde çıkarılmış imgelerin bilgisayarlı görü ve makine öğrenimi yöntemleriyle sınıflandırılmasının etkin bir *statik* çözüm olabileceğini göstermektedir. Bu çalışmada kötücül yazılımları tespit amaçlı olarak görsel sınıflandırma probleminde başarısı kanıtlanmış modern evrişimsel sinir ağları (Resnet, Inception, DenseNet, VGG ve AlexNet) kullanılarak kestirim performansları ile birlikte model üretme ve tahmin süreleri karşılaştırılmıştır. Bununla birlikte çalışmada 8750 eğitim, 3644 test örneği içeren 25 sınıflı özgün bir kötücül yazılım veri kümesi önerilmiş ve kullanılmıştır. Elde edilen 3 kanallı (RGB) görsellerle yapılan deneyler sonucunda doğruluk bağlamında en yüksek başarı DenseNet ağlarında 97.48% olarak tespit edilmiştir.**

Anahtar Kelimeler — kötücül yazılım; kötücül yazılım tespiti; evrişimli sinir ağları; bilgisayarlı görü; gözetimli öğrenme.

***Abstract*— Advances in Industry 4.0, IoT and mobile systems have led to an increase in the number of malware threats that target these systems. The research shows that classification via the use of computer vision and machine learning methods over byte-level images extracted from malware files could be an effective *static* solution. In this study, in order to detect malware, we have employed various contemporary convolutional neural networks (Resnet, Inception, DenseNet, VGG, AlexNet) that have proven success in image classification problem and compared their predictive performance along with duration of model production and inference. In addition, a novel malware data set involving 8750 training and 3644 test instances over 25 different classes was proposed and used. As a result of the experiments carried out with 3-channel (RGB) images obtained, the highest success in terms of accuracy was determined as 97.48% by using DenseNet networks.**

Keywords — malware; malware detection; convolutional neural networks; computer vision; supervised learning.

978-1-7281-1904-5/19/$31.00 ©2019 IEEE

I. GİRİŞ

Son yıllarda gerek internet gerekse de bilgi tabanlı sistemler başta olmak üzere IoT (*Internet of Things*) ve mobil cihazlardaki gelişmeler beraberinde kötücül yazılımlarda (malware) üssel bir artışa neden olmuştur. Kötücül yazılımlar virüs, Truva atı, reklam gösterici gibi genellikle zararlı amaçlar taşıyan ve saldırganlara çeşitli faydalar sağlayan yazılımlardır. Aktas ve Sen [1]'e göre kötücül yazılım geliştirmede temel amaçların başında haksız finansal kazanç gelmektedir. Bu nedenle ortaya çıkan KY sayısı ve çeşitliliği her geçen gün artmaktadır [2]. Bir örnek vermek gerekirse [2], 2016 yılında bir önceki yıla nazaran 36% artışla 430 milyon farklı KY örneği tespit edilmiştir.

Kötücül yazılımların tespiti ve tanınmasında genel olarak *statik* ve *dinamik* çözümleme yöntemleri ön plana çıkmaktadır. Statik çözümlemeler zararlı kodun çalıştırılmaksızın ihtiva ettiği ikili (*binary*) dizilimleri, kütüphane çağrıları, opcode (*operational code*) sıklık dağılımları, akış kontrol çizgeleri gibi örüntüleri keşfetmeyi ve bu örüntüler üzerinden tanımlayıcı bir imza oluşturmayı hedefler [3]. Öte yandan dinamik çözümleme yaklaşımında potansiyel zararlı kod bir kum kutusu ya da sanal makine üzerinde çalıştırılarak davranışsal örüntüleri ortaya konmaktadır. Bu noktada, statik çözümleme kaynak tüketimi yönünden etkin ve çok daha hızlı sonuçlar ortaya koyarken polimorfik ve metamorfik zararlıların tespitinde yetersiz kalabilmektedir. Diğer yandan dinamik çözümleme bu türden zafiyetlerin önüne geçerken zaman ve kaynak tüketici olmaktadır.

Statik çözümleme tabanlı bir yaklaşım olarak Nataraj v.d. [4] kötücül yazılımlara ait ham ikili dosyalardaki byte dizilimlerinin gri ölçekli resimlere dönüştürülmesi sonrasında bilgisayarlı görü ve makine öğrenimi yöntemlerinden yararlanılarak sınıflama yapılabilirliğini göstermiştir. Bu kavramın dayandığı temel fikir, kötücül dosyaların ait oldukları familyalarda genellikle küçük değişiklikler göstermesidir. Nataraj v.d [4] yaptıkları ilk çalışmada KY görsellerinden elde ettikleri GIST betimleyicilerinden yararlanarak çok sınıflı KY

tanıması yapmışlardır. Sonraki yıllarda bilgisayarlı görü alanında derin öğrenmenin önemli bileşenlerinden biri olan evrişimsel sinir ağlarının (*convolutional neural networks – cnn*) önerilmesi ve yaygınlaşmasıyla birlikte KY görsellerinin analizinde CNN ağlarının kullanımı gündeme gelmiştir. Güncel çalışmalarda [5,6] ham KY görselleri üzerinden uçtan uca eğitim gerçekleştirilerek doğruluk (*accuracy*) ölçeğinde 97%'yi aşan başarılar yakalanmıştır. Keza, yine Saxe ve Berlin [7] derin öğrenme yöntemlerinin bahsi geçen problem üzerinde etkinliğini ölçümlemişlerdir.

İlgili çalışmalara bakıldığında CNN kullanılarak yapılan araştırmaların büyük kısmının VGG [9] ve Alexnet [8] ağları veya özgün ağ mimarileri üzerinde gerçekleştirilmiş olduğu gözlemlenmektedir. Ne var ki, CNN yöntemleri günümüze değin hız kaybetmeden gelişmiş, daha etkin ağ tasarımları daha az parametre içerecek şekilde tasarlanmış ve önerilmiştir. Bu çalışmada 2012 yılından bu yana alan yazınına girmiş ve doğal görsellerin sınıflandırılması probleminde başarısı kanıtlanmış modern CNN mimarilerinin kötücül yazılım sınıflandırmadaki etkinliği araştırılmış ve incelenmiştir. Bu bağlamda bu çalışmanın katkıları şu şekilde sıralanmaktadır:

- AlexNet ve VGG ağlarından farklı olarak Inception [10], ResNet [11] ve DenseNet [12] mimarileri farklı varyasyonlarıyla birlikte problem uzayında denenerek başarımları ve eğitim süreleri ölçülmüştür.
- Gri ölçekli görsellerin kullanımı terkedilerek 3 kanallı RGB görseller elde edilmiş ve deneylerde kullanılmıştır.
- Deneylerin yapılabilmesi adına "MaleVis" [13] ismi verilmiş olan özgün bir kötücül yazılım veri kümesi kurulmuştur ve gelecek çalışmalar için erişime açılmıştır. Bu veri kümesi farklı familyalara ait kötücül yazılımların ham PE (*portable executable*) dosyaları ile bu dosyalardan elde edilen farklı çözünürlükteki imgeleri içermektedir.

Bildirinin geri kalanı şu şekilde düzenlenmiştir. Bölüm 2'de çalışmanın kullandığı yöntem ve araçlar kısaca tanıtılmıştır. Bölüm 3'de veri kümesine ait ayrıntılar paylaşılmıştır. Bölüm 4'de deney tasarımı ve sonuçlar sunulmuştur. Son kısımda sonuç bölümü yer almaktadır.

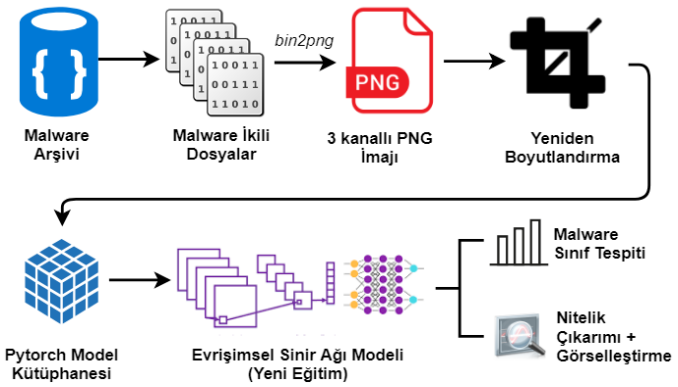
II. YÖNTEM VE ARAÇLAR

A. Evrişimsel Sinir Ağları

İlk olarak 2012 yılında Krizhevksy v.d. [8] tarafından doğal görüntülerin sınıflandırılması probleminde önerilen (Alexnet) ve elde ettiği başarıyla devrim yaratan evrişimsel sinir ağları zaman içerisinde nesne tanıma, segmentasyon ve hatta metin sınıflandırma gibi problemlere uyarlanmıştır. İlerleyen yıllar içerisinde bu yeni nesil yapay sinir ağları önceleri derinlik anlamında gelişme göstermiştir. Oxford Üniversitesi Visual Geometry Group bünyesinde önerilen VGG ağları (Vgg-11, Vgg-16, Vgg-19) [9] daha yüksek kestirim performansı göstermekle birlikte gerekli eğitim süresi ve parametre sayısı bakımından maliyetlidir. Öte yandan VGG ağları yeteri sayıda eğitim verisi bulunmadığı takdirde kaybolan eğim (*vanishing gradient*) problemine karşı duyarlılık taşımaktadır. Bu problemleri aşmak adına 1×1, 3×3 ve 5×5 boyutunda filtrelerin birlikte kurgulandığı "inception" bloklarını içeren 22 katmanlı GoogleNet ağı Szegedy v.d. [10] tarafından önerilmiş ve ImageNet 2015 yarışmasında birincilik kazanmıştır. İlerleyen dönemde daha derin ağların kestirim performansına olumlu katkı koyduğu gözlemlenmiş ancak kaybolan eğim problemini aşabilmenin ve 100 katmandan daha derin ağları üretebilmenin bir yolu olarak birim matrislerinin evrişim operasyonu sonrasında *toplanmasına* dayalı Resnet [11] mimarisi önerilmiştir. Bu yaklaşımdan farklı olarak Huang v.d. [12] her katman bloğunun diğer katman bloklarına bilgi taşımasına izin veren yoğun (*dense*) blok ve geçiş (*transition*) katmanlarının peşi sıra birleştirilmesi kavramını önermiştir. Bu yaklaşım DenseNet mimarisinin temelini oluştururken elde edilen nitelik haritalarının Resnet mimarisindeki toplama işlemi yerine birleştirme (*concatanation*) işlemine sokulmasına dikkat edilmelidir. Sayfa sınırı olmasından dolayı okuyucunun daha ayrıntılı bilgi için ilgili numaralandırılmış kaynakları incelemeleri önerilmektedir.

B. Ön İşlemler

Bu çalışmada Şekil 1 de gösterildiği üzere farklı CNN mimarilerinin kötücül yazılımlardan elde edilen imgeler üzerindeki sınıflama başarımları gözlemlenmeye çalışılmıştır. Bu nedenle ilk olarak ikili dosyaların imgelere dönüştürülmesi gerekmiştir. Bu noktada https://github.com/ESultanik/bin2png adresinde yer alan ve Python dilinde kodlanmış olan "bin2png" betiğinden yararlanılmıştır. Bin2png betiği kendisine verilen herhangi bir ikili dosyayı PNG biçimine kayıpsız olarak dönüştürebilmektedir. Ancak ilgili betiğin özgün hali Python 2.7 platformuna göre kodlanmıştır. İlk olarak daha etkin bir kullanım için özgün betik tarafımızca Python 3 platformuna uyumlandırılmış ve veri kümesinde yer alan tüm KY sınıflarına ait PNG imgeler oluşturulmuştur. İmgelerin oluşturulma aşamasında görsellerin genişliği 224 veya 300 piksel olacak şekilde ayarlanmıştır. Daha sonrasında farklı CNN ağlarının farklı boyuttaki girdi ihtiyaçlarına yönelik birbirinden farklı uzunluktaki imgeler OpenCV [14] kütüphanesi üzerinde 8×8 komşuluğu dikkate alan Lanczos interpolasyonu yöntemi kullanılarak 224×224 ve 300×300 boyutuna ölçeklendirilmiştir. Böylelikle ölçeklendirme aşamasında imgelere ait piksellerdeki kaybın minimize edilmesi hedeflenmiştir. Elde edilen imgelerden oluşan veri kümesi genel olarak 70% eğitim – 30% geçerleme olacak şekilde bölümlenerek nihai veri kümesi oluşturulmuştur.



Şekil. 1.Analiz sürecinin akış çizeneği

C. *Derin Öğrenme Kütüphane ve Çatıları*

Çalışma boyunca testleri gerçekleştirilen CNN modelleri Pytorch v0.4 [15] ve Caffe v0.16.1 [16] derin öğrenme çatıları kullanılarak eğitilmiştir. Eğitimlerde kullanılan ağların ön eğitimli modelleri yerine baştan (*from scratch*) eğitim tercih edilmiştir. Inception (GoogleNet) ağı haricindeki tüm diğer ağlar Pytorch içerisinde tanımlı olduğundan bu ağların eğitiminde ve testlerinde Pytorch çatısından faydalanılmıştır.

III. VERİ KÜMESİ

Bu belgede bahsi geçen çalışmaya başlanmadan önce ilgili çalışmaların kullandıkları veri kümeleri incelendiğinde bu veri kümelerinin ham ikili dosyaları içermediği veya gri ölçekli imgelerden oluştukları gözlenmiştir. Dolaysıyla bu içerikler üzerinden çalışmanın yürütülmesi olanaksız hale gelmiştir. Bu nedenle çalışma için kötücül yazılım laboratuvarı bulunan ve bu alanda faaliyet gösteren bir siber güvenlik firmasıyla ortak çalışılarak 25 farklı familyadan oluşan ham PE dosya kümesi derlenmiştir. Derlemdeki dosyalar 2017-2018 yıllarında ortaya çıkan kötücül yazılımlardan elde edilmiştir. Her bir familya için 500 veya yakın sayıda KY örneği içeren ve MaleVis – "**Mal**ware **E**valuation with **Vis**ion" [12] olarak adlandırılan veri kümesi dengeli bir dağılıma sahiptir. Aşağıda yer alan Tablo 1 de veri kümesinde yer alan kötücül yazılımların familya, kategori ve sayıca dağılımları sunulmuştur. Tabloda görüleceği üzere eğitim ve test örnekleri 70%-30% şeklinde oluşturulmuştur. Veri kümesi toplamda 8750 eğitim, 3644 test örneği barındırmaktadır.

TABLO I. MALEVIS VERİ KÜMESİ İÇERİĞİ

No	Sınıf İsmi	Kategori	Eğitim/Test Örneği Sayıları
1	Win32/Adposhel	Adware	350/144
2	Win32/Agent-fyi	Trojan	350/120
3	Win32/Allaple.A	Worm	350/128
4	Win32/Amonetize	Adware	350/147
5	Win32/Androm	Backdoor	350/150
6	Win32/AutoRun-PU	Worm	350/146
7	Win32/BrowseFox	Adware	350/143
8	Win32/Dinwod!rfn	Trojan	350/149
9	Win32/Elex	Trojan	350/150
10	Win32/Expiro-H	Virus	350/150
11	Win32/Fasong	Worm	350/150
12	Win32/HackKMS.A	Trojan	350/149
13	Win32/Hlux!IK	Worm	350/150
14	Win32/Injector	Trojan	350/145
15	Win32/InstallCore.C	Adware	350/150
16	Win32/MultiPlug	Adware	350/149
17	Win32/Neoreklami	Adware	350/150
18	Win32/Neshta	Virus	350/147
19	Win32/Regrun.A	Trojan	350/135
20	Win32/Sality	Virus	350/149
21	Win32/Snarasite.D!tr	Trojan	350/150
22	Win32/Stantinko	Backdoor	350/150
23	VBA/Hilium.A	Virus	350/150
24	Win32/VBKrypt	Trojan	350/146
25	Win32/Vilsel	Trojan	350/146

MaleVis veri kümesi gerek ham KY dosyalarını gerekse de RGB kanallı PNG imgelerini doğrudan derin öğrenme kütüphanelerinin kullanılabileceği şekilde kurulmuştur. Bununla birlikte ilgili veri kümesi yapılacak sonraki çalışmalar için araştırmacıların ücretsiz kullanımına açılmıştır.

IV. DENEY TASARIMI VE SONUÇLAR

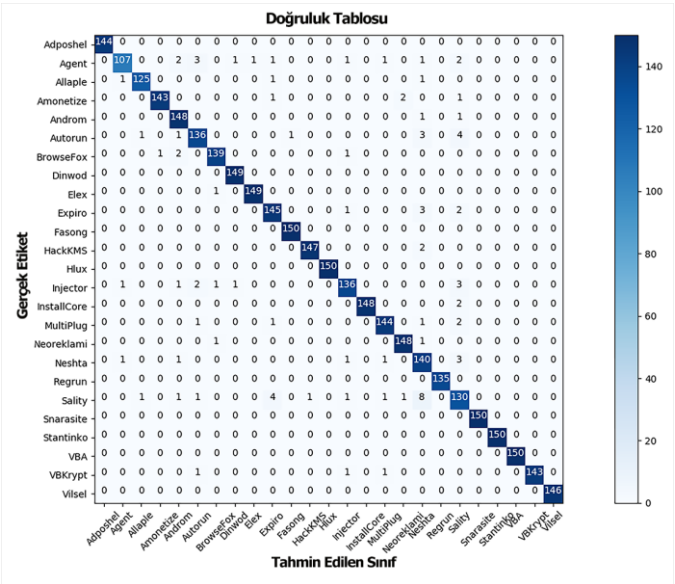
A. *Deney Süreci*

Bu çalışmada daha önceki kesimlerde belirtildiği üzere AlexNet, VGG (11 ve 16 katmanlı), Resnet (18, 34, 50 ve 101 katmanlı) Inception (sürüm 1 ve 3) ve Densenet (121, 169 ve 201 katmanlı) CNN mimarileri kullanılmıştır. Yapılan eğitimlerin ilk basamağında tüm evrişimsel filtreler rassal ağırlıklarla başlatılmıştır. Gerçekleştirilen deneylerde öğrenme katsayısı (*learning rate*) 0.01, sönümleme katsayısı (*momentum*) 0.8, adım sayısı (*step size*) 10 olarak belirlenmiş ve tüm eğitimler 60 dönemi (*epoch*) kapsayacak şekilde uygulanmıştır. Ayrıca geri yayılım eniyilemesinde SGD (Stochastic Gradient Descent) yöntemi benimsenmiştir.

Deneyler,16 GB DDR4 hafızaya, Intel i7 6700K işlemciye, 512 GB SSD sabit diske ve NVidia GTX 1060 6GB GPU'ya sahip bir bilgisayar üzerinde GPU tabanlı gerçekleştirilmiştir. Farklı CNN mimarilerinin farklı bellek gereksinimleri bulunmasından ötürü tüm CNN modelleri kullandığımız GPU'nun belleğine maksimum derecede sığacak yığın boylarında (*batch size*) eğitilmiştir.

B. *Sonuçlar ve Tartışma*

Gerçekleştirilen deneylere ait sonuçlar Tablo 2'de sunulmuştur. Sonuçlar elde edilen en yüksek doğruluk (*accuracy*), her bir dönemin eğitim ve geçerleme süresi, toplam eğitim süresi ve yığın sayısını kapsamaktadır. Buna göre problem uzayında doğruluk bağlamında en başarılı mimari 97.48% doğrulukla Densenet türevleri olmuştur. Ayrıca Densenet ağları diğer ağlara nazaran çok daha kısa sürede yakınsayabilmiştir. Aşağıda yer alan Şekil 2'de geçerleme verisi üzerinde Densenet 201 ağıyla elde edilmiş örnek bir doğruluk tablosu sunulmuştur. Densenet mimarisinin katman sayısındaki farklılıklar süreleri etkilemekle birlikte geçerleme başarısında bir farklılığa neden olmamıştır. Bu bağlamda 121 katmanlı DenseNet mimarisinin kullanımı yeterli olabilmektedir.



Şekil. 2. DenseNet201 ağıyla elde edilen doğruluk tablosu

TABLO II. FARKLI CNN MİMARİLERİNDEN ELDE EDİLEN EN YÜKSEK DOĞRULUK SONUÇLARI VE DİĞER ÖZELLİKLER

Ağ-Mimari	Doğruluk (Eğitim)	Doğruluk (Geçerleme)	Epoch Süreleri (Eğitim/Geçerleme)	Toplam Eğitim Süresi	Yığın Sayısı
AlexNet	98.73%	94.43%	11/2 saniye	13 dakika	128
VGG11	99.99%	96.46%	132/16 saniye	153 dakika	16
VGG16	99.82%	96.10%	242/31 saniye	278 dakika	16
Resnet18	99.99%	97.17%	39/5 saniye	45 dakika	64
Resnet34	99.98%	96.84%	76/9 saniye	84 dakika	48
Resnet50	99.97%	97.03%	119/16 saniye	136 dakika	16
Resnet101	99.97%	97.09%	212/26 saniye	233 dakika	12
Inception (Googlenet)	99.99%	96.38%	42/8 saniye	50 dakika	32
Inception V3	99.53%	96.62%	180/24 saniye	214 dakika	12
Densenet121	99.98%	**97.48%**	122/16 saniye	138 dakika	12
Densenet169	99.92%	**97.48%**	169/23 saniye	192 dakika	8
Densenet201	99.98%	**97.48%**	217/29 saniye	247 dakika	8

Başarı anlamında Resnet18 mimarisi çok daha az katmana sahip olmakla birlikte 97.18% doğruluk ile ikinci sırada yer almıştır. Resnet18 ağının parametre sayısındaki tutumluluğu, hızlı eğitimi ve toplamda 3644 imgeyi 5 saniye gibi kısa sürede geçerleyebilmesi problem uzayında tercih edilebilir bir mimari olduğunu göstermektedir. VGG ağları daha önceki çalışmalarda kullanılmış ve başarısı kanıtlanmış ağlar olmakla birlikte daha yeni ağlar karşısında geride kalmaktadır.

Sonuçların ortaya koyduğu iki önemli bulgu bulunmaktadır. İlki kötücül yazılımların tanınması probleminde kullanılan ağlar bağlamında evrişimsel katman sayısının başarım üzerinde ciddi bir etki yaratamadığıdır. İkincisi ise KY imgelerinde çıkarılan nitelik ve nitelik haritalarından elde edilen bilginin ilk katmanlardan uç katmanlara kadar taşınabilmesinin ve *birleştirilebilmesinin* başarımda pozitif etki göstermesidir. Densenet mimarisinin diğer ağlara nazaran daha ilk dönemlerde hızlı yakınsaması ve gösterdiği başarının altında yatan temel nedenin bu olduğunu düşünmekteyiz.

V. SONUÇ

Bu çalışmada kötücül yazılımların tanınması probleminde 2012 yılından bugüne değin geliştirilmiş birçok CNN mimarisi önerilen bir veri kümesi üzerinde test edilerek sonuçları ortaya koyulmuştur. Yapılan analizler neticesinde Densenet ve Resnet mimarilerinin diğer mimarilere göre daha başarılı olduğu, 3 kanallı RGB imgelerin problemin çözümünde sorun yaratmadığı gözlemlenmiştir. Bununla birlikte gelecek çalışmalarda problemin açık küme (*open-set*) problemi haline dönüştürülerek, kötücül yazılımlarla birlikte güvenilir yazılımların tek bir çatı altında sınıflandırılabilmesinin araştırılması planlanmaktadır.

BİLGİLENDİRME

Yazarlar, bu çalışmada kullanılan kötücül yazılım veri kümesinin temininde Comodo Inc. firmasına teşekkürlerini sunar.

KAYNAKLAR

[1] K. Aktas and S. Sen,"UpDroid: Updated Android Malware and Its Familial Classification",in Nordsec'18 2018.

[2] S. Ni, Q. Qian and R. Zhang, "Malware identification using visualization images and deep learning", *Computers & Security*, vol. 77, pp.871-885, 2018. .

[3] L. Nataraj, D. Kirat, B.S. Manjunath, G. Vigna, "SARVAM: Search and RetrieVAl of Malware", in NGMAD'13, 2013.

[4] L. Nataraj, S. Karthikeyan, G. Jacob and B.S. Manjunath, "Malware Images: Visualization and Automatic Classification", in VizSec'11, 2011.

[5] K. Özkan, Ş. Işık and Y. Kartal, "Evaluation of Convolutional Neural Network Features for Malware Detection", In *6th International Symposium on Digital Forensic and Security*, 2018.

[6] S. Yue, "Imbalanced Malware Images Classification: a CNN based Approach", *arXiv: 1708.08042*, 2017.

[7] J.Saxe and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features", In *10th International Conference on Malicious and Unwanted Software*, 2015.

[8] A. Krizevsky, I. Sutsever and G. Hinton, "Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097-1105, 2012.

[9] K. Simonyan ve A. Zisserman, "Very deep convolutional network for large scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[10] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9, 2015.

[11] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition", *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016.

[12] G. Huang, Z. Liu, L.V.D. Maaten, K. Q. Weinberger, "Densely connected convolutional network", *arXiv preprint arXiv: 1608.06993*, 2016.

[13] Malevis Dataset, "https://web.cs.hacettepe.edu.tr/~selman/malevis/"

[14] OpenCV, https://opencv.org/ (8.2.2019 tarihinde ziyaret edilmiştir)

[15] Pytorch, https://pytorch.org/ (8.2.2019 tarihinde ziyaret edilmiştir)

[16] Caffe, http://caffe.berkeleyvision.org/ (8.2.2019 tarihinde ziyaret edilmiştir)