ThermalAttackNet: Are CNNs Making It Easy To Perform Temperature Side-Channel Attack In Mobile Edge Devices?

Somdip Dey [1], Amit Kumar [2], and Klaus D. Mcdonald-Maier [2]

[1]University of Essex
[2]Affiliation not available

October 30, 2023

Abstract

Side-channel attacks remain a challenge to information flow control and security in mobile edge devices till this date. One such important security flaw could be exploited through temperature side-channel attacks, where heat dissipation and propagation from the processing cores are observed over time in order to deduce security flaws. In this brief, we study how computer vision based convolutional neural networks (CNNs) could be used to exploit temperature (thermal) side-channel attack on different Linux governors in mobile edge device utilizing multi- processor system-on-chip (MPSoC). We also designed a power- and memory-efficient CNN model that is capable of performing thermal side-channel attack on the MPSoC and can be used by industry practitioners and academics as a benchmark to design methodologies to secure against such an attack in MPSoC.

ThermalAttackNet: Are CNNs Making It Easy To Perform Temperature Side-Channel Attack In Mobile Edge Devices?

Somdip Dey, *Student Member, IEEE,* Amit Kumar Singh, *Member, IEEE,*
and Klaus Dieter McDonald-Maier, *Senior Member, IEEE*

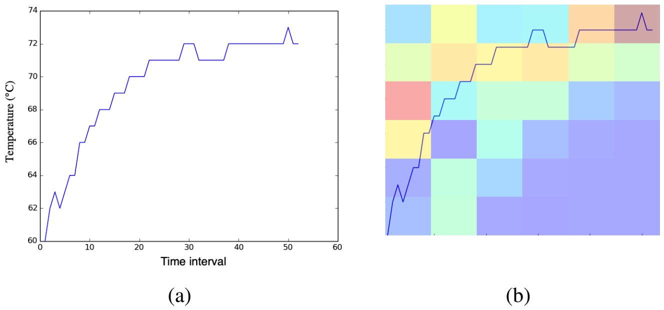
***Abstract*—Side-channel attacks remain a challenge to information flow control and security in mobile edge devices till this date. One such important security flaw could be exploited through temperature side-channel attacks, where heat dissipation and propagation from the processing cores are observed over time in order to deduce security flaws. In this brief, we study how computer vision based convolutional neural networks (CNNs) could be used to exploit temperature (thermal) side-channel attack on different Linux governors in mobile edge device utilizing multiprocessor system-on-chip (MPSoC). We also designed a power- and memory-efficient CNN model that is capable of performing thermal side-channel attack on the MPSoC and can be used by industry practitioners and academics as a benchmark to design methodologies to secure against such an attack in MPSoC.**

***Index Terms*—multiprocessor system-on-chip (MPSoC), thermal behaviour, temperature side-channel attack, security, machine learning, convolutional neural network (CNN), deep learning, energy efficiency, memory efficiency**

I. Introduction

RECENTLY, mobile devices have become an integral part of daily life. These mobile devices are utlised to run different types of applications, including video calling, web browsing, gaming, navigation, and hence, energy-efficient processing on these battery-empowered mobile devices is of utmost importance [1], [2]. Mobile cloud computing, where most of the computations happen in the cloud (also known as *Cloud Offloading*) [3], is considered to be a potential solution for energy-efficient processing. However, application processing that needs privacy and security, such as banking app and secure data storage app, is often processed on the mobile device instead of cloud offloading. Moreover, as mobile edge computing becomes more and more ubiquitous, security issues in these mobile devices become more paramount. Such mobile devices have to face hostile security threats [4]–[7] such as physical, logical/software-based and side-channel/lateral attacks. Amongst these, side-channel attack [4]–[6] is a popular security threat due to ease of access to the physical hardware where attacks are performed by observing the properties and behavior of the system such as power consumption, thermal dissipation, electromagnetic emission, etc.

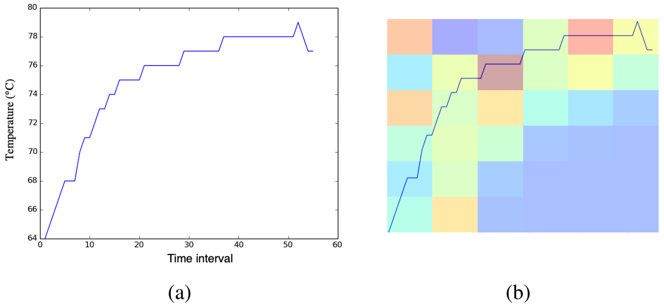
Comparatively a lot less documented studies are performed in temperature (thermal) based side channel attacks [4]–[6] in mobile edge devices. Most of these mobile devices come equipped with heterogeneous multi-processors systems-on-chip (MPSoC), which consists of multiple heterogeneous processors on a single chip, capable of processing different types of applications to cater for performance and energy-efficiency of the executing applications. Due to an increase in the usage of MPSoCs [2], [8]–[13] in mobile edge devices and a rise in studies on thermal side channel attacks [6], [14], [15], it is crucial that side channel attacks in such platform should be addressed with utmost importance [15].



(a)

(b)

Fig. 1. Focus area of ResNet network on a representative graph of password: 111111



(a)

(b)

Fig. 2. Focus area of ResNet50 network on a representative graph of password: passw0rd

To explore feasibility of thermal side-channel attack in a real commercial mobile device we designed a new type of attack which involved computer vision based Convolutional Neural Network (CNN). Among all the fields of Neural

This work is supported by the UK Engineering and Physical Sciences Research Council EPSRC [EP/R02572X/1 and EP/P017487/1].

Network based machine learning and pattern recognition, computer vision based Neural Networks, especially CNNs and Deep Learning (DL) [16], [17], are well studied and mature comparatively. Recently, CNN models have achieved high prediction accuracy in applicative fields to solve several real-life challenges such as traffic categorization [18], [19], human rights violation [20], weather forecasting [21], etc. Given the high success rate in understanding patterns, we utilized a CNN model based attack. To perform the attack we chose 4 of the 25 most common passwords of 2017 and 2018 [22], [23] as surveyed by the Internet security firm SplashData. The 4 common passwords used by the user, which are chosen for our attack, are *123456, passw0rd, 111111* and *football*. We then executed AES-256 [24] encryption on a text file using the aforementioned passwords on Odroid XU4 development board [25] running on ondemand Linux governor and recorded the thermal behaviour of the CPUs. We trained ResNet model [26], a pre-trained CNN model trained on ImageNet using *transfer learning*, with the graphical representation of the thermal behaviour (as shown in Fig. 1.(a) & Fig. 2.(a)). ResNet was able to achieve a training prediction accuracy of 46.88% and a testing prediction accuracy of 31.99%, which means that ResNet is able to predict the correct password, one out of every four attempts on an average. Fig. 1.(b) & Fig. 2.(b) show the region of interest on the graphical representation of the thermal behaviour which is used by the CNN model to predict the password. In order to determine whether ResNet is classifying the thermal data based on the features of the thermal peaks, we utilized Gradient-weighted Class Activation Mapping (Grad-CAM) [27] to visualize in which areas of the graphical data the CNN was focusing on to predict the password being used for encryption process. In Fig. 1.(b) & Fig. 2.(b), the area highlighted (heat map) with shades of yellow/red is the active region where the CNN is looking to determine the password used. In the heat map, the regions range from blue to red, where blue means least active region and red means the most active one. The observations from the aforementioned figures prove that visual based CNNs could be successfully utilized to perform thermal side-channel attack and to the best of our knowledge this is the first documented study to do so. In summary, this brief makes the following contributions.

1) Design and explore thermal side-channel attack using computer vision based CNN models.
2) Evaluate popular CNN models and their accuracy in predicting password for different Linux governors.
3) Design and implementation of a power- and memory-efficient CNN model, ThermalAttackNet, to perform thermal side-channel attack on a real consumer mobile device.

The main motive to design and implement a computer vision based CNN model to perform thermal side-channel attack is to provide a benchmark that could be used by industry practitioners and researchers to improve security against such an attack in mobile devices utilizing MPSoCs.

II. Preliminaries

A. Convolutional Neural Networks and Deep Learning

A Deep Learning (DL) model [28] consists of an input layer, several intermediate (hidden) layers, which are stacked on top of each other, and an output layer. In the input layer, which is the first layer of the model, the raw values of data features are fed into it. In each of the hidden layers a mathematical operation called convolution is applied to extract specific features, which is then utilized to predict the label of the raw data in the last (output) layer of the DL network. Most of the time, if a model utilize an input layer, a hidden layer and an output layer then the model is denoted as Convolutional Neural Network (CNN) model or simply, CovNet. If such a model uses a lot of stacked hidden layers only then it is denoted as a DL model or Deep Neural Networks (DNN).

B. Pre-trained Networks and Transfer Learning

A conventional approach to enable training of DNN/CNN on relative small datasets is to use a model pre-trained on a very large dataset, and then use the CNN as an initialization for the applicative task of interest. Such a method of training is called "*transfer learning*" [29] and we have followed the same principle. The chosen CNN models mentioned in Sec. III-C are pre-trained on ImageNet [28]. For the proposed attack, we have utilized the following popular pre-trained CNN models: VGG (VGG19) [30], ResNet (ResNet152v2) [26], MobileNet (MobileNetv2) [31] and NASNet (NASNetMobile) [32].

III. Thermal Side-Channel Attack using CNN

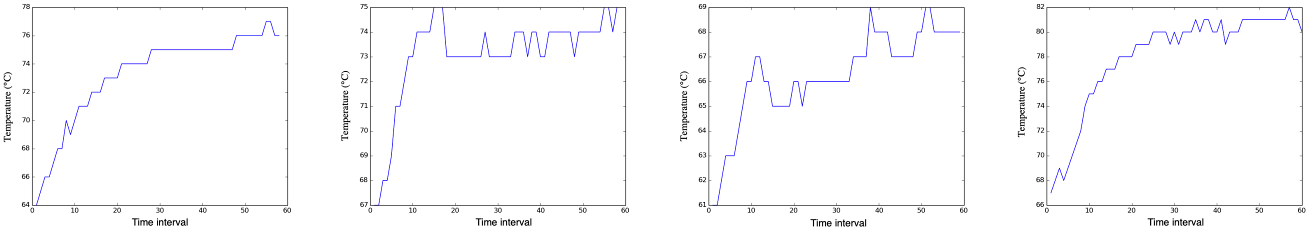
A. Hardware & Software Setup for Experiments

We also chose Odroid XU4 [25] board to execute the attack in order to verify the affect of thermal side-channel exploitation. Odroid XU4 employs the Samsung Exynos 5422 [33] MPSoC, which is popularly used in Samsung mobile devices, especially Samsung Galaxy S5. The Odroid XU4 is a representational development board of Galaxy S5 smart-phone. Exynos 5422 MPSoC contains clusters of big (4 Cortex A-15) and LITTLE cores (4 Cortex A-7). This MPSoC provides DVFS feature per cluster, where the big core cluster has 19 frequency scaling levels, ranging from 200 MHz to 2000 MHz with each step of 100 MHz and the LITTLE cluster has 13 frequency scaling levels, ranging from 200 MHz to 1400 MHz, with each step of 100 MHz.

The Odroid XU4 was running on UbuntuMate version 14.04 (Linux Odroid Kernel: 3.10.105). During the time of performing the attack the average ambient temperature of the room was 21°C. When we executed the attack we changed the governor between conservative, ondemand, performance, interactive and powersaver to study which Linux governor is more vulnerable to such attack.

B. Dataset and CNN Model

To generate a dataset of thermal behaviour we choose 4 most common passwords (*123456, passw0rd, 111111 & football*) and used AES-256 encryption algorithm to encrypt a text file



(a) 111111

(b) 123456

(c) football

(d) passw0rd

Fig. 3. Graphical representation of thermal behavior (time interval vs temperature in °C) of encryption operation using the following passwords: *111111, 123456, football & passw0rd*

using the aforementioned passwords. For each aforementioned password the encryption was performed on the same text file 500 times. The reason to choose AES-256 is because of its popularity. The encryption operations were performed on CPU 7, which is one of the big CPUs (A-15) of the Exynos 5422 MPSoC while one of the LITTLE cores (CPU 3) snoops the operating temperature data of the big CPU. After the temperature data for each password were collected, we transformed the data points into a graphical representation in order to be fed to a pre-trained CNN for training and prediction purposes. Fig. 3 shows the graphical representation of the thermal behaviour of CPU 7 for 111111 (Fig. 3.(a)), 123456 (Fig. 3.(b)), football (Fig. 3.(c)) and passw0rd (Fig. 3.(d)) respectively during the encryption process.

C. Training CNN To Predict Password

We choose a pre-trained CNN model, which is trained on 1000 classes of ImageNet[1], and removed the classifier module and modified it to be able to predict our chosen classes of password. We fine-tuned [34] the CNN model by adding a new randomly initialized classifier (output layer), and training the last fully connected layer by freezing all the layers of the base model (frozen layers represented with gray colour in Fig. 4) and unfreezing the last fully connected layer (unfrozen layers represented with green colour in Fig. 4). Freezing the layers mean that no updates to the weights are made in those layers during the training process. The new output layer of the model is then trained to take the lower level features passed through the model network and map them to the desired output classes (password), using optimization techniques such as stochastic gradient descent (SGD) [35]. SGD is an iterative optimization algorithm, which estimates the error gradient for the CNN model during the training process and updates the weights of the model using back-propagation [36].

IV. ThermalAttackNet: Proposed CNN Architecture

Since, most of the pre-trained CNNs come with several fully connected layers, using such a model consumes a lot of memory space on the device as well as power. In order to overcome these challenges we designed a CNN model, named *ThermalAttackNet*, which performs similar to popular CNNs (ResNet, VGG, NASNet and MobileNet), however, at the same time consumes least power and memory comparatively. Given the fact that graphical representation of thermal behaviour (as shown in Fig. 3) consists of regular temperature peaks characterized by edges, we designed the CNN to be able to extract such features as accurately as possible. The architecture of ThermalAttackNet is illustrated in Fig. 5. ThermalAttackNet consists of 6 convolutional layers (denoted by Conv2D in Fig. 5) and we discard the fully connected layers in favour of retaining higher resolution feature maps at the deepest output layer. This also reduces the number of parameters (only 48,804) used in ThermalAttackNet compared to ResNet, VGG, NASNet and MobileNet (as shown in Table I). In Fig. 5, it should be kept in mind that X is a variable batch size, which will depend on the implementation of the model, and C is the output classes, which is 4 (passwords) in our case. Each convolutional layer (Conv2D) performs convolution with a filter bank to produce a set of feature maps and then an element-wise rectified-linear non-linearity (ReLU) $max(0, x)$ is applied. Following that, max-pooling (denoted as MaxPooling2D in Fig. 5) is used to achieve translation invariance over small spatial shifts in the input image. Table I shows the comparison between ThermalAttackNet and other popular models.

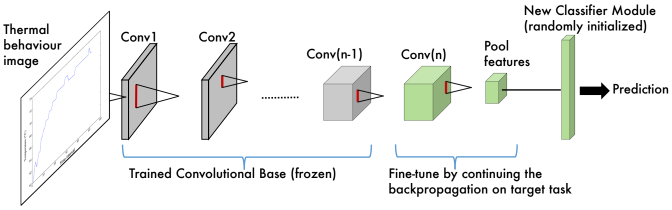


Fig. 4. Network architecture used for fine-tuning

Note: ThermalAttackNet is trained on thermal dataset by performing augmentation to the data improve its training. The following data augmentation approaches were performed on the dataset: Horizontal and Vertical Shift, Random Zoom & Shear Intensity.

[1] CNN model is pre-trained with 1000 different labels (classes) such as eskimo dog, madagascar cat, cougar, lifeboat of ImageNet database.

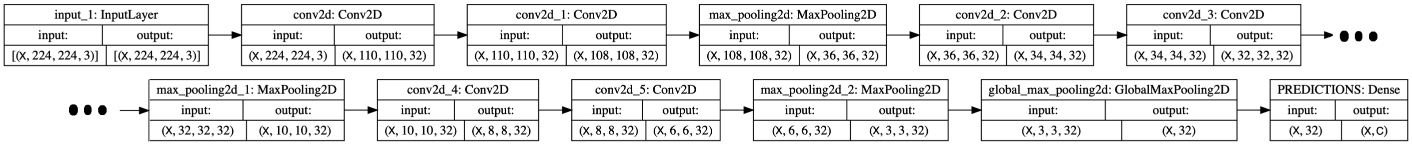


Fig. 5. An illustration of the ThermalAttackNet architecture

TABLE I
COMPARISON BETWEEN MODELS BASED ON DISK SIZE AND PARAMETERS

	Size	Parameters
ResNet152v2	232 MB	60,380,648
NASNetMobile	23 MB	5,326,716
VGG19	549 MB	143,667,240
MobileNetv2	14 MB	3,538,984
ThermalAttackNet	0.455 MB	48,804

TABLE II
TRAINING PREDICTION ACCURACY (%) ACHIEVED BY DIFFERENT CNNS ON DIFFERENT LINUX GOVERNORS: CONSERVATIVE (CONS.), ONDEMAND (OND.), PERFORMANCE (PERF.)

	cons.	ond.	perf.	inter.	pow.
ResNet152v2	45.875	46.875	65.625	29	41.625
NASNetMobile	26.6875	27.6875	34.25	25.8125	27.6875
VGG19	25.4375	26.1875	26	24.625	26.1875
MobileNetv2	55.625	66.0625	**69.6875**	42.5625	52.75
ThermalAttackNet	25	27	25.0625	25.1875	25.375

V. EXPERIMENTAL AND EVALUATION RESULTS

From the 500 graphical data for each password label, we separated 100 graphical data for cross-validation testing purpose. Whereas, 75% of the remaining 400 graphical data for each password label were used for training and rest of the 25% is used for validation during the training period. Validation data is used to provide an unbiased evaluation of a model fit on the training dataset while tuning hyperparameters of the model. Table II shows the training prediction accuracy and Table III shows the testing prediction accuracy achieved by MobileNetv2, NASNetMobile, ResNetv2, VGG19 & ThermalAttackNet respectively on different Linux governors: conservative (cons.), ondemand (ond.), performance (perf.), interactive (inter.) & powersaver (pow.).

A. Which CNN model is best at predicting password

In Table II, we could notice that MobileNetv2 achieves the highest training prediction accuracy of 69.6875 for performance governor, however, for the same governor the testing prediction accuracy drops to 25.7499% (see Table III). Since, testing prediction accuracy is more important to determine if the CNN is able to predict accurately, based on Table III ResNet152v2 achives the best prediction accuracy of 31.999%. Therefore, among these compared CNN models, ResNet152v2 is best at predicting password using our proposed thermal side-channel attack.

Which governor is least secure: From Table III it is evident that ondemand governor is least secure among other Linux governors if ResNet152v2 is used as model for the attack.

TABLE III
TESTING PREDICTION ACCURACY (%) ACHIEVED BY DIFFERENT CNNS ON DIFFERENT LINUX GOVERNORS: CONSERVATIVE (CONS.), ONDEMAND (OND.), PERFORMANCE (PERF.)

	cons.	ond.	perf.	inter.	pow.
ResNet152v2	25.99	**31.999**	31	25.499	25.499
NASNetMobile	27.5	25.7499	31	29.2499	27.25
VGG19	27.75	31.4999	28.49999	25	25
MobileNetv2	30.75	25.4999	25.7499	24.5	24.75
ThermalAttackNet	25.75	26	25	25	25

B. Power consumption of CNNs

The average power consumption (in Watt) during inference while utilizing ResNet15v2, MobileNetv2, VGG19, NASNetMobile & ThermalAttackNet on ondemand governor are 10.69, 9.56, 10.67, 8.79 & 7.63 respectively. Given the fact that ThermalAttackNet is fraction of a size of popular CNNs (see Table I) while is able to predict close to other popular CNNs (see Table III), utilizing ThermalAttackNet for such an attack on the device is more power efficient.

VI. CONCLUSION

In this brief, we studied the accuracy of different CNN models: ResNet15v2, MobileNetv2, VGG19 and NASNetMobile, to predict passwords exploiting thermal side-channel attacks for different Linux governors in mobile MPSoCs. Based on empirical data ondemand governor is the least secure among other Linux governors if ResNet152v2 is used as a CNN model for the attack. We also proposed a power-efficient CNN, ThermalAttackNet, which is able to predict passwords almost equally as ResNet152v2 CNN, however, in a more power-efficient manner while consuming least disk storage memory on the device.

VII. CODE AVAILABILITY

The program codes to implement the attack and generate the dataset could be accessed from ##Code will be provided upon acceptance##.

REFERENCES

[1] T. Q. Dinh, J. Tang, Q. D. La, and T. Q. Quek, "Offloading in mobile edge computing: Task allocation and computational frequency scaling," *IEEE Transactions on Communications*, vol. 65, no. 8, pp. 3571–3584, 2017.

[2] A. K. Singh, S. Dey, K. R. Basireddy, K. McDonald-Maier, G. V. Merrett, and B. M. Al-Hashimi, "Dynamic energy and thermal management of multi-core mobile platforms: A survey," *IEEE Design & Test*, 2020.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE internet of things journal*, vol. 3, no. 5, pp. 637–646, 2016.

[4] J. A. Ambrose, R. G. Ragel, D. Jayasinghe, T. Li, and S. Parameswaran, "Side channel attacks in embedded systems: A tale of hostilities and deterrence," in *Quality Electronic Design (ISQED), 2015 16th International Symposium on*. IEEE, 2015, pp. 452–459.

[5] J. De Haas, "Side channel attacks and countermeasures for embedded systems," *Black Hat, Las Vegas, NV, USA*, p. 82, 2007.

[6] M. Hutter and J.-M. Schmidt, "The temperature side channel and heating fault attacks," in *International Conference on Smart Card Research and Advanced Applications*. Springer, 2013, pp. 219–235.

[7] T. van Elsloo, "Multi-objective optimization of secure embedded systems architectures," 2016.

[8] S. Dey, E. Z. Guajardo, K. R. Basireddy, X. Wang, A. K. Singh, and K. McDonald-Maier, "Edgecoolingmode: An agent based thermal management mechanism for dvfs enabled heterogeneous mpsocs," in *2019 32nd International Conference on VLSI Design and 2019 18th International Conference on Embedded Systems (VLSID)*. IEEE, 2019, pp. 19–24.

[9] S. Dey, A. K. Singh, X. Wang, and K. D. McDonald-Maier, "Deadpool: Performance deadline based frequency pooling and thermal management agent in dvfs enabled mpsocs," in *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*. IEEE, 2019, pp. 190–195.

[10] S. Dey, A. K. Singh, D. K. Prasad, and K. D. Mcdonald-Maier, "Socodecnn: Program source code for visual cnn classification using computer vision methodology," *IEEE Access*, vol. 7, pp. 157 158–157 172, 2019.

[11] S. Isuwa, S. Dey, A. K. Singh, and K. McDonald-Maier, "Teem: Online thermal-and energy-efficiency management on cpu-gpu mpsocs," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 438–443.

[12] S. Dey, A. Singh, X. Wang, and K. McDonald-Maier, "User interaction aware reinforcement learning for power and thermal efficiency of cpu-gpu mobile mpsocs," in *2020 DATE*. IEEE, 2020, pp. 1728–1733.

[13] S. Dey, A. K. Singh, S. Saha, X. Wang, and K. D. McDonald-Maier, "Rewardprofiler: A reward based design space profiler on dvfs enabled mpsocs," in *2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*. IEEE, 2019, pp. 210–220.

[14] R. J. Masti, D. Rai, A. Ranganathan, C. Müller, L. Thiele, and S. Capkun, "Thermal covert channels on multi-core platforms." in *USENIX Security Symposium*, 2015, pp. 865–880.

[15] D. B. Bartolini, P. Miedl, and L. Thiele, "On the capacity of thermal covert channels in multicores," in *Proceedings of the Eleventh European Conference on Computer Systems*. ACM, 2016, p. 24.

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[17] S. Chakradhar, M. Sankaradas, V. Jakkula, and S. Cadambi, "A dynamically configurable coprocessor for convolutional neural networks," in *ACM SIGARCH Computer Architecture News*, vol. 38, no. 3. ACM, 2010, pp. 247–257.

[18] S. Dey, G. Kalliatakis, S. Saha, A. K. Singh, S. Ehsan, and K. McDonald-Maier, "Mat-cnn-sopc: Motionless analysis of traffic using convolutional neural networks on system-on-a-programmable-chip," in *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE, 2018, pp. 291–298.

[19] S. Dey, A. K. Singh, D. K. Prasad, and K. D. Mcdonald-Maier, "Ironman: An approach to perform temporal motionless analysis of video using cnn in mpsoc," *IEEE Access*, vol. 8, 2020.

[20] G. Kalliatakis *et al.*, "Detection of human rights violations in images: Can convolutional neural networks help?" *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 5: VISAPP*, 2017.

[21] G. Zhang *et al.*, "Forecasting with artificial neural networks:: The state of the art," *International journal of forecasting*, vol. 14, no. 1, pp. 35–62, 1998.

[22] "The 25 worst passwords of 2017," http://fortune.com/2017/12/19/the-25-most-used-hackable-passwords-2017-star-wars-freedom, accessed: 2018-01-31.

[23] "The 25 most popular passwords of 2018 will make you feel like a security genius," https://gizmodo.com/the-25-most-popular-passwords-of-2018-will-make-you-fee-1831052705, accessed: 2018-01-31.

[24] V. Rijmen and J. Daemen, "Advanced encryption standard," *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology*, pp. 19–22, 2001.

[25] "Odroid-xu4," https://goo.gl/KmHZRG, accessed: 2018-07-23.

[26] 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*, 2016, pp. 770–778.

[27] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.

[28] A. Krizhevsky *et al.*, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012.

[29] S. J. Pan *et al.*, "A survey on transfer learning," *IEEE TKDE*, vol. 22, no. 10, 2009.

[30] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[31] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[32] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.

[33] "Exynos 5 octa (5422)," https://www.samsung.com/exynos, accessed: 2018-07-23.

[34] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1449–1457.

[35] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.

[36] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.