IoTAI 2024

The First International Conference on IoT-AI

ISBN: 978-1-68558-183-1

June 30 - July 4, 2024

Porto, Portugal

IoTAI 2024 Editors

Yasushi Kambayashi, Sanyo-Onoda City University, Japan

IoTAI 2024

Forward

The First International Conference on IoT-AI (IoTAI 2024), held on June 30 – July 4, 2024 focused on blending AI and IoT (Applied intelligence) to various domains.

Joining Artificial Intelligence (AI) and Internet of Thinks (IoT) is a technical convenience of complementary capabilities. IoT deals with devices interacting using the Internet, while AI makes the devices learn from their data and experience. Almost all domains are greatly benefiting from the marriage IoT-AI for processing high volumes of real-time data. The myriad of IoTs deserves a careful data selection, data patterns identification, controlled frequency for data gathering, high data quality, and appropriate filtering mechanisms.

In essence, by using AI principles and AI-based tools, IoT networks and devices can learn from past decisions, predict future activity, and continuously improve performance and decision-making capabilities. The successful combination of AI and IoT leverages the quality of real data to benefit system customers.

We take here the opportunity to warmly thank all the members of the IoTAI 2024 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to IoTAI 2024. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions.

We also thank the members of the IoTAI 2024 organizing committee for their help in handling the logistics and for their work that made this professional meeting a success.

We hope that IoTAI 2024 was a successful international forum for the exchange of ideas and results between academia and industry and to promote further progress in the area of AI and IoT. We also hope that Porto provided a pleasant environment during the conference and everyone saved some time to enjoy the historic charm of the city.

IoTAI 2024 Chairs

IoTAI 2024 Steering Committee

Narito Kurata, Tsukuba University of Technology, Japan
Yasushi Kambayashi, Sanyo-Onoda City University, Japan
Bharath Sudharsan, General Motors, Ireland
Christoph P. Neumann, Ostbayerische Technische Hochschule Amberg-Weiden, Germany
Young-Joo Suh, POSTECH, South Korea
Marko Jäntti, University of Eastern Finland, Finland

IoTAI 2024

Committee

IoTAI 2024 Steering Committee

Narito Kurata, Tsukuba University of Technology, Japan
Yasushi Kambayashi, Sanyo-Onoda City University, Japan
Bharath Sudharsan, General Motors, Ireland
Christoph P. Neumann, Ostbayerische Technische Hochschule Amberg-Weiden, Germany
Young-Joo Suh, POSTECH, South Korea
Marko Jäntti, University of Eastern Finland, Finland

IoTAI 2024 Technical Program Committee

Prashant Anantharaman, Narf Industries, USA
Yang Bai, University of Maryland College Park, USA
Grigorios N. Beligiannis, University of Patras - Agrinio Campus, Greece
Zoran Bojkovic, University of Belgrade, Serbia
Hongmei Chi, Florida A&M University, USA
Kuan-Lin Chiu, Columbia University, USA
Giulia Cisotto, University of Milan-Bicocca, Milan, Italy
Volkan Dedeoglu, CSIRO, Australia
Declan Delaney, University College Dublin, Ireland
Raffaele Della Corte, Federico II University of Naples, Italy
Yepeng Ding, The University of Tokyo, Japan
Smain Femmam, UHA University France, France
Vasileios Gkioulos, NTNU, Normay
Myriam Guedey, University of Applied Sciences Stuttgart, Germany
Hariyanto Gunawan, Chung Yuan Christian University, Taiwan
Kun Hua, Lawrence Technological University, USA
Mauro Iacono, Università degli Studi della Campania "Luigi Vanvitelli", Italy
Murat Isik, Stanford University, USA
Essa Imhmed, Eastern New Mexico University, USA
Razib Iqbal, Missouri State University, USA
Yasser Ismail, Southern University and A&M College, USA
Rossitza Ivanova Goleva, New Bulgarian University, Bulgaria
Marko Jäntti, University of Eastern Finland, Finland
Yasushi Kambayashi, Sanyo-Onoda City University, Japan
Alvi Ataur Khalil, Florida International University, USA
Zaheer Khan, University of the West of England, Bristol, UK
Shivanjali Khare, University of New Haven, USA
Narito Kurata, Tsukuba University of Technology, Japan
Mikel Larrea, University of the Basque Country UPV/EHU, Spain
Yin Li, Cornell University, USA
Kai Lin, eBay Inc., USA
Frederico Lopes, Metropole Digital Institute | Federal University of Rio Grande do Norte, Brazil

Giuseppe Loseto, LUM "Giuseppe Degennaro" University, Italy
Massimo Marchiori, University of Padua, Italy / European Institute for Science, Media and Democracy, Belgium
Michele Mastroianni, Università degli studi della Campania “Luigi Vanvitelli”, Italy
Javier Medina Quero, University of Jaen, Spain
Zewei Mo, University of Pittsburgh, USA
Sumali Morapitiya, General Sir John Kotelawala Defence University, Ratmalana, Sri Lanka
Óscar Mortágua Pereira, University of Aveiro, Portugal
Gyu Myoung Lee, Liverpool John Moores University, UK
Christoph P. Neumann, Ostbayerische Technische Hochschule Amberg-Weiden, Germany
Klimis Ntalianis, University of West Attica, Greece
Bogdan Oancea, University of Bucharest, Romania
Kartik Palani, iManage LLC, USA
Prasoon Patidar, Carnegie Mellon University, USA
Flavio Oquendo, IRISA (UMR CNRS) - University of South Brittany, France
Addisson Salazar, Universitat Politècnica de València, Spain
Sandeep Singh, Abacus.AI, USA
Joana Sousa, NOS Inovação S.A., Portugal
Bharath Sudharsan, General Motors, Ireland
Young-Joo Suh, POSTECH, South Korea
Chibuzo Ukegbu, Boise State University, USA
Harsh Vardhan, Vanderbilt University, Nashville, USA
Dimiter Velev, University of National and World Economy, Bulgaria
Gaurav Verma, Stony Brook University, USA
Stefanos Vrochidis, Information Technologies Institute Centre for Research and Technology Hellas, Greece
Ching-Nung Yang, National Dong HwaUniversity, Taiwan
Xiangchen Zhao, Pure Storage Inc., USA

Copyright Information

Table of Contents

Identifying Vulnerable Third-party Components in IoT Firmware Using Deep Learning

Chia-Mei Chen, Sheng-Hao Lin and Zheng-Xun Cai
Department of Information Management
National Sun Yat-Sen University
Kaohsiung, Taiwan

Gu-Hsin Lai
Department of Technology Crime Investigation
Taiwan Police College
Taipei, Taiwan

Ya-Hui Ou
General Competency Center
National Penghu University
Penghu, Taiwan

***Abstract*—To reduce the development time and the cost of Internet of Thing (IoT) products, vendors leverage Third-Party Components (TPCs) to manufacture various types of IoT products. However, such third-party software might not be validated with proper software testing or might contain vulnerabilities. Furthermore, existing research rarely proposed a cross-architecture solution for detecting both top IoT vulnerabilities. Therefore, this study proposes a cross-architecture IoT vulnerability detection method that identifies vulnerable third-party components used in IoT firmware. This study leverages a Siamese Neural Network (SNN) architecture and designs a similarity algorithm to identify vulnerable functions on different processor architectures. The evaluation results demonstrate that the proposed method can identify vulnerable TPCs effectively.**

Keywords- IoT attacks; vulnerability detection; deep learning.

I. INTRODUCTION

With the prevalence of the Internet of Things (IoT) and its flourishing development, about 83% of organizations rely on IoT technologies to boost their productivity [1]. IoT devices are becoming increasingly ubiquitous. In 2025, the IoT market is expected to reach 27 billion active connections [2], and the IoT-related services will grow to 58 billion dollars [3]. Such network-connected devices in enterprise networks are hard for security teams to properly identify and monitor them, which may become security blind spots for the organizations.

To reduce development costs and to shorten the time to market for a new device, the functionalities of the IoT device typically are provided through previously developed software. Furthermore, the current development leverages third-party software heavily to improve development efficiency. The use of TPCs in IoT products tripled during the recent years [4]. Including TPCs in an IoT device implies that the device inherits the vulnerabilities existing within the TPCs. Such external components mostly are not secure, and their vulnerabilities influence IoT security [5]. TPCs play an imperative role in IoT firmware development.

IoT devices commonly adopt embedded Linux systems. Examining such embedded systems requires a comprehensive understanding of the operating systems and the experience of reverse engineering. In addition, manufacturers adopt various processor architectures.

Based on the literature review, past research paid little attention to the vulnerabilities caused by TPCs, so an automatic and effective solution to identify vulnerable TPCs used in IoT firmware is desired. In addition, existing work mostly focused on a single architecture and rarely provided a solution for multiple architectures. Even though the recent work proposed solutions for cross-architecture, their detection performance needed to be improved.

To fill in the aforementioned research gaps, this study designs an automatic firmware analysis and vulnerability detection approach for multi-architecture IoT devices, which integrates several Open-Source Software (OSS) solutions to automate the firmware analysis process and facilitates the state-of-the-art machine learning technologies to extract features of function codes and to identify vulnerable components.

The remainder of this paper is constructed as follows. Section 2 reviews the related research. Section 3 presents the proposed detection method, followed by the performance evaluation in Section 4. The last section draws the conclusion remark and the future directions of this study.

II. LITERATURE REVIEW

To enhance detection performance, some research adopted ensemble approach which aggregates the multiple classifiers. Essa and Bhaya [6] applied two feature selection approaches: mean and hard-voting schemes, with ensemble soft voting classifier. According to the evaluation, their method achieved better results than other ensemble and individual classifiers.

Zhao et al. [7] conducted a large-scale analysis of TPC usage in IoT firmware. During the analysis process, their approach requires several stages of manual work to facilitate the detection. It applied the tool Binwalk [8] for file system decomposition, extracted the features from Control Flow Graphs (CFGs), and utilized the version check to determine if a target TPC contains vulnerabilities. Even though this past work involved intensive human analysis effort, it provided statistical analysis results and highlighted the IoT cyber risk.

Ngo et al. [9] reviewed the existing IoT malware detection work based on static analysis and pointed out that most existing solutions only detect malware in a single architecture. They summarized commonly used static features including function call graphs, CFGs, operation codes (opcodes), strings, and file headers.

A control flow graph is a directed graph $G(V, E)$ that represents all the possible execution paths of a code piece, where V is a set of basic execution blocks and E is a set of edges representing the connections between these basic blocks. Recent work adopted a CFG variation that contains extra information about a code piece. An Attributed Control Flow Graph (ACFG), *G(V, E, M)*, is an extended version of CFG, where *M* is the labeling function that maps a basic execution block in *V* to a set of attributes in *A*. The attribute set *A* can be tailored to capture the semantic meaning of the basic blocks or to characterize the blocks.

Feng et al. [10] adopted ACFGs to represent binary functions and extracted their statistical and structural attributes, such as the number of calls, the number of instructions, and the number of offspring. They utilized the bipartite graph matching algorithm to measure the similarity of two ACFGs.

Xu et al. [11] improved the previous work by employing the structured data embedding technique Structure2Vec to transfer ACFGs into feature spaces. They utilized a large-scale training dataset obtained from compiling the same source code on different architectures and with different compiler optimization techniques to train an SNN for code similarity detection. Sun et al. [12] also concluded that ACFGs can capture relevant features of binary codes and SNNs measure cross-architecture code similarity efficiently.

Feature selection plays an important role in the construction of efficient detection classifiers. The literature review concluded that most existing solutions only detect malware in a single architecture and commonly adopted static features. An improved CFG, ACFG, can extract better semantic features to represent the algorithm, instead of platform-dependent features. Therefore, it is suitable for detecting cross-platform TPCs.

III. METHODOLOGY

The proposed solution is outlined in Figure 1. As mentioned above, the tools for firmware image decomposition and binary analysis are available, but some are unstable. After a preliminary investigation, this study selects reliable tools (Binwalk [8] and Angr [13]) to automate the binary code extraction process.

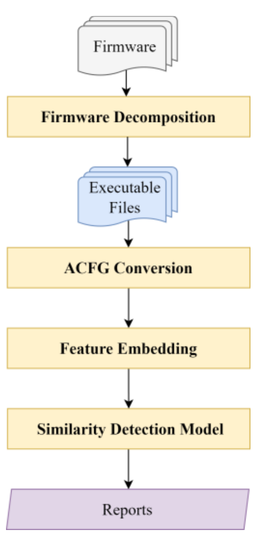

Figure 1. The proposed system architecture.

The Firmware Decomposition module applies Binwalk to decompose firmware images. The module identifies file types through the file header and then extracts executable files for vulnerable TPC inspection. The executable files serve as inputs for the ACFG Conversion module are converted into ACFGs, where the conversion applies the binary analysis tool Angr to convert binary codes into ACFGs. The Feature Embedding module adopts Struc2Vec [14] to encode the graphic structure features of ACFGs, and the detection module applies a SNN to compare the similarity of two embedded ACFGs: a known vulnerable TPC and a function code to be examined.

A. *Firmware Decomposition*

The Firmware Decomposition module utilizes Binwalk to disassemble firmware and retrieves the file system and image files. IoT firmware mostly uses the file system SquashFS to compress a Linux operating system, where most research applied Binwalk to perform the decompression. This module searches for all the file directories recursively, starting from the root directory. The file format Executable Linkable Format (ELF) is a common standard file format for executable files, object codes, and shared libraries. By using the ELF information, this module identifies the above executable files to be inspected.

B. *ACFG Conversion*

The ACFG Conversion module applies the tool Angr [13] to convert executables into ACFGs. Angr decomposes a target executable file into binary functions and then converts each binary function into its corresponding ACFG, where a basic block in ACFGs uses program control-related instructions, such as Jump, Call, or Return, as an edge connecting to another basic block. In other words, the ACFG represents the target binary function. Angr translates binary codes of different processor architectures in a set of intermediate instructions so that opcodes from different architectures map to the same or similar intermediate instructions. Based on Angr's instruction set, this study defines two types of relevant instructions: data movement and arithmetic & logic, to represent the functionaflity of a basic block.

C. *Feature Embedding*

This study applies Struc2Vec [14] to embedding the structural feature of ACFG and the features of a basic block in the graph, namely to encode the ACFG features. Struc2Vec learns latent representations that capture the structural identity of nodes in a graph. It measures node similarity hierarchically at different scales, constructs a multilayer graph to encode structural similarities, and generates the embeddings for nodes.

Struc2Vec consists of the following main steps to learn latent representations for a graph *G(V, E)*. (1) It determines the structural similarity between each node pair in the graph for different neighborhood sizes, where the structural distance is explained in the following paragraph. In this way, it measures node similarity hierarchically at different scales. (2)

Copyright (c) IARIA, 2024. ISBN: 978-1-68558-183-1

TABLE I. THE DETECTION RESULTS OF THE OPENSSL VULNERABILITY.

Firmware	Executable file	Vulnerable function	Similarity
DAP-1562	wpa_supplicant	ssl3_get_new_session_ticket	0.8913
E1550USB-NVRAM60K	libssl.so.1	ssl3_get_new_session_ticket	0.8602
E4200USB-NVRAM60K	libssl.so.1	ssl3_get_new_session_ticket	0.8602
ddwrt.v24-23838	openvpn	ssl3_get_message	0.8491
E3000USB-NVRAM60K	libssl.so.1	ssl3_get_message	0.8133
K26-1.28.RT-MIPSR1	libssl.so.1	ssl3_get_new_session_ticket	0.8084
Tomato-K26USB-1.28RT-N5X	libssl.so.1	ssl3_get_new_session_ticket	0.8084

It constructs a weighted multi-layer graph hierarchically, where each layer corresponds to a level of the hierarchy in measuring structural similarity. (3) it applies a biased random walk to generate node sequences, where the sequences include more structural similarity of the graph. (4) It learns the latent representation from the node sequences.

D. Similarity Detection

The above feature embedding module encodes ACFGs into graph-embedded networks. This study employs a SNN to learn the similarity of two ACFGs, where the two networks in the SNN are designed as ACFG graph embedded networks and convert the input, a pair of ACFGs, into vectors.

To detect binary codes of cross-architecture, the binary functions of the same category have similar control flows and similar ACFGs. Hence, the loss function is defined as the cosine distance of the two vectors.

IV. SYSTEM EVALUATION

This study designs the experiment that investigates the performance of the proposed system in two aspects: the success rate of firmware decomposition and detection efficiency, as automatic decomposition is as important as vulnerability detection. It collects 50 firmware samples from two sources (FirmAE [15] and DD-WRT [16]) in order to examine diversified IoT devices.

This experiment employs AUC to evaluate the model efficiency, as it counts as a measure of the ability of the detection model to distinguish between the classes. A high AUC implies that the model distinguishes between the positive and negative classes efficiently. The AUC values between 0.9~1 are considered excellent; those between 0.8-0.9 are good; those between 0.7-0.8 are fair; those between 0.6-0.7 are poor. Figure 2 illustrates the ROC curves. The evaluation results show that the proposed method yields AUC-ROC value of 0.971 and achieves efficient detection performance.

For inspecting the detection performance on cross-platform, this experiment targets OpenSSL vulnerabilities in multiple architecture environments. The proposed system can automatically decompose firmware images with a high

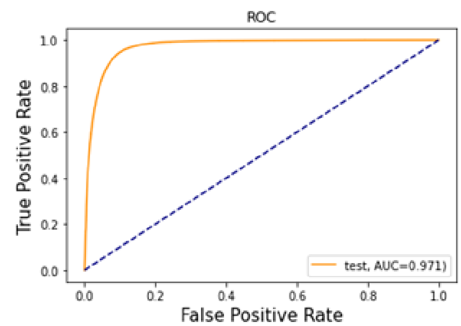

Figure 2. AUC-ROC

success rate of 94%. Table 1 lists the detected vulnerable firmware and functions that contain the OpenSSL vulnerabilities. The proposed detection method identifies the vulnerabilities with high similarity scores over multiple architecture environments, which implies that the proposed system can identify vulnerable functions effectively.

V. CONCLUSION

This study proposes a cross-architecture vulnerability detection method for IoT devices that discovers vulnerable TPC functions. Most past studies address one of the above IoT cyber risks, which motivates this research to discover these two critical IoT vulnerabilities. By evaluating the proposed system with real-world firmware images, the experimental results show that the proposed method can identify efficiently the aforementioned IoT vulnerabilities.

The proposed method employs the graphical representation ACFG to represent the semantic meaning of a binary function and the graph embedding technique Struc2Vec to extract the structural and code features of a function code. It applies a Siamese network to identify a function. The evaluation demonstrates that the proposed solution can identify the top IoT vulnerabilities mentioned above effectively. The real-world case studies demonstrate that the proposed system can automatically decompose firmware effectively.

Future work directions can investigate hybrid analysis approaches or extract more features from function codes to

 ISBN: 978-1-68558-183-1

improve detection performance. Furthermore, investigating other similarity detection models is another possible research direction as well. IoT devices are prevailing. Hence, reducing IoT cyber risks can improve network security for businesses and users.

REFERENCES

[1] S. Bennett. "IoT security statistics 2023 - Everything you need to know." [Online]. Available from: https://webinarcare.com/best-iot-security-software/iot-security-statistics/ [retrieved: May, 2024].

[2] M. Hasan. "State of IoT 2022: Number of connected IoT devices growing 18% to 14.4 billion globally." [Online]. Available from: https://iot-analytics.com/number-connected-iot-devices/ [retrieved: May, 2024].

[3] Gartner. "Forecast: IT services for IoT, worldwide, 2019-2025." [Online]. Available from: https://www.gartner.com/en/documents/4004741 [retrieved: May, 2024]

[4] F. Baldassari. "IoT developers can't afford to ignore third-party code." [Online]. Available from: https://www.techtarget.com/iotagenda/post/IoT-developers-cant-afford-to-ignore-third-party-code [retrieved: May, 2024].

[5] Micro.ai. "The risk of using third-party components in IoT devices." [Online]. Available from: https://micro.ai/blog/the-risk-of-using-third-party-components-in-iot-devices [retrieved: May, 2024].

[6] H. A. Al Essa and W. S. Bhaya, "Ensemble learning classifiers hybrid feature selection for enhancing performance of intrusion detection system," Bulletin of Electrical Engineering and Informatics, vol. 13, no. 1, pp. 665-676, 2024.

[7] B. Zhao et al., "A large-scale empirical analysis of the vulnerabilities introduced by third-party components in IoT firmware," Proceedings of the 31st ACM SIGSOFT International Symposium on Software Testing and Analysis, 2022, pp. 442-454.

[8] ReFirm Labs. "Binwalk." [Online]. Available from: https://github.com/ReFirmLabs/binwalk [retrieved: May, 2024].

[9] Q.-D. Ngo, H.-T. Nguyen, V.-H. Le, and D.-H. Nguyen, "A survey of IoT malware and detection methods based on static features," ICT Express, vol. 6, no. 4, pp. 280-286, 2020/12/01/ 2020, doi: https://doi.org/10.1016/j.icte.2020.04.005.

[10] Q. Feng, R. Zhou, C. Xu, Y. Cheng, B. Testa, and H. Yin, "Scalable graph-based bug search for firmware images," Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 480-491.

[11] X. Xu, C. Liu, Q. Feng, H. Yin, L. Song, and D. Song, "Neural network-based graph embedding for cross-platform binary code similarity detection," Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, 2017, pp. 363-376.

[12] H. Sun, Y. Tong, J. Zhao, and Z. Gu, "DVul-WLG: Graph Embedding Network Based on Code Similarity for Cross-Architecture Firmware Vulnerability Detection," Cham, 2021: Springer International Publishing, in Information Security, pp. 320-337.

[13] Y. Shoshitaishvili et al., "Sok:(state of) the art of war: Offensive techniques in binary analysis," 2016 IEEE Symposium on Security and Privacy (SP), 2016: IEEE, pp. 138-157.

[14] L. F. Ribeiro, P. H. Saverese, and D. R. Figueiredo, "struc2vec: Learning node representations from structural identity," Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017, pp. 385-394.

[15] M. Kim, D. Kim, E. Kim, S. Kim, Y. Jang, and Y. Kim, "Firmae: Towards large-scale emulation of iot firmware for dynamic analysis," Annual Computer Security Applications Conference, 2020, pp. 733-745.

[16] DD-WRT. "Router Database." [Online]. Available from: https://dd-wrt.com/support/router-database/ [retrieved: May, 2024].

Development of AI Learning Materials Using Physical Computing

Toshiyasu Kato
Department of Information and Media Engineering
Nippon Institute of Technology
Minamisaitama-gun - Saitama, Japan
email: katoto@nit.ac.jp

Yuto Chino
Department of Technology
Fuchu City Fuchu 6th Junior High School
Fuchu - Tokyo, Japan
email: edu.yutochino@gmail.com

***Abstract*—AI services, including generative AI, have become widespread globally. We are using Artificial Intelligence daily. However, without proper knowledge, users may not achieve the desired results, and there is a risk of inaccuracy. Educational institutions are beginning to establish the groundwork for AI learning. Due to broad learning standards, however, there is few educational materials that cover the fundamental knowledge and skills of using AI. To address this problem, we have developed educational materials that enable basic learning about the mechanisms of AI and motivate learning. For this purpose, we utilize physical computing. This paper reports on the process from the composition of learning standards to the development of educational materials.**

Keywords-AI learning materials; physical computing; learning standards.

I. INTRODUCTION

The proliferation of Artificial Intelligence (AI) services, including generative AI, has made artificial intelligence a familiar presence worldwide. A study conducted by a research group of Massachusetts Institute of Technology involved a task where participants used ChatGPT, one of the generative AIs, to write texts specialized in their areas of expertise. The results showed that the group using ChatGPT reduced the average time required by 40% and increased the quality of output by 18% [1]. In Japan, the Cabinet Office has committed to educational reforms by defining "Mathematics, Data Science, and AI" as the new basics of reading, writing, and arithmetic for the digital society in its AI Strategy 2019 [2]. Acquiring AI literacy is becoming indispensable for thriving in the digital society.

However, there are few examples of educational materials that enable the learning of foundational AI knowledge and skills. We can observe only a few classroom practices in middle school technology and high school industrial arts courses [3] [4]. Consequently, there is a lack of materials that facilitate active learning by students. Furthermore, although practical lessons for acquiring AI literacy are being conducted in primary and secondary education, there is an insufficient learning foundation for instructors.

Thus, this study focuses on developing physical computing educational materials intended for university students who have some experiences with using computers. The rationale for incorporating physical computing is that it has been used as an accessible teaching method for beginners in programming education [5]. Physical computing allows learners to perceive errors through physical movements. By utilizing physical computing materials, the authors have found that students easily acquire of AI literacy.

Section 2 describes related research, Section 3 describes the proposed teaching materials, Section 4 examines the teaching materials, and Section 5 concludes our discussion.

II. RELATED WORK

Scratch is a well-known programming learning material. Scratch is extensible and now it includes materials focused on machine learning. An example is ML2Scratch, which enables image classification using MobileNet through TensorFlow.js [6]. Furthermore, based on this study, researchers have developed another extension that allows for the learning of advanced deep learning techniques such as transfer learning [7].

Google's Teachable Machine is a web-based tool that easily allows to create machine learning models [8]. It supports to create models of three categories, i.e., image, sound, and pose. We can create and export a TensorFlow.js models through collecting learning data directly on the site by taking pictures or recording sounds, and with the press of a training button. A research at the University of Potsdam has shown that utilizing physical computing educational materials promotes not only intrinsic motivation but also creative and constructive learning [9].

Materials using Scratch are web-based, resulting in outcomes being displayed on the screen, akin to the initial experiences of text display in programming learning. Teachable Machine specializes in model creation. While exporting models allows for a broad range of learning opportunities, advancing in applied learning requires prior knowledge of the application areas. A commonality among these materials is their use of transfer learning, which tends to produce relatively accurate results. Although it is easy to obtain results from machine learning through these examples, they do not help for deepening knowledge. We address this problem.

III. SUGGESTED AI LEARNING MATERIALS

In this study, we have carried out the following steps for the development of our educational materials.

1. Establish learning standards for AI literacy based on literatures.
2. Develop physical computing educational materials modeled on autonomous driving, based on the established learning standards.
3. Verify whether the materials can be used for learning.

ISBN: 978-1-68558-183-1

4. After verification, conduct an evaluation experiment of the proposed materials to assess their effectiveness in improving awareness and knowledge related to AI. (Planned for the future)

A. Developing Learning Standards for AI Literacy

In this study, we have established learning standards necessary for acquiring AI literacy, aiming to experiential learning for everyday use of AI. To define the learning standards, we have investigated models of curricula recommended by consortia dedicated to strengthening education in mathematics, data science, and AI, as well as the G exam, a Japanese certification that tests knowledge of machine learning [10] [11]. This approach ensures that the learning standards cover essential aspects of AI and responsibilities using AI technologies in daily life.

1) Mathematics/Data Science/AI Model Curriculum

We employed the "Mathematics, Data Science, & AI (Literacy Level) Model Curriculum – Cultivating Data Thinking" for setting the AI learning standards [10]. The learning objective of this curriculum is defined as "to proactively acquire the foundational proficiency necessary to proficiently apply mathematics, data science, and AI in daily life, work, and other scenarios." The emphasis is on "the capability to make appropriate, human-centered decisions." The fundamental approach includes "a focus on teaching the 'joy' and 'significance' of engaging with and learning about mathematics, data science, and AI." The curriculum orderly presents learning items and is systematically structured as shown in Table 1. Within this structure, the areas related to AI learning include sections 1-3, 1-4, 1-5, 1-6, and 3-1.

In the first section, "Utilization of Data & AI in Society," the focus is mainly on AI knowledge and application, presenting skill sets for specialized AI and general-purpose AI, among others. The second section "Data Literacy" discusses how to handle data, but it hardly mentions AI, hence we do not focus on this study. The third section, "Considerations in the Utilization of Data & AI," suggests covering negative examples of AI utilization and data ethics, among other topics.

TABLE I. STRUCTURE OF AI LITERACY LEVEL MODEL CURRICULUM

1. Introduction ***Utilization of data and AI in society***	1-1 Changes occurring in society
	1-2 Data used in society
	1-3 Application areas of data and AI
	1-4 Technology for data/AI utilization
	1-5 Fields of data/AI utilization
	1-6 Latest trends in data and AI utilization
2. Basic ***Data literacy***	2-1 Read the data
	2-2 Explain the data
	2-3 How to use data
3. Knowledge Considerations in the Utilization of Data & AI	3-1 Points to note when handling data and AI
	3-2 Points to note when protecting data

2) DLA Deep Learning For GENERAL

In order to survey the required knowledge of machine learning, we investigated the official textbook for the Deep Learning G Certification, "Deep Learning G Certification Official Textbook 2nd Edition," which is structured according to the syllabus of the qualification examination. The official textbook is comprised of seven chapters, with contents as follows:

1. What is Artificial Intelligence (AI)?
2. Trends Surrounding Artificial Intelligence.
3. Issues in the Field of Artificial Intelligence.
4. Specific Methods of Machine Learning.
5. Overview of Deep Learning.
6. Methods of Deep Learning.
7. Toward the Social Implementation of Deep Learning.

We have focused on Chapters 1, 2, and 7. Chapter 1 discusses the nature of AI, including its history and classification, and explains the differences between machine learning and deep learning at various levels. Chapter 2 addresses the trends in AI, emphasizing the history and relationship of machine learning and deep learning research. It particularly notes that desirable results can be achieved through accumulating data in machine learning and explains the mechanisms of machine learning and deep learning differ, and how they are different. Chapter 7 covers methods and considerations for utilizing AI towards social implementation. The chapter also discusses how to handle data, including the quality of datasets. It emphasizes how to eliminate bias, and how to process and to analyze data fairly, and how to learn regularity from data.

Based on the observation of Chapters 1 and 2, we have formulated the foundational learning criteria and perspectives on AI, which are presented in Table 2.

TABLE II. FOUNDATIONAL LEARNING CRITERIA AND PERSPECTIVES ON AI

	Learning Criteria & Perspectives	***Points of Understanding***
A	Generality & Specificity	Specializes in performing certain tasks (e.g., image & voice recognition)
B	Learning & Training Data	The operation of AI is indispensable for learning data, with the quality and quantity of data being crucial
C	Validity of Inference Results	The quantity and quality of training data can affect achieving the desired results

The rationales for formulating each perspective are as follows:

A. From the perspective of "human-centered" importance in mathematics, data science, and AI, it is necessary to learn about what AI can and cannot do.

B. As handled in prior research and teaching practices, approaches to collecting learning data for image recognition, the emphasis on the consciousness of statistical work for data utilization in mathematics, data science, and AI, and the G Certification's point on the necessity of processing, analyzing, and learning the training data for AI's social implementation are reasons for this perspective.

 ISBN: 978-1-68558-183-1

C. The G Certification mentions that desirable results can be achieved depending on the quantity and quality of data, underlining the necessity to understand that the desired outcomes may not always be attainable depending on the data.

B. *Physical Computing Teaching Materials*

We developed educational materials for experiential physical computing that allow students for comprehensive learning of the established learning criteria and perspectives. The goal of these materials is to motivate AI learning and enable active learning.

1) *Specifications of the Educational Material*

In this research, we developed a mobile robot-like educational material that can recognize signs through image recognition using the Jetson Nano B01, a single-board computer for AI learning released by NVIDIA [12]. It controls the robot according to the meaning of the signs. Figure 1 shows the developed robot-like educational material, Figure 2 shows the hardware configuration. The robot-like educational material communicates with the server using wireless network so that it executes the AI learning model.

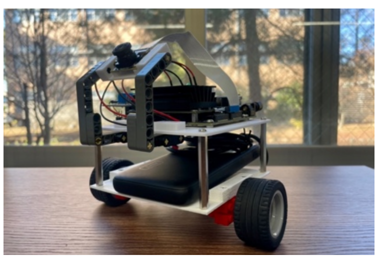

Figure 1. Developed physical computing teaching materials.

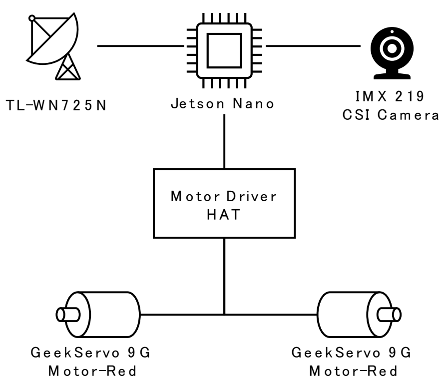

Figure 2. Hardware configuration diagram.

The software configuration is as follows. We employed the JetPack 4.6 platform for Jetson, published by NVIDIA, and Docker containers used by the NVIDIA Deep Learning Institute. They allow operators to access Jupyter Lab via a browser. Additionally, we used pre-installed PyTorch, which is the machine learning library used in this container.

2) *Overview of the Robot-like Educational Material*

This robot-like educational material recognizes signs and proceeds according to the meaning of those signs through image classification using Convolutional Neural Networks (CNN). The material developed for this occasion classifies two classes (background and signs). We conducted experiments utilizing a sign indicating a speed limit of 10 km/h to slow down the operational speed of the material. Additionally, as an advanced application, there is a program that classifies six classes. Table 3 shows the recognized objects and corresponding actions.

TABLE III. CORRESPONDENCE TABLE OF RECOGNIZED OBJECTS AND ACTIONS

Recognition Objects	***Actions***
Background	Normal operation
Speed limit 10km	Operating speed 10
Speed limit 30km	Operating speed 30
Stop	Pause (1 second)
No entry	allowed End of operation
People	Stop until no more people are classified

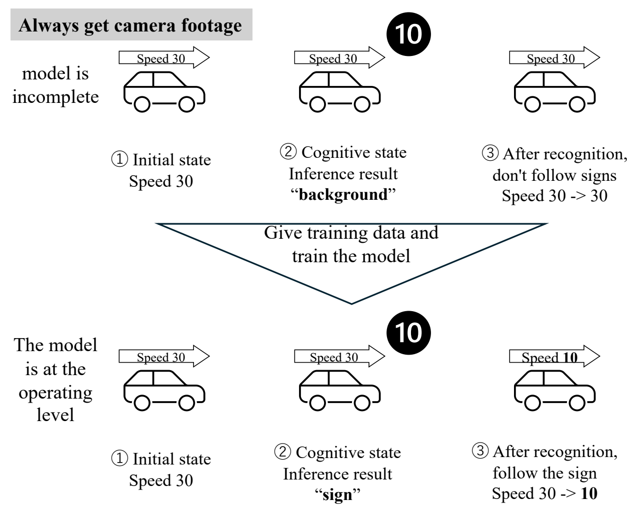

Figure 3. Operational Image of the Educational Materials.

Figure 3 shows the conceptual image of the operations of the robot-like educational material.

IV. VERIFICATION OF EDUCATIONAL MATERIALS

Learners can perform a series of AI learning activities by accessing Jupyter Lab via a browser. He or she must perform the following procedures.

1. Prepare the learning data.
2. Define the model.
3. Train the model.
4. Test the model.
5. Adjust the data according to the results.

First, the learner performs the step 1 through 4. Of course, the robot-like educational material cannot classify signs, and executes incorrect actions. Then, the learner proceeds to step 5 to adjust training data and the frequency of training sessions. Then he or she iterates the procedure steps 3, 4 and

 ISBN: 978-1-68558-183-1

5 until the robot-like educational material achieves the accurate inference. Figure 4 shows the experiments of this procedure.

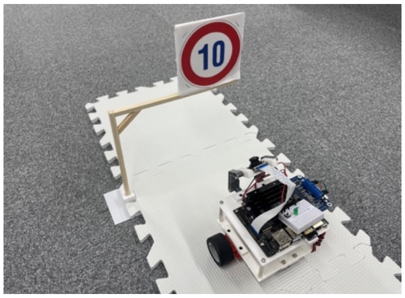

Figure 4. Operational Aspect of the Proposed Educational Material.

In step 3, the robot-like educational material collects learning data through a camera mounted on it.

For step 5, adjusting the training data, learners individually modify the learning data and model training. Adjusting the learning data involves increasing the data volume based on the operational results. For the model training, we increase the number of learning iterations until the loss is stabilized, since the system presents the number of epochs and the loss graph. After adjustments, learners check the accuracy of the model through the operation of the robot-like educational material. Figure 5 shows the control panel of the system. The learners can adjust the data and learning.

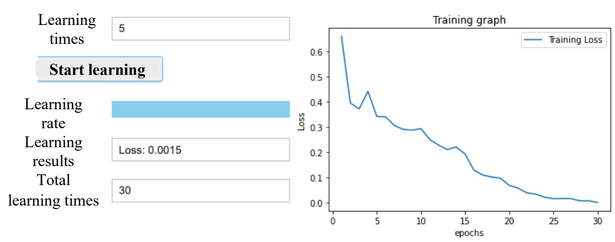

Figure 5. Adjustment of Model Training.

This teaching material is built based on the official PyTorch tutorial. Learners can observe how the model training progresses using the control panel shown in Figure 5 without programming.

The proposed educational materials incorporate image recognition. We must investigate whether the learners can comprehend the learning standards through the experiences of image recognition learners. We made learners engage in an AI experience focused on character recognition (image recognition) aligned with the learning standards to assess their adequacy.

TABLE IV. VALIDITY OF LEARNING CRITERIA FOR IMAGE RECOGNITION N=3

Learning Criteria	*Understood*	*Not Understood*
A	3	0
B	2	1
C	3	0

Following this experience, Table 4 shows the collected responses on the comprehensibility of each learning standard. We have confirmed that the learners deeply comprehend the learning standards we proposed by studying image recognition.

V. CONCLUSION AND FUTURE WORK

We have investigated learning standards for acquiring AI literacy and have created robot-like educational materials to help learners to comprehend the machine learning based on these standards. The validation of these learning standards revealed that learners enhanced their understanding of AI literacy through image recognition. Future efforts will focus on conducting evaluation experiments to assess the educational impact of the proposed materials.

ACKNOWLEDGMENT

This work was partially supported by JSPS KAKENHI, Grant Number JP24K06237.

Yasushi Kambayashi gave us useful comments. We appreciate them very much.

REFERENCES

[1] S. Noy, and W. Zhang, "Experimental evidence on the productivity effects of generative artificial intelligence," Science 381.6654 (2023): pp.187–192.

[2] Cabinet Office: AI Strategy 2019. [Online]. Available from: https://www8.cao.go.jp/cstp/ai/aistratagy2019.pdf 2023.12.23

[3] N. Takezawa, T. Yamamoto, and H. Koura, "Proposal and Evaluation of a Teaching Process for Learning Image Recognition Technology Using Artificial Intelligence through Programming," Japan Society of Educational Information, vol.38, no.1, pp.37–48, 2022.

[4] N. Mukaida, "Examination of Classes to Think About How to Face AI in AI Literacy Education," Japan Educational Research Society of the AI era, vol.5, pp.9–15, 2022.

[5] P. Mareen and R. Ralf, "Impact of Physical Computing on Learner Motivation," Koli Calling '18, pp.1–10, 2018.

[6] J. Ishihara, "Introducing an extension that makes machine learning available from Scratch," Interface, vol.47, no.5, CQ Publishing, pp.148-152, 2021.

[7] T. Yagi, and T. Yamaguchi, "Building a machine learning experience system for beginners using Scratch, " Bulletin of Edogawa University, vol.30, pp.473-485, 2020.

[8] Google: Teachable Machine [Online]. Available from: https://teachablemachine.withgoogle.com/ 2024.01.17

[9] Mareen Przybylla, and Ralf Romeike, "Impact of Physical Computing on Learner Motivation, " Koli Calling 2018, pp.1-10 (2018)

[10] Consortium for Strengthening Mathematics and Data Science Education in Japan, "Mathematics/Data Science/AI (Literacy Level) Model Curriculum -Cultivating Data Thinking-, " [Online]. Available from: http://www.mi.u-tokyo.ac.jp/consortium/pdf/model_literacy.pdf 2024.01.04

[11] Japan Deep Learning Association, "Deep Learning G Test Official Text, " 2nd edition, 403p, 2021.

[12] NVIDIA, "Jetson Nano developer kit, " [Online]. Available from: https://www.nvidia.com/ja-jp/autonomous-machines/embedded-systems/jetson-nano-developer-kit/ 2024.01.04

 ISBN: 978-1-68558-183-1

Development of a UAV-based Disaster Evacuation Support System: an Interim Report

Yuta Sekiguchi
Graduate School of Engineering
Nippon Institute of Technology
Saitama, Japan
email: 2238007@stu.nit.ac.jp

Toshiyasu Kato
Department of Information Technology and Media Design
Nippon Institute of Technology
Saitama, Japan
email: katoto@nit.ac.jp

Munehiro Takimoto
Department of Information Sciences
Tokyo University of Science
Noda, Japan
email: mune@rs.tus.ac.jp

Yasushi Kambayashi
Department of Informatics and Data Science
Sanyo-Onoda City University
Onoda, Japan
email: yasushi@rs.socu.ac.jp

***Abstract*— We have developed a system that supports evacuation after disasters using Unmanned Aerial Vehicles (UAVs). We focus on addressing disasters in mountainous areas. We assume that the information infrastructure is paralyzed. We have two types of UAVs for this system. One type of UAVs, which we call monitoring UAVs, conducts road condition checks and collects information about affected individuals, while another type of UAVs, which we call guiding UAVs, guides the evacuees. The guided route is not simply the shortest path. The UAV selects the most suitable route based on the factors such as route safety and congestion levels. The monitoring UAV that grasps the situation informs the guiding UAVs of the route status as needed, allowing the guiding UAVs to dynamically adjust the evacuation route. Both UAVs share information by exchanging messages by way of a control server through ad hoc communication through an opportunistic network. We have developed a simulator to demonstrate the effectiveness of our system.**

Keywords- UAV Swarm ; Multi-Agent ; Swarm Robot.

I. INTRODUCTION

Japan faces a wide range of disasters, including earthquakes, typhoons, and tsunamis. Evacuation routes may become blocked due to river flooding, collapsed buildings, or landslides. Therefore, it is important to search for evacuation routes while considering these obstacles. In this paper, we propose an evacuation guidance system with the aim of preventing secondary disasters after the primary disaster. The reason why we focus on the secondary disasters is that evacuees may inadvertently go to the unpassable and dangerous area, resulting in delayed evacuation and exposing evacuees to secondary disasters. Particularly, in sparsely populated areas, such as mountainous regions, evacuees may not have access to information to assess the disaster situation due to the paralysis of communication infrastructure. Therefore, we have developed a system for evacuating using UAVs equipped with opportunistic networks, assuming situations where normal communication infrastructure is unavailable. Figure 1 illustrates the concept of our system.

As part of disaster preparedness, there are studies utilizing multiple UAVs for evacuation assistance. These studies assume that information infrastructure becomes paralyzed during disasters, and multiple UAVs equipped with agents collaborate to establish ad hoc communication networks, aiming to prevent secondary disasters. Efficient evacuation assistance using multiple UAVs in a Mobile Ad hoc Network (MANET) environment has been studied [1].We verified the evacuation assistance system through simulation, replicating the environment during actual disasters. In this study, each evacuee is treated as an independent object, enabling the simulation of individual behavior. This allows us to explore potential challenges that were previously undiscovered.

Figure 1. Conceptual diagram of the evacuation supports system.

In Section II, we review related research on UAV-based evacuation support systems. Section III details our proposed system, including its design and components. Section IV discusses the evacuation route algorithm. Section V covers the system's implementation in a simulator and its performance. Finally, Section VI concludes the paper and suggests future work directions.

II. Related Research

Various evacuation support system researches have been conducted using mobile terminals and UAVs. Katayama et al. are pursuing one of the most notable projects that is an agent-based evacuation guidance support system using a UAV [1] [2]. Their system incorporated Internet of Things (IoT) devices and UAVs to monitor disaster situations and indicate the shortest evacuation route for evacuees while avoiding tsunami effects. Their UAVs mount visible light identification (ID) to guide evacuation routes. Even though the system is excellent, their system is specialized for coastal areas [3]. Our system is similar but is specialized for mountainous areas.

Taga et al. conducted another notable research project [4] [5]. They proposed a system that supports evacuation after the occurrence of a large-scale disaster using multi-agents built on a mobile ad hoc network (MANET). Their aim was to let the evacuees share information on MANET and to provide safe routes to their destinations based on the collected information. Their system is situated in rather crowded urban areas, and our system aims to provide evacuation support in a mountainous area.

Fujisawa and Miwa proposed a real-time disaster information sharing system based on an opportunistic communication for evacuation guidance [6]. We also plan to implement our system based on an opportunistic communication system. As Uno and Kashiyama stated, it is important to evaluate the evacuation behaviors of evacuees in the time of a disaster [7].

For efficient use of UAVs, it is important to provide evacuation routes as short as possible. Patel et al. proposed a Forward Backward Shortest Path (FBSP) evacuation routing algorithm that models the evacuation area as a graph with evacuation places as source nodes, shelter places as destination nodes, roads between places as edges, and evacuation routes as paths [8]. We employ a similar algorithm for our UAV flight planning. In order to achieve accurate and safe guidance for evacuees, we have to consider the moving speed of pedestrians. Rothkrantz and Popa provided efficient routing for pedestrians in a crowded city [9]. Even in mountainous areas, rushed evacuations can make crowded places along the evacuation route. We take into account this fact too.

III. Evacuation Support System

In this section, details our proposed evacuation support system, including its components.

A. System Overview

We conducted research on a system that assists in evacuation support by controlling a group of UAVs with agents to navigate autonomously. In the areas evacuation support is provided, we create a graph connecting points where people are present as nodes and connecting passable routes as edges. This graph represents the range of human activity. Each point is assigned a default priority for support. The derivation formula for priority is defined as equation (1).

$$priority = \alpha d + \beta t + \gamma U + \delta D \qquad (1)$$

Here, α, β, γ, δ are empirically determined coefficients, d is the distance to the evacuation center, t is the average time to reach the destination from that point, U indicates the percentage of evacuees around that point, and D indicates the level of secondary disaster risk in and around that point. Thus, the impact of disasters on priority is considered.

In this study, wireless communication is assumed to be unusable during disasters. Multiple evacuation centers and points are provided in the area of operations, and an opportunistic communications network is established using UAVs. UAVs navigate between points, search for evacuees, and guide them to an evacuation center. Servers are installed at the evacuation centers to receive disaster information from UAVs visiting the centers and to calculate the optimal route.

The priority is updated each time a UAV reports on the disaster situation. UAVs have two roles: collecting information on disaster situations and guiding evacuees. Therefore, this system uses two types of UAVs for information gathering and evacuation guidance. UAVs for information gathering are referred to as UAV1 and those for evacuation guidance as UAV2.

B. UAV1

UAV1 is an information-gathering UAV that acquires information on disaster evacuees by flying in a width-priority exploratory flight over the graph, starting from the evacuation center. The data are conveyed through an opportunistic network and accumulated on a server located in the evacuation center. The server calculates an efficient evacuation route from the collected information on the evacuees. At that time, the priority of each point is updated, and the search is started again from the point with the highest priority. Therefore, there is a risk that once the priority has been lowered, the point may not be observed. However, UAV2 passes the information on evacuees found during evacuation guidance to UAV1, which can increase the priority.

C. UAV2

UAV2 is used for evacuee guidance. It receives the evacuation route generated by the server and guides the evacuees accordingly. It issues instructions for the evacuees to follow and provides guidance to the evacuation center. Simultaneously, if evacuees emitting signals from smartphones are detected, UAV2 transfers this information to UAV1, if it is in communication range. This helps prevent the disadvantage of points whose priority has decreased through not being observed again.

D. System Flow

UAV1 and UAV2 must coordinate their behaviors. The coordination is achieved through the information shared by them through the control server. Initially, UAV1 does not have information on the number of evacuees or the disaster situation. It first observes the overall situation by traversing all points in a breadth-first manner. The observed data are shared with the server installed at the evacuation center upon

 ISBN: 978-1-68558-183-1

UAV1's return. The server receives and stores information from the visited UAVs. Then, it generates and updates the priority score on the basis of this information, which is used to determine UAV1's subsequent movement routes. UAV1 focuses on navigating between points with high priority. During this process, if a UAV encounters another UAV, they exchange the information about the number of evacuees and the disaster situation.

Once the patrol of the points is complete, UAV1 returns to the departure evacuation center. It reports the number of evacuees and the disaster situation to the server at the evacuation center. The server updates the priority score and determines evacuation routes for each point based on the priority. The determined evacuation routes are then passed to UAV2 one by one.

When a UAV2 discovers evacuees, it sends a message to the evacuees to follow it. This message is transmitted to the evacuees' devices and emitted through flashing lights. UAV2 then guides them to the evacuation center.

If UAV2 confirms that a previously passable evacuation route observed by UAV1 is disrupted during guidance, making it impossible for evacuees to evacuate, it halts guidance, returns to the departure point, and informs the server accordingly. The server recalculates the new evacuation routes and relaunches UAV2. Additionally, UAV2 reports the number of evacuees, the disaster situation around the points, and its own status to other UAVs whenever they pass by. The server compiles this information. Consequently, the value of the priority scores necessary for UAV1's path generation changes in real time. This process repeats until all evacuees have completed evacuation.

IV. Algorithm for Determining Evacuation Routes

This system maps UAVs and human behavioral areas as a graph. Therefore, it can utilize evacuation route determination algorithms based on conventional shortest path algorithms. We took advantages of the algorithm of Patel et al [8].They invented an algorithm that simultaneously determines multiple routes in addition to the shortest path between the current location and destination (Figure 2).

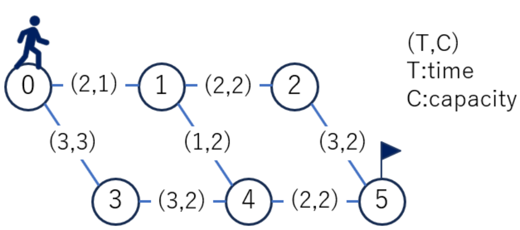

Figure 2. Graph example.

Consider the scenario in which an evacuee at node 0 evacuates to node 5. In this case, we select the shortest path from 0 to 5, which is 0-1-4-5. Then, the capacities of edges 0-1, 1-4, and 4-5 decrease by 1. The capacity becomes 0 for edge 0-1, while edges 1-4 and 4-5 become 1. Since the capacity of edge 0-1 becomes 0, it becomes impassable. Therefore, in the next step, we select an alternative shortest path that does not include edge 0-1, such as 0-3-4-5. This illustrates how multiple routes are selected simultaneously.

V. Implementation on a Simulator

In this study, to verify the evacuation support system, we developed a GUI simulator using Unity. Figure 3 and Figure 4 show the simulator in action. Unity is a game engine provided by Unity Technologies and is known for its versatility across various platforms. The development environment used was Unity 2020.3.27f1.

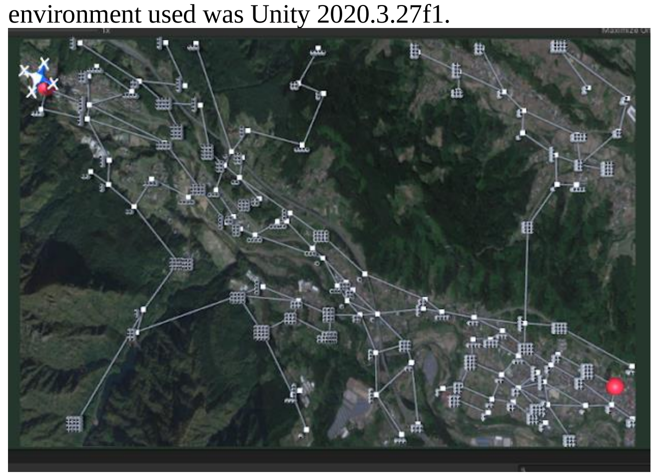

Figure 3. The GUI simulator.

The paths connecting points must allow UAVs to navigate and guide people that necessitate passable roads. In cases where neighboring points are connected by roads, linear movement is assumed between those points. Additionally, each point is assigned one of three levels of hazard: "Safe Point," "Disaster Alert Point," and "Impassable Point." The initial hazard level values were assigned as 0.25, 0.5, and 0.75, respectively. These hazard levels can be dynamically changed as the monitoring UAVs perceives the disaster situation.

We created a map based on actual road networks. Referring to maps provided by the Geospatial Information Authority of Japan, we set up road networks, as shown in Figure 3, focusing on roads designated for evacuation guidance.

When the simulation starts, the evacuees and UAVs are placed at each point. UAVs initiate evacuation support from the base, which is the evacuation shelter, where they are initially placed. Evacuees are assumed to move to the nearest intersection, and the simulation starts with evacuees placed at each intersection.

Evacuees are initially displayed as translucent objects before being discovered by UAVs. Once discovered by UAVs, they become opaque and visible. This allows for visual tracking of evacuees already discovered by UAVs during the simulation as shown in Figure 4. Additionally, observing the distribution of discovered evacuees provides insight into the progress of information gathering for evacuation support. Figure 5 shows the screen when the UAV for evacuation guidance (blue UAV) has completed eight attempts for guidance. The paths already traversed by the evacuation guidance UAV are colored green.

 ISBN: 978-1-68558-183-1

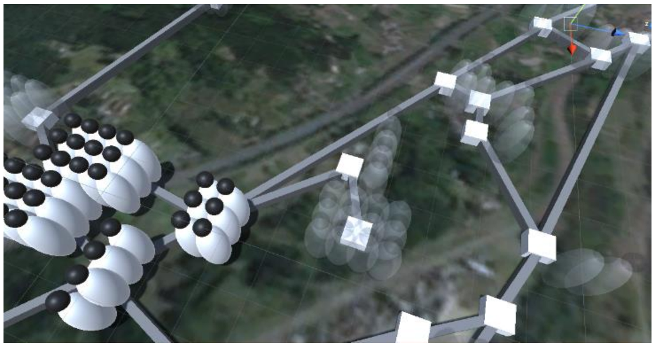

Figure 4. Representation of undiscovered evacuees as translucent.

Figure 5. UAV paths during evacuation guidance.

At this point, approximately half of the evacuees have been discovered, with their distribution observed to be biased toward the left side of the map. This bias is due to the monitoring UAVs initially follow a predetermined route to observe all points. Figure 5 also shows that UAVs guide evacuees in a situation where the distribution of evacuees remains unknown; because, at this time, not all the points have been observed yet, and UAVs are flying the predetermined routes.

Focusing on the flight paths of the displayed UAV2, we can observe that it has navigated points around the area relatively close to the evacuation shelter. Since guidance starts without information on the disaster situation, it is expected to begin from points relatively close to the shelter.

VI. CONCLUSION AND FUTURE WORK

We have developed a system that supports evacuation after disasters using UAVs. We focus on addressing disasters in mountainous areas. Since our assumption is a situation where the information infrastructure is paralyzed, we set two types of UAVs for this system. We created a simulator that displays the movement history of UAVs and people on a map so that we can observe their actions and discover problems. The map is based on Annaka City for a real-world scenario. As a result, we could observe the UAVs' actions and evacuees' behaviors as we predicted. We confirmed that this simulator can visually demonstrate UAV actions even to non-developer disaster experts.

As we proceed to build a better simulator, we must implement human behaviors more realistic than those of the current simulator. In actual disaster situations, it is presumed that evacuees autonomously undertake evacuation actions even when they are not guided by UAVs. For example, by applying ant colony optimization algorithms, it may be possible to mimic the actions of people avoiding dangerous areas and heading toward their destination. Simulating actions during periods when they are not guided by UAVs allows the verification of the impact of the evacuation support system on evacuation actions. In addition, we can focus on the people who failed to evacuate. Then, we must analyze their behavioral history to analyze whether they received adequate evacuation assistance. We believe our attempt to build this evacuation support system to create a new method to save people living in mountainous areas that are previously rather neglected in Japan.

ACKNOWLEDGMENT

This work was partially supported by JSPS KAKENHI Grant Number JP20K05010.

REFERENCES

[1]. K. Katayama, H. Takahashi, S. Yokoyama, K. Gafvert, and T. Kinoshita, "Evacuation Guidance Support Using Cooperative Agent-based IoT Devices, " Proc. of the 6th IEEE Global Conference on Consumer Electronics, 2017.

[2]. K. Katayama, H. Takahashi, N. Yokota, K. Sugiyasu, G. Kitagata, and T. Kinoshita, "An Effective Multi-UAVs-based Evacuation Guidance Support for Disaster Risk Reduction," Proc. of the 6th IEEE International Conference on Big Data and Smart Computing, Feb 2019.

[3]. H. Takahashi, K. Katayama, N. Yokota, and K. Sugiyasu, "Evacuation Guidance Supporting System Using UAV for Coastal Area," Proc. of the 3rd IEEE Global Conference on Life Science and Technologies, 2021.

[4]. S. Taga, T. Matsuzawa, M. Takimoto, and Y. Kambayashi; "Multi-Agent Base Evacuation Support System Using MANET," Vietnam Journal of Computer Science, vol.6, no.2, pp.177–191, 2019.

[5]. S. Taga, T. Matsuzawa, M. Akimoto and Y. Kambayashi; "Multi-Agent Base Evacuation Support System Considering Altitude, Proceeding of the Eleventh International Conference on Agents and Artificial Intelligence," vol. 1, pp.299-306, Feb 2019.

[6]. A. Fujihara and H. Miwa, "Real-time Disaster Evacuation Guidance Using Opportunistic Communications," Proc. of IEEE/IPSJ 12th International Symposium on Applications and the Internet, 2012.

[7]. K. Uno and K. Kashiyama, "Development of Simulation System for the Disaster Evacuation Based on Multi-Agent Model Using GIS," Tsinghua Science and Technology, 13(S1), pp. 348-353, 2008.

[8]. N. Patel, M. Min, and S. Lim, "Accurate evacuation route planning using forward-backward shortest paths," Proc. of Annual IEEE Systems Conference, 2016.

[9]. L. Rothkrantz and M. Popa, "Routing Pedestrian in Smart City Networks," Proc. of Smart City Symposium, 2018.

Deciphering Brand Identity from package: Visual Feature Analysis through Convolutional Neural Networks

Asaya Shimojo and Shoichi Uratani
KONICA MINOLTA, Inc.
Tokyo, Japan
email: asaya.shimojo@konicaminolta.com, shiyouichi.uratani@konicaminolta.com

***Abstract*— Many brands traditionally rely on qualitative methods to design their product packaging, leaving uncertainties about the consistency of brand identity across different packages. This study leverages machine learning to quantitatively extract and analyze design elements that resonate with consumers' perception of brand identity. Specifically, we employed Grad-CAM, an interpretative method for Convolutional Neural Networks (CNNs), to identify crucial visual features—termed Visual Identities—within the middle layers of a model trained on specific brand package images. These features were analyzed to determine their influence on package classification and their alignment with human perception of brand identity. Our findings demonstrate that the machine learning approach approximates human perception closely, providing a novel quantitative method to enhance and maintain brand identity. Additionally, we quantified the contribution of each identified Visual Identity to overall brand recognition, offering a more systematic approach to understanding and preserving a brand's distinctiveness that has traditionally been handled qualitatively.**

Keywords- Grad-CAM; Brand Identity; Visual Identity; Package Design; Consumer Recognition.

I. INTRODUCTION

A product's packaging design reflects its Brand Identity (BI), significantly impacting consumer perception. Previous research has explored which elements of package design contribute to BI. However, this evaluation process can burden evaluators and vary individually, focusing on limited and abstract aspects like color tones and fonts, reducing its practicality. The purpose of this study is to utilize the latest machine learning technologies to automatically extract design elements from packaging and quantitatively analyze their impact on consumer brand recognition. This approach assists brand managers in developing more effective packaging design strategies.

A. The Aim of This Study

The study visualized intermediate model layers to extract packaging design elements. These elements are then compared with factors that humans consider indicative of BI, to validate the relevance of the former. Through this process, we develop a simple yet precise method that compares human-detected BI features with those identified by the model.

B. Related Works

BI is the embodiment of a company's values and characteristics [1]. Specifically, BI is composed of visual elements such as logos, colors, and design styles [2]. Thus, design elements that visualize the values and concepts of a Brand and symbolize the brand are called Visual Identity (VI). For example, package design influences consumers' emotions and perceptions through its VI, including its colors, shapes, materials, logos, and text [2]. It has also been found that the visual attractiveness of a package draws consumers' attention and forms a favorable impression of the product, as well as a high perception of the product's quality and value [3]. Therefore, the VI of the packages is considered to play a central role in BI communication [4]. Thus, most brands need to express their identity through their VI and keep consumers consistently identifying with the brand. In fact, VI consistency has been found to improve purchase intent and brand loyalty [5].

On the other hand, the specific visual elements used to represent the brand need to change with time and trends. However, this idea is inconsistent with maintaining consistency in the VI. In contrast, when interviews were conducted on the consistency of the BI with successive changes in the VI, art directors and other professionals had a narrow range of acceptance of consistency, and some consumers with a high aesthetic sense were also sensitive to changes in the VI [6]. However, while art directors emphasized the complexity of the VI construct, it was also clear that VIs are changed based on preference and emotion, making it a very inadequate method for capturing VIs in an exhaustive and quantitative manner.

In addition, quantitative studies have examined BI by focusing on logos [7][8]. However, VI requires an exhaustive study because other factors such as object edges and color contrasts are also considered to be involved in its composition [9].

C. Theoretical Background

In recent years, AI technology has advanced to the point where it is now possible to build CNN models that learn

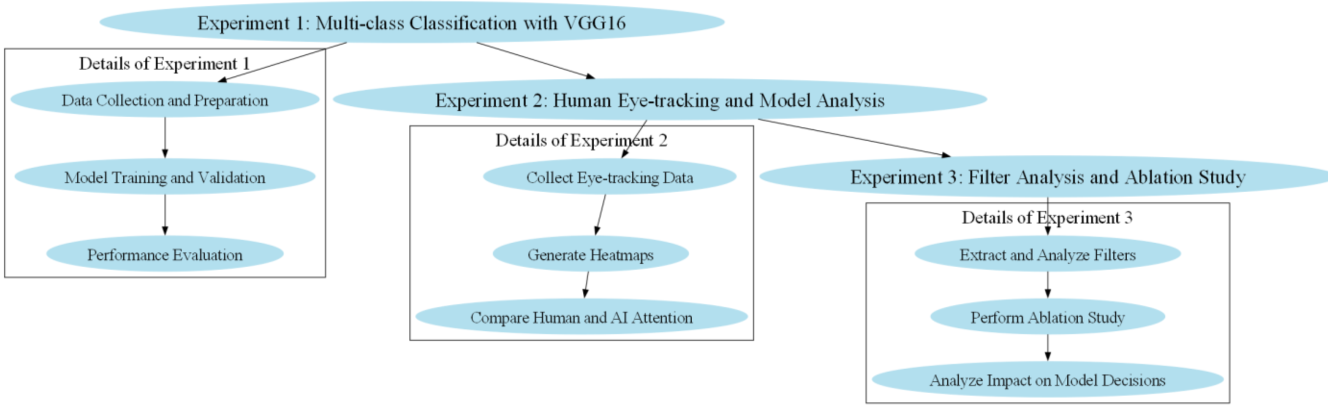

Figure 1. Flowchart of three experiments' procedures in this study.

package images of specific brands and classify whether the brand is that Brand. In fact, brand identification by CNN has been verified on brand logos and fashion show runway photos [10][11].

Moreover, a technique called Grad-CAM has also been established to visualize where in the image the model focused its attention in making decisions in classification [12]. Specifically, the gradient information for the feature map of the final convolutional layer associated with a particular output class can be used to present important regions in the input image as a heat map. The visualization of the heatmap facilitates the interpretation of the model, as this output indicates the image regions that the CNN focuses on when making a particular decision [13].

However, this method only visualizes the decision-making process of the model and does not directly reveal where humans, as consumers, focus their attention on the package to identify the brand. In fact, the comparison of the AI model's point of attention with the human visual area of attention is considered important but has not yet been validated [14]. Therefore, this study will examine the extent to which the visual factors that humans and machines focus on when identifying the packaging design of a particular brand coincide by collecting human eye-tracking data and comparing this with the results of a deep learning model using Grad-CAM. This examination would reveal the usefulness of the features automatically extracted by the model.

In summary, this study has two major contributions. One is to propose a methodology for automatically extracting the features used to determine whether a model is the relevant brand or not by training the model on the package data of a specific brand and visualizing the middle layer of the model using the above-mentioned techniques. This would enable quantitative VIs management for package design and more efficient communication of BI. The second is to verify the degree to which the trajectory of the human gaze matches the visualization of the middle layer of the learned model. In addition to validating the accuracy of the methodology, this could reveal the degree of agreement between the model's point of interest and the human visual area of interest. To realize these two contributions, three experiments were conducted in this study; their appearance was shown in Figure 1.

Specifically, in Section II, a multi-class classification model and Grad-CAM was developed to identify key design elements in package designs. In Section III, human eye-tracking data was compared with Grad-CAM visualizations to evaluate the consistency between human and machine recognition. In Section IV, the CNN model's processing of visual information was analyzed to identify design elements contributing to Brand A's BI. Finally, in Section V, we discussed the study's findings and limitations.

II. EXPERIMENT 1

This experiment built a multi-class classification model by fine-tuning VGG16 with package design data from brands A through E. Subsequently, the intermediate layers of the model were visualized using Grad-CAM to identify which design elements were instrumental in distinguishing between the package designs of the five brands.

A. Brand Selection and Dataset Construction

This experiment used a self-created dataset comprising approximately 6,000 images, specifically focusing on the packaging of brands with a long history and established BI. Based on scale [15], five writing instrument brands were selected, categorized into two luxury brands (Brands A and B), two general consumer brands (Brands C and D), and one lesser-known brand (Brand E). Figure 2 shows a thumbnail of Brand A's packaging design images of the dataset.

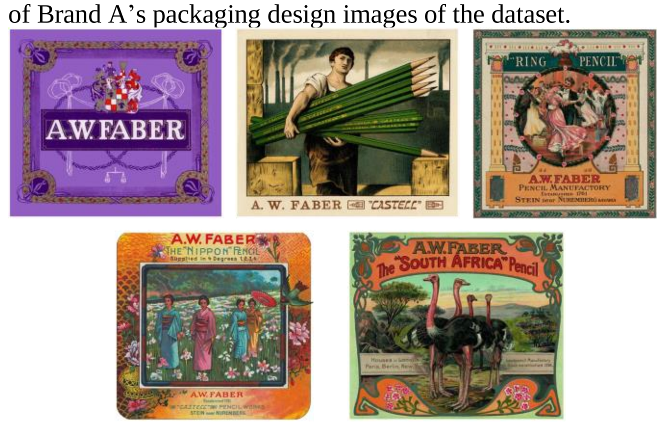

Figure 2. A thumbnail of some packaging design images of the dataset.

 ISBN: 978-1-68558-183-1

This selection was aimed at testing the generalizability across a broad consumer base. By conducting a multi-class classification that encompasses a variety of brands, the model could have the validity of this approach for tail brand with small sample sizes (e.g., Brand E). In the data collection process, considerations were made for copyright issues, acknowledging that the data has been publicly available for over 70 years. If the brand name was written on the package, it was masked in gray to hide it.

B. *Construction of Classification Model based on VGG16*

This study used VGG16, pre-trained on ImageNet, to develop a multi-class classifier capable of identifying package designs from Brands A through E. The model was trained with 70% of the dataset images, adjusting the output size of the final layer to 5 to classify each image into one of five brands. The remaining 30% of the dataset was used for validation. The training and validation processes were conducted 2,000 times, shuffling the data each time, and using a batch size of 32, with the number of epochs set to 10.

Performance was evaluated by varying the neuron counts in the model's dense layer from 51 (10% of full capacity) to 512 (100%). The model showed optimal performance with 512 neurons, achieving a validation accuracy of 91.18%. Models with 384 neurons also performed well, achieving an accuracy of 87.47%. However, models with fewer neurons—256, 128, and 51—all showed validation accuracies below 80%. Moreover, when the training dataset was reduced, the model with 512 neurons showed varying accuracies: 62% with 600 images (10% of full data), 77% with 1,500 images (25%), 82% with 3,000 images (50%), and 89% with 4,500 images (75%). A training dataset of at least 4,500 images was considered necessary for optimal performance. These performance changes are shown in Figure 3.

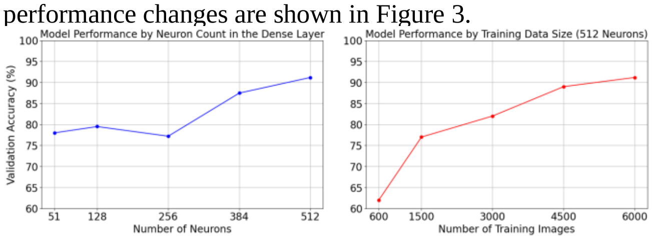

Figure 3. An Impact of neuron count or training data size on model accuracy.

This study used the largest model size and number of samples available (100% for both) because accuracy was given priority.

C. *Filter Visualization with Grad-CAM*

In the model, image features were transformed into a feature vector of 4,096 dimensions through the convolution and all the coupling layers. This made it difficult to directly understand which elements of the package design affected the classification results. Therefore, Grad-CAM was used to visualize the filter output of "block5_conv3," the layer closest to the output of the convolution layer (Conv2D), to determine which image regions had the most influence on the classification decision. As a result of evaluating the contribution of each layer to brand identification in our ablation study, it was found that the validation accuracy decreased by over 70% when this layer was disabled. The result indicated the layer's critical role in accurate brand recognition. Consequently, we determined that it was important to visualize this layer to identify VIs that affect brand recognition. For instance, the following package image showed a visualization of the areas in the image that contributed to being classified as Brand A (Figure 4).

Figure 4. Brand A classification contribution area acquired by Grad-CAM. "Predicted: 0" meant that the image was determined to be Brand A.

This gave the model's predictive basis in a form that was intuitively understandable to humans. In the next section, we clarified the extent to which the results of this visualization correspond to the areas that humans pay attention to when recognizing brands.

III. EXPERIMENT 2

This experiment investigated similarities that exist between human visual perception of package designs and the results of image analysis by the machine learning model obtained in the previous section. Specifically, human eye-tracking data were collected and compared with the features of the model visualized by Grad-CAM.

A. *Method*

Six adults (2 females, 4 males; Age: 39.8 ± 7.9 years) participated in the experiment. The participants had normal visual acuity. They were presented with 10 packages each of five brands and learned their VIs. The participants' heads were then fixed by a chinrest placed approximately 60 *cm* from the monitor screen. Images were displayed across the entire monitor screen. Then, one image from the image set was presented at random, and the participant gazed at it for 5 seconds. There were 20 images totally in the image set, which consisted of 10 packages for each brand except the package used for the learning.

The participants' eye movements were recorded while gazing at the package images. To record the trajectory, a webcam (ELECOM UCAM-C750FBBK; resolution 1920 × 1080 *px*, frame rate 30 *FPS*, angle of view 66 degrees, 1/4-*inch* CMOS sensor), a monitor (I-O DATA KH240V-B; 23.8-*inch* wide, resolution 1920 × 1080 *px* resolution, ADS panel, brightness 250 cd/m^2, response time 5 *ms*), and *GazeRecorder* [16], a line-of-sight measurement software. The experiment room brightness was kept constant at 500 *lux*.

After gazing at each image, participants responded with a confidence level ranging from 0 % (definitely not a Brand A package) to 100 % (definitely a Brand A package) that they thought each image was a Brand A package.

B. Results and Discussion

Comparison both heat maps (one from human eye-tracking and one from Grad-CAM) quantitatively assessed eye-tracking data with an average confidence level of at least 70 % were used to perform the analysis only on package images that the participants were confident were Brand A.

This analysis used the Jaccard Index to investigate the similarity between two different image generation processes: a heatmap based on human visual tracking and a computerized Grad-CAM heatmap. The Jaccard index is a measure of similarity between sets, with values varying from 0 to 1. Here, 1 indicates that the two heatmaps match perfectly, while 0 means no overlap at all.

Specifically, heatmaps based on human visual tracking were generated from viewpoint data as participants viewed each package. The visual tracking data was processed using the built-in systems of *GazeRecorder*. In contrast, the Grad-CAM heatmaps were obtained by analyzing the same images using the Python *cv2* library within a specified CNN model. Each heatmap was visualized as a color intensity map indicating areas of visual attention. In these heatmaps, red indicated the most focused areas of attention, and blue indicated the least focused areas.

These heat maps were then overlaid at the pixel level to quantitatively assess the size and distribution of commonly noticed regions of interest. The process involved loading the images in grayscale, resizing, and binarizing them using a Python script. The intersection (common area) and union (total area) of these images were then computed to derive the Jaccard index. Figure 5 was shown as an example of the comparison.

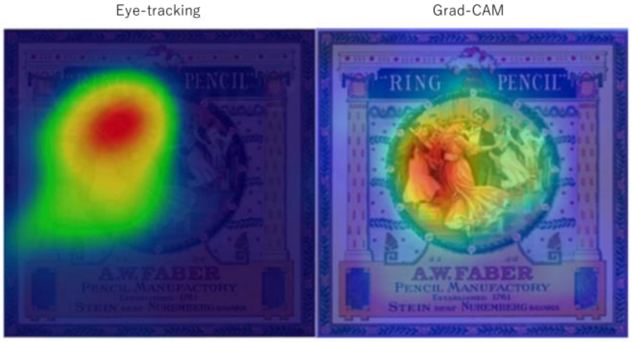

Figure 5. An example of a comparison of the features of the middle layer of the visualized CNN and the gaze trajectory.

The analysis resulted in a mean value of 0.32 (*SD*=0.12) for the obtained Jaccard index. This value indicated that although a certain degree of similarity was observed between the heat maps, differences existed in several regions. However, it was also visually readable that in most package pairs, the areas of highest attention matched.

Thus, the results suggest that there is a partial match between the AI-generated attention maps and the areas of human attention focus, supporting the possibility that the the extent to which human visual attention and machine learning models' judgments are consistent. The analysis was conducted to identify similarities and differences between the human recognition process and the machine learning model's ability to recognize patterns. Note that only human

AI's visual processing algorithms may be somewhat like the human visual recognition process.

IV. EXPERIMENT 3

This experiment demonstrated that the CNN model processes visual information in a manner somewhat analogous to human perception. However, delineating the specific features the model recognizes and uses for classification remains challenging. Therefore, we focused on identifying individual design elements and assessed their contribution to the BI of Brand A.

A. Extraction of Brand classification contribution filters

This experiment aimed to understand how the model extracts and interprets image features by visualizing filter activation in the middle layer of the CNN model fine-tuned in Experiment 1. Specifically, filter weights were extracted from the intermediate layer, "block5_conv3," and, activation maps for each filter were generated. The Python *TensorFlow* and *Keras* libraries were used to generate images starting with random noise for each filter and iteratively update the images in the direction of maximizing filter activation.

This analysis resulted in 512 output filters, of which 40 were significantly effective in classifying Brand A. Figure 6 showed one such filter—specifically, the 110th filter—which was particularly interpretable. Upon visualizing the activation map using the jet colormap, areas with the highest activations corresponded remarkably to objects, such as pencils, fountain pens, and geometric shapes. This suggests that the model might prioritize sharp and defined tips of objects, which are characteristic elements in the VI of Brand A. Indeed, this distinct emphasis on pointed features was visually confirmed to be more pronounced in Brand A's packaging compared to other brands' ones, aligning with the brand's distinctive aesthetic attributes. Other contributing filters also were described below.

This process allowed us to understand what is involved in the identification of Brand A by extracting the filter used by the CNN for discrimination and overlaying it on the original image in the form of a heat map to see where it is applied. In other words, the physical characteristics of Brand A, or VI, since that is where it is used to recognize Brand A. This process was useful in understanding how the model identified a specific VI and how it contributes to the identification of the brand.

B. Ablation Study for the Contribution Ratio for each VI

Based on the results of the filter visualization, an ablation study was conducted to better understand its functional importance. In this study, the weights of each

 ISBN: 978-1-68558-183-1

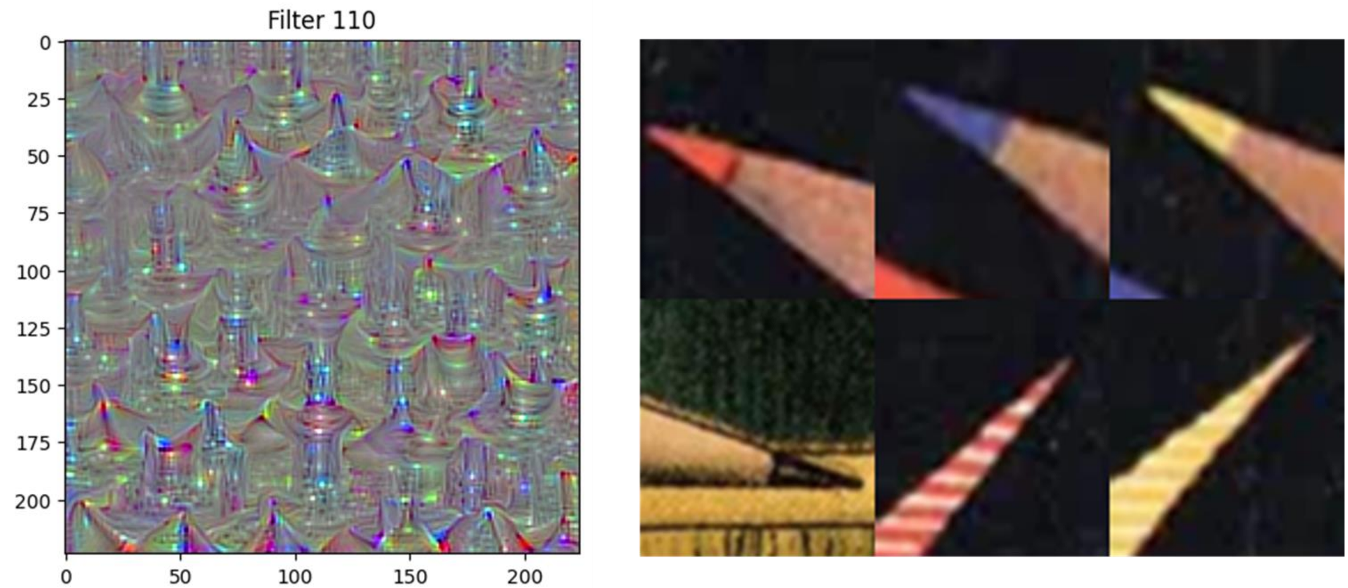

Figure 4. Activation Map of the 110th Filter Highlighting Response to Sharp Object Tips.

filter in the middle tier were individually set to zero, and the impact of these changes on the model's overall testing accuracy was systematically evaluated. By disabling each filter, we quantitatively analyzed the extent to which the filter contributed to the model's ability to make decisions.

The analysis showed that the filters' contribution ranged from a maximum of 25% to a minimum of 0.12%. Of note was the 110th filter, which, when disabled, resulted in a 13% reduction in the ability to identify Brand A. This indicates that the sixth filter plays a particularly important role in extracting the visual features of Brand A. Other contributing points to Brand A recognition detected were the strength of the curve of the product (the 93rd filter; 11.2% contribution to Brand A recognition), the light reflectance of the metal body (the 103rd filter; 8.9%), and the strength of the color contrast between the background and the product (the 106th filter; 8.1%).

Findings such as these are valuable in clarifying the key visual elements in brand identification and understanding how they affect the performance of the model. Therefore, the process of extracting VIs from the package and calculating their impact on BI could be automated.

V. CONCLUSION AND FUTURE WORK

This study used machine learning to quantify the impact of packaging design on BI. Specifically, CNN and Grad-CAM were used to explore the extent to which machine feature extraction is consistent with human BI recognition. The results show that machine learning models can effectively identify and highlight important design elements in brand recognition.

It was confirmed that the CNN model can identify brands based on specific elements of the package design, such as hue and logo, but also on detailed representations, such as the edges in an illustration. This showed that machines can capture important elements of VI and use them to make classification decisions.

Furthermore, visualization with Grad-CAM reveals that the areas that the model focuses on coincide with the areas that humans focus on when recognizing BI. This suggested that machine learning models may be able to mimic the human recognition process, indicating the existence of common ground between human and machine recognition.

The study also provided a method for quantifying the impact of individual design elements on brand recognition. This would enable brand managers to understand the specific impact of each packaging design element on BI and make more strategic design decisions.

A. Limitation and Future works

While Grad-CAM effectively highlights crucial areas within an image, it could focus on regions with high visual saliency, potentially overlooking subtler yet important features that contribute to the overall understanding of the image [12]. This phenomenon, known as the saliency bias, would raise concerns about the comprehensiveness of visual explanations provided by convolutional networks.

To address this limitation, integrating attention mechanisms that adjust focus based on the context of the entire image rather than just visual salient features has gained traction. For instance, the Transformer relies entirely on an attention mechanism, discarding the need for recurrent layers [17]. This model dynamically weights the influence of different parts of the input data, which can be particularly beneficial for understanding complex images in a more human-like manner. Furthermore, multi-dimensional scaling techniques can complement attention mechanisms by reducing high-dimensional data into a space where relationships between features are preserved, allowing for a clearer visualization of how features interact and contribute to the model's decisions [18].

These methodologies could avoid the saliency bias, which helps identify subtle yet crucial patterns that might be missed by traditional saliency-based approaches. They not only perform well but also align more closely with human cognitive processes, potentially making machine learning tools more intuitive and trustworthy for users in real-world applications.

Moreover, Experiment 2 showed that feature extraction using Grad-CAM is consistent with human visual regions of interest. This suggests that the proposed model captures visual elements like the human brand recognition process. However, due to limitations in the number of participants, further validation is required before these results can be widely generalized. It is essential that future research extensively test the generalizability and effectiveness of the proposed model through a variety of brand categories and a large set of experiments.

In addition, increasing the diversity and comprehensiveness of the data set would allow us to capture a broader range of visual elements of BI. This would include data from brands from different time periods and cultural backgrounds, which would enhance the generalizability of the model and provide a more generic quantitative method for VI management.

B. *Social Contribution*

This study would contribute to strategic improvements in package design in that the visibility elements of the BI could be automatically extracted. Specifically, it would help package design to maintain consistency in BI and differentiate it from other brands. As a result, brands would be able to manage their VI in package design more strategically, leading to stronger relationships with consumers and increased brand value.

REFERENCES

[1] Jean-Noël. Kapferer, "*Strategic Brand Management*." London:Kogan Page. 1992.

[2] D. A. Aaker, "*Building Strong Brand*." New York: The Free Press—A division of Simon & Schuster Inc. 1996.

[3] M. E. H. Creusen, and J. P. L. Schoormans, "The different roles of product appearance in consumer choice." *Journal of Product Innovation Management*, vol. 22, no. 1, pp. 63–81, 2005.

[4] R. L. Underwood, and N. M. Klein, "Packaging as brand communication: Effects of product pictures on consumer responses to the package and brand." *Journal of Marketing Theory and Practice*, vol. 10, no. 4, pp. 58–68, 2002.

[5] M. Lindstrom, "Broad sensory branding." *Journal of Product & Brand Management*, vol. 14, no. 2, pp. 84–87, 2005.

[6] B. J. Phillips, E. F. McQuarrie, and W. G. Griffin, "How visual brand identity shapes consumer response." *Psychology & Marketing*, vol. 31, no. 3, pp. 225–236, 2014.

[7] R. Beise-Zee, "Brand equity retention after rebranding: a resource-based perspective." *Journal of Brand Management*, vol. 29, no. 2, pp. 208–224, 2022.

[8] A. Jakšić-Stojanović, and N. Šerić, "Brand Identity of Montenegro through Verbal and Visual Elements of its Logo." *Journal of Marketing Development and Competitiveness*, vol. 12, no. 4, pp. 134–144, 2018.

[9] V. U. Vinitha, D. S. Kumar, and K. Purani, "Biomorphic visual identity of a brand and its effects: a holistic perspective." *Journal of Brand Management*, vol. 28, no. 3, 272–290, 2021.

[10] N. T. Van Trang, T. L. Nghiem, and T. M. Do, "Improving the competitiveness for enterprises in brand recognition based on machine learning approach." *Global Changes and Sustainable Development in Asian Emerging Market Economies*, vol. 1, pp. 359–373, 2022.

[11] E. Martarello, "*Exploring CNNs and Attention Mechanisms for Brand Identification in Fashion Runway Shows*." Master's thesis, Ca' Foscari University of Venice, 2023.

[12] R. R. Selvaraju, M. Cogswell, A. Das, Vedantam, R., Parikh, D., and Batra, D. "Grad-CAM: Visual explanations from deep networks via gradient-based localization." *International Journal of Computer Vision*, vol. 128, pp. 336–359, 2017.

[13] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2921–2929, 2016.

[14] J. An, and I. Joe, "Attention map-guided visual explanations for deep neural networks." *Applied Sciences*, vol. 12, no. 8, 3846, 2022.

[15] Jean-Noël. Kapferer, "Why are we seduced by luxury brands?." *Journal of Brand Management*, vol. 6, no. 1, pp. 44–49, 1998.

[16] GazeRecorder. Online Eye Tracking Software. [Online]. Available from: https://gazerecorder.com/ 2024.05.06

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6000–6010, 2017.

[18] I. Borg and P. J. F. Groenen, "*Modern Multidimensional Scaling: Theory and Applications*," Springer Series in Statistics, Springer-Verlag New York, 2005.