

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Setelah dilakukan penelitian dengan empat model yang berbeda dari model vanilla Pix2code dapat disimpulkan bahwa penggunaan GRU tidak dapat mempercepat waktu pelatihan model. Dengan penggunaan GRU sebagai *language model* dan *decoder* menghasilkan waktu pelatihan yang lebih lama, dimana pada arsitektur Pix2code yang menggunakan LSTM memerlukan waktu pelatihan 1.9352 jam sedangkan vanilla GRU memerlukan waktu pelatihan 1.9872 jam. Hal tersebut dapat dimungkinkan karena penggabungan ekstraksi fitur dari input gambar dan konteks yang membuat data *sequence* menjadi cukup kompleks sehingga GRU memerlukan waktu pelatihan yang lebih lama.

Penggunaan pre-trained model baik Xception, EfficientNetV2B0, maupun EfficientNetV2M sebagai fitur ekstraktor gambar memberikan pengurangan waktu yang cukup signifikan. Waktu pelatihan yang dibutuhkan untuk Xception-GRU adalah 1.0386 jam, EfficientNetV2B0-GRU 1.0061 jam, dan EfficientNetV2M-GRU 1.1652 jam. Penurunan waktu pada penggunaan pre-trained model untuk ekstraktor fitur dikarenakan jumlah trainable parameter yang lebih sedikit dibanding jumlah trainable parameter saat menggunakan scratch CNN.

Kemudian pengujian dengan kompleksitas input yang berbeda dilakukan untuk mengetahui kemampuan model dalam memprediksi DSL kode dengan kompleksitas input yang berbeda. Vanilla Pix2code mendapatkan skor kompleksitas rendah 0.4730, sedang 0.5941, dan tinggi 0.6098. Vanilla GRU mendapatkan skor kompleksitas rendah 0.5310, sedang 0.4340, dan tinggi 0.4279. Xception GRU mendapatkan skor kompleksitas rendah 0.8071, sedang 0.6340, dan tinggi 0.68344. EfficientNetV2B0 GRU mendapatkan skor kompleksitas rendah 0.2964, sedang 0.5665, dan tinggi 0.5852. EfficientNetV2M mendapatkan skor kompleksitas rendah 0.2651, sedang 0.3892, dan tinggi 0.6442. Dari skor BLEU

tersebut dapat dilihat bahwa Xception GRU menjadi model paling *robust* dalam pengujian dengan kompleksitas data yang berbeda.

Pengujian secara kualitatif juga dilakukan dengan cara membandingkan secara visual gambar masukan dengan hasil tangkapan layar setelah kode dijalankan. Dari pengamatan tersebut didapatkan kesimpulan bahwa keseluruhan model tidak terlalu baik dalam mengenali warna elemen pada gambar. Hal tersebut dimungkinkan karena kelemahan pada visual model. Ekstraksi fitur yang dilakukan baik oleh *scratch* CNN maupun *pre-trained* CNN tidak dapat menangkap secara keseluruhan fitur dari gambar.

5.2 Saran

Penggunaan GRU dan *pre-trained model* yang dilatih dengan dataset ImageNet terbukti dapat berpengaruh dalam waktu pelatihan dan performa. Meskipun begitu, model masih belum mempunyai performa yang cukup baik dalam prediksi dengan kompleksitas input yang berbeda. Adapun saran untuk penelitian berikutnya adalah sebagai berikut.

1. Mencoba mengimplementasikan arsitektur yang sama dengan *hyper parameters* yang berbeda.
2. Mencoba implementasi arsitektur RNN lain baik sebagai *language model* maupun *encoder*.
3. Mencoba penerapan *pre-trained* CNN model lain. Akan lebih bagus apabila model *pre-trained* yang digunakan dilatih berdasarkan dataset *user interface*.

U N I V E R S I T A S
M U L T I M E D I A
N U S A N T A R A

DAFTAR PUSTAKA

- [1] X. Pang, Y. Zhou, P. Li, W. Lin, W. Wu, and J. Z. Wang, 'A novel syntax-aware automatic graphics code generation with attention-based deep neural network', *Journal of Network and Computer Applications*, vol. 161, p. 102636, 2020.
- [2] S. Barnett, R. Vasa, and J. Grundy, 'Bootstrapping Mobile App Development,' in *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, 2015, vol. 2, pp. 657–660.
- [3] T. A. Nguyen and C. Csallner, 'Reverse Engineering Mobile Application User Interfaces with REMAUI (T),' *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, Lincoln, NE, USA, 2015, pp. 248-259, doi: 10.1109/ASE.2015.32.
- [4] T. Beltramelli, 'pix2code: Generating Code from a Graphical User Interface Screenshot', arXiv [cs.LG], 22-May-2017.
- [5] M. A. Al-Malla, M. A. Al-Malla, A. Jafar, and N. Ghneim, 'Pre-trained CNNs as feature-extraction modules for image captioning', *ELCVIA Electron. Lett. Comput. Vis. Image Anal.*, vol. 21, no. 1, pp. 1–16, May 2022.
- [6] S. Yang, X. Yu, and Y. Zhou, 'LSTM and GRU neural network performance comparison study: Taking yelp review dataset as an example', in *2020 International Workshop on Electronic Communication and Artificial Intelligence (IWECAI)*, Shanghai, China, 2020.
- [8] Keras. 'Keras Application' keras.io. <https://keras.io/api/applications/> (Accessed Feb. 17 2023).
- [9] M. Zaveri, 'What is boilerplate and why do we use it? Necessity of coding tyle guide.' freecodecamp.org. <https://www.freecodecamp.org/news/whats-boilerplate-and-why-do-we-use-it-let-s-check-out-the-coding-style-guide-ac2b6c814ee7/>. (Accessed Feb. 17, 2023)

- [10] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, and H. Laga, 'A Comprehensive Survey of Deep Learning for Image Captioning', *arXiv [cs.CV]*. 2018.
- [11] K. Wołk and K. Marasek, 'Enhanced Bilingual Evaluation Understudy', *arXiv [cs.CL]*. 2015.
- [12] Pix2code Dataset: pix2code, 2017. [Online]. Available: <https://github.com/tonybeltramelli/pix2code>
- [13] K. Simonyan and A. Zisserman, 'Very deep convolutional networks for large-scale image recognition', *arXiv [cs.CV]*, 04-Sep-2014.
- [14] S. Sakib, N. Ahmed, A. J. Kabir, and H. Ahmed, 'An overview of Convolutional Neural Network: Its architecture and applications', *Preprints*, 14-Feb-2019.
- [15] J. Kenshian. 'Implementasi on-device Offline Learning Menggunakan Algoritma Long-Short Term Memory untuk Next-word Prediction dalam Bahasa Indonesia', 2022.
- [16] M. Tan and Q. V. Le, 'EfficientNetV2: Smaller Models and Faster Training', *arXiv [cs.CV]*. 2021.
- [17] A. van Deursen, P. Klint, and J. Visser, 'Domain-specific languages', *SIGPLAN Not.*, vol. 35, no. 6, pp. 26–36, Jun. 2000.
- [18] A. van Deursen, P. Klint, and J. Visser, 'Domain-specific languages', *SIGPLAN Not.*, vol. 35, no. 6, pp. 26–36, Jun. 2000.
- [19] NuxtJs. (2023). Accessed: Jun. 04, 2023. [Online]. Available: <https://v2.nuxt.com/docs/get-started/>
- [20] Stackoverflow. '2022 Developer Sruvey'. Stackoverflow.co. Accessed: Jun. 03, 2023.[Online]. Available: <https://survey.stackoverflow.co/2022/>

[21] I. Sutskever, O. Vinyals, and Q. V. Le, ‘Sequence to Sequence Learning with Neural Networks’, *arXiv [cs.CL]*. 2014.

[22] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, ‘Show and Tell: A Neural Image Caption Generator’, *arXiv [cs.CV]*. 2015.

[23] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, ‘BLEU’, in *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, Philadelphia, Pennsylvania, 2001.

[24] T. Ghandi, H. Pourreza, and H. Mahyar, ‘Deep Learning Approaches on Image Captioning: A Review’, *arXiv [cs.CV]*. 2022.

