# Towards an AI playing *Touhou* from pixels: a dataset for real-time semantic segmentation

Dario Ostuni
*Department of Computer Science*
*University of Verona*
Verona, Italy
dario.ostuni@univr.it

Ettore Tancredi Galante
*Department of Computer Science*
*University of Milan*
Milan, Italy
ettoretancredi.galante@studenti.unimi.it

***Abstract*—When playing from pixels, AIs share some of the struggles that humans face when playing a game, namely not knowing its internal state. In this paper we begin the exploration of the AI-playing-from-pixels problem for *Touhou*, a bullet hell game. Albeit being a massively popular game in some niches, the community has yet to produce an AI capable of beating it, while looking only at pixels, in *Lunatic* mode, the hardest difficulty.**

**We propose, as a first step, to build a semantic segmentation model to create a bridge to the internal-state-looking AIs. To achieve this, we created a dataset to train models for this task. This dataset is procedurally generated using manually labeled assets from classic era *Touhou* games.**

**After selecting five state-of-the-art real-time semantic segmentation networks, we trained them using our dataset. The results indicate that the models produced have a high classification performance over the validation set. However all models, but one, are too slow to run in real-time at the game's target frame rate. On real game footage the models show promising results, but the dataset needs to be strengthen to account for noise sources in the real game.**

***Index Terms*—Semantic segmentation, AI playing from pixels, Touhou, Artificial intelligence, Procedural dataset generation**

## I. Introduction

The research on Artificial Intelligence is of vital importance for an ever expanding set of fields with immediate applications (e.g. autonomous driving and robotics). However, an important driving force for it has come from the field of Game AI, whose results are being applied to other fields [20].

One active topic of research is AIs playing games from pixels. The seminal paper in this regard is from *DeepMind*, in which their general AI managed to play *Atari* games from pixels [16]. Nonetheless, there is still interest in developing AIs playing from pixels for a single game. For instance, the *ViZDoom* competition [10] is about AIs playing *Doom* from pixels only. According to the competition report, although the competition is centered around a single game, AIs are still not able to compete at the same level as humans [25].

In this paper we aim to start the investigation for pixel-reading AIs for the *Touhou* game series. The *Touhou Project*, or simply *Touhou*, is a series of *bullet hell* (a sub-genre of *shoot 'em up*) games that gained massive popularity in Japan, the country from which its creator is from, and internationally [12]. As of 2021, there are 30 installments in the *Touhou* series, with 18 main titles and 12 spin-offs. The gameplay of the main titles consists of a vertical-scrolling setting where the main character must dodge a barrage of bullets from the enemies while trying to shoot them down. Each title offers four levels of difficulty: *Easy*, *Normal*, *Hard* and *Lunatic*.

Developing an AI for playing a *Touhou* game at the *Lunatic* difficulty by reading only the raw output pixels is a challenging task. One of the focuses of *Touhou* is to get very fast reaction times from players to dodge bullets. By running at 60 *FPS*[1], AIs playing it must make their decisions quickly. In fact, delaying them by even a few frames could be fatal for the main character, thus AIs should be able to keep up in real-time with the game. Moreover, playing *Touhou* occasionally requires long-term planning of movements, such as in some boss fights. Some AIs for *Touhou* exist [21], however they rely on internal game state information, such as the position of the main character and enemies, or need to swap the game assets to easily recognize the different kind of objects by mere color-coding.

We propose the creation of a real-time semantic segmentation model as a first step towards achieving the creation of such an AI. This approach aims to achieve a similar starting condition as the assets-swapping technique. In order to create a model, we first need a dataset for training and testing possible models. In the following section of this paper we show how we created a dataset for this task starting from the assets of classic era *Touhou* games[2]. Then, we give an overview of real-time semantic segmentation and we present five state-of-the-art networks. We then explore how we trained these networks using our dataset and how the resulting trained models performed at labeling generated and real images from *Touhou* games.

## II. Dataset Generation

Creating a big dataset for semantic segmentation can be an arduous task: collecting, preparing and annotating data requires large amounts of effort. Nonetheless, several large

[1]Frames Per Second
[2]The *classic era* refers to the games from *Touhou* 6 to *Touhou* 9.5

978-1-6654-3886-5/21/$31.00 ©2021 IEEE

2021 IEEE Conference on Games (CoG) | 978-1-6654-3886-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/COG52621.2021.9619112

Authorized licensed use limited to: UNIVERSITY OF BATH. Downloaded on April 16,2024 at 19:47:19 UTC from IEEE Xplore. Restrictions apply.

generic and domain-specific datasets exists, such as *Pascal VOC* [9][8], *Microsoft COCO* [14] and *Cityscapes* [7]. All these datasets were created by outsourcing the work necessary for the most time-consuming tasks, primarily image annotation. A similar endeavor could be sought to create a dataset for semantic segmentation of *Touhou* game frames, capturing thousands of screenshots of the game and then manually labeling them. However, such a dataset can be created in another way: by only manually labeling the game assets and then crafting generated game frames that can be labeled automatically.

Such an approach is possible because *Touhou* satisfies the following requirements:

- it is a primarily 2D game[3], thus creating an artificial game frame can be done by simply compositing the 2D assets on a blank canvas;
- a typical real game frame has a quite simple structure, thus it is easy to replicate by using simple operations such as scaling and rotation of the assets;
- the amount of assets is small and unambiguous enough to label manually;
- there are no assets used in semantically different ways inside the game.

Under such requirements, procedurally generating a dataset, given the manually labeled assets, should yield similar results to a manually labeled dataset of game frames when such datasets are used to train semantic segmentation models. The former approach, however, comes at a fraction of the cost and can generate an arbitrary large dataset.

We extracted the assets of *Touhou* games from 6 to 9.5 with the help of *Touhou Toolkit* [2] and we manually labeled them into 15 categories. To aid the labeling we created a *YAML*-serialized [3] format to hold the labeling information. A total of 4129 bitmaps were labeled.

We then created *ThGen*, a program to procedurally generate *Touhou*-like game frames of the main playing field. To ensure high generation speed, the programming language chosen for *ThGen* was *Rust* [15], but bindings for *Python* were also written in order to be able to access it from a more traditional *machine learning* environment. The default target size for the generated image has been set to $576 \times 672$ to match the original one.

*ThGen* reads the labeled assets extracted from the games and generates a fictional game frame of the playing field by randomly placing objects on an initially blank canvas. While placing such objects, it separately records the positions of the pixels of the placed objects to also generate a label for the created image. The process used by *ThGen* to generate an image and its associated label from a given seed is completely deterministic, thus the same dataset can be generated on different runs starting from just the labeled assets and *ThGen*.

We have selected 15 semantic labels, colored in this paper with the *hue* palette from *Seaborn* [23], that should almost entirely cover the kind of objects present in the game: *Player*, *Boss*, *StandardEnemy*, *MajorEnemy*, *StaticEnemy*, *PlayerBullet*, *EnemyBullet*, *PlayerBomb*, *PowerItem*, *PointItem*, *GameSpecificItem*, *LifeItem*, *BombItem*, *Text* and *Background*.

[3]Some backgrounds are rendered in 3D.

## III. Semantic Segmentation Networks

Semantic segmentation is the process of identifying the different kinds of objects in an image by labeling each pixel with its semantic class. The pixels belonging to different objects in the same semantic class will be labeled in the same way.

To evaluate the performance of a semantic segmentation model there are two common metrics: *pixel accuracy* and *mean intersection over union* (*mIoU*). *Pixel accuracy* is the fraction of correctly labeled pixels over the total number of pixels in an image. The *mIoU* is a measure of intersection of true positives for each semantic class. Let $n$ be the number of classes, let $T_i$ be the set of pixels labeled with the class $i$ in the ground truth image and let $P_i$ be the set of pixels labeled with the class $i$ in the predicted image. Then, the *mIoU* is calculated as follows:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{|T_i \cap P_i|}{|T_i \cup P_i|}$$

In this section we present five semantic segmentation networks specialized for real-time applications. Such specialization is needed since *Touhou* runs at $60$ *FPS* and a real-time AI agent should be able to run at the same, or higher, frame rate to have good performance. The selected semantic segmentation networks all advertise good *pixel accuracy* or *mIoU* in their original evaluations.

*Fast-SCNN* [18], short for *Fast Segmentation Convolutional Neural Network*, is an architecture for *semantic image segmentation* over high resolution images. In the original paper, the model was evaluated against the *Cityscapes* dataset, yielding a *mIoU* of $68.0\%$. They also measured the inference speed, which was $123.5$ *FPS* at a resolution of $1024 \times 2048$, on an *NVIDIA TITAN Xp GPU*.

The *Bilateral Segmentation Network V2*, *BiSeNetV2* [26], is an architecture focused on fast model inference and high scores over both *pixel accuracy* and *mIoU*. The model, in the original paper, was evaluated against three datasets, *Cityscapes*, *CamVid* [5][4] and *COCO-Stuff* [6], yielding the following results:

- *Cityscapes*: a *mIoU* of $76.6\%$ with an inference speed of $156$ *FPS* at a resolution of $1024 \times 512$;
- *CamVid*: a *mIoU* of $72.4\%$ with an inference speed of $124.5$ *FPS* at a resolution of $960 \times 720$;
- *COCO-Stuff*: a *mIoU* of $25.2\%$ and a *pixel accuracy* of $60.5\%$ with an inference speed of $87.9$ *FPS* at a resolution of $640 \times 640$.

The inference speed was measured using a *NVIDIA GeForce GTX 1080Ti GPU*.

*DFANet* [13], short for *Deep Feature Architecture Network*, is an efficient *CNN* architecture striving for a balance between speed and segmentation performance. In the original paper, the

Authorized licensed use limited to: UNIVERSITY OF BATH. Downloaded on April 16,2024 at 19:47:19 UTC from IEEE Xplore. Restrictions apply.

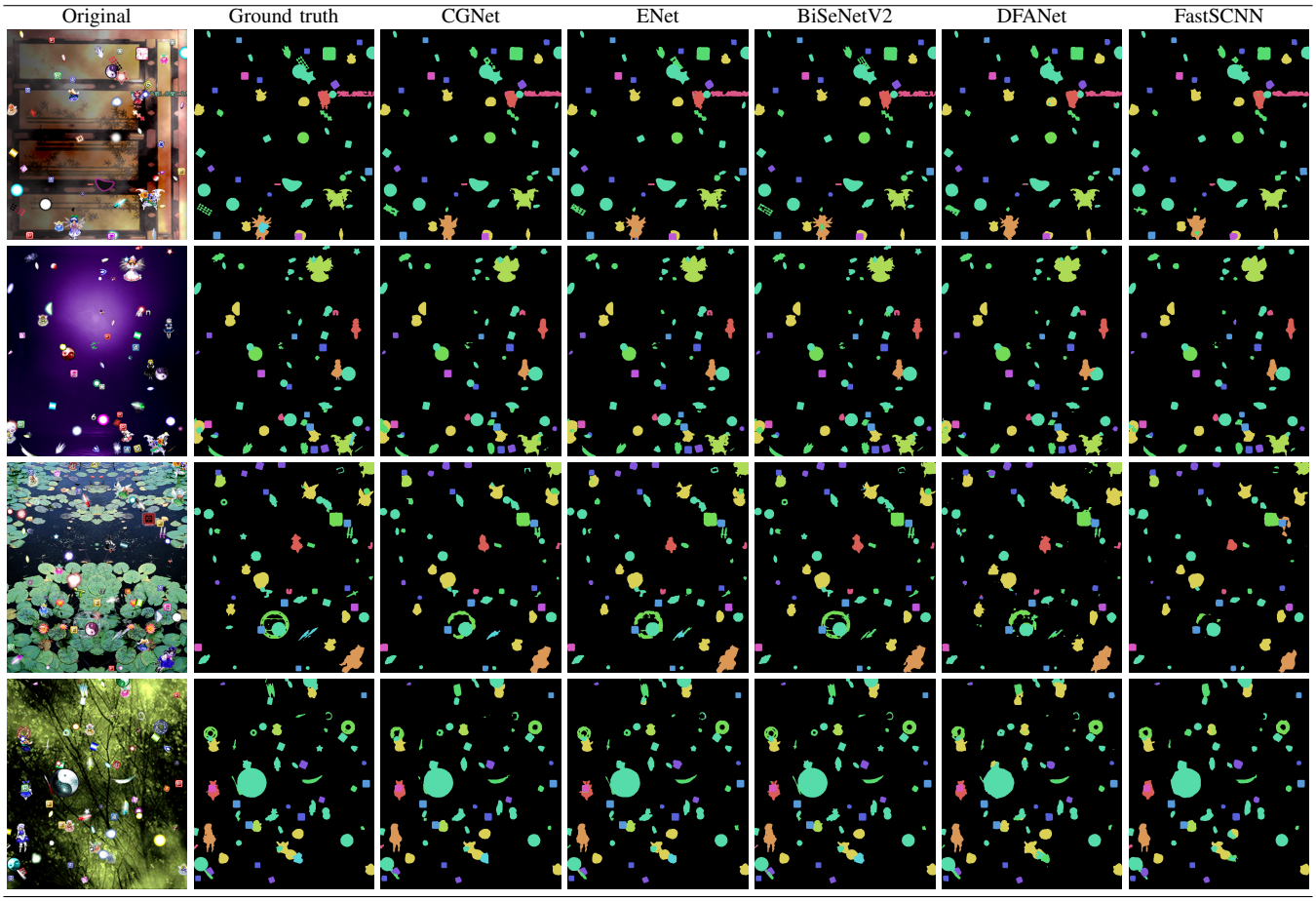


TABLE I
LABELING RESULTS OF THE BEST ITERATION FOR EACH MODEL

*DFANet* model was evaluated against the *Cityscapes* dataset, yielding a *mIoU* score of 71.3% in the best iteration of the model. The inference speed, measured on an *NVIDIA TITAN X GPU*, was equal to 100 *FPS* at a resolution of $1024 \times 1024$.

The *Efficient Neural Network*, *ENet* [17], is a neural network architecture for real-time semantic segmentation. In the original paper, *ENet* was evaluated over three datasets, achieving the following results:

- *Cityscapes*: a *mIoU* of 58.3%;
- *CamVid*: a *mIoU* of 68.3%;
- *SUN RGB-D* [22]: a *mIoU* of 19.7%.

The inference speed was measured on a resolution of $1280 \times 720$ using a *NVIDIA Titan X GPU*, which resulted in 46.8 *FPS*.

The *Context Guided Network*, *CGNet* [24], is a neural network architecture for semantic segmentation designed to work on mobile devices. In the original paper, the model was evaluated on the *Cityscapes* dataset, resulting in a *mIoU* of 64.8%. By using two *NVIDIA V100 GPUs*, the measured inference speed was 17.61 *FPS* at a resolution of $2048 \times 1024$.

| | |
|---|---|
| CPU | Intel Core i7-6700K @ 4.00GHz |
| RAM | 64 GB DDR4 @ 2133 MHz |
| GPU | NVIDIA GeForce GTX 980M |
| OS | Arch Linux — kernel 5.12.6-zen |

TABLE II
EXPERIMENTS MACHINE SYSTEM SPECIFICATIONS

## IV. EXPERIMENTS AND RESULTS

To assess the quality of our dataset we trained all the real-time semantic segmentation networks described in Section III with a generated instance of our dataset containing 16384 images for the training set and 512 for the validation set. We then evaluated the trained networks against the validation set using two common metrics for semantic segmentation: *pixel accuracy* and *mean intersection over union*. We also use them to label real game footage to give a qualitative analysis of the models performance.

All the models were implemented using the *PyTorch* framework [11]. All the training, validation and evaluation processes took place on a machine whose characteristics are described in Table II. As specified in Section II, the resolution of the images in the dataset is $576 \times 672$ and they are segmented

Authorized licensed use limited to: UNIVERSITY OF BATH. Downloaded on April 16,2024 at 19:47:19 UTC from IEEE Xplore. Restrictions apply.

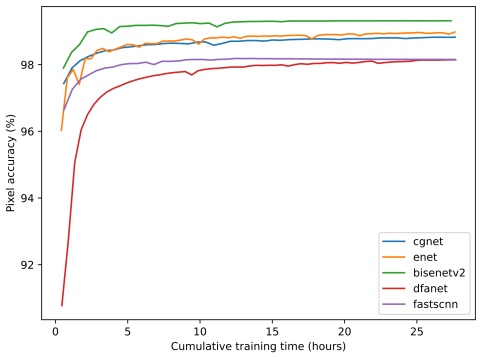


Fig. 1. *Pixel accuracy* over cumulative training time

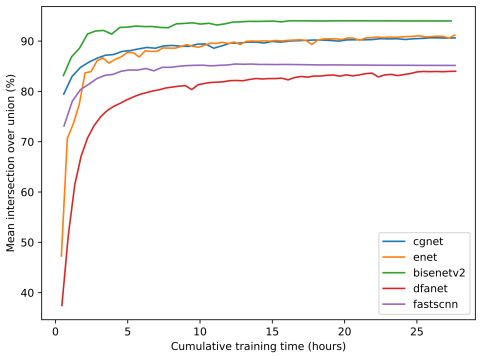


Fig. 2. *mIoU* over cumulative training time

using 15 different semantic labels.

Each model was trained for a time between 27 and 28 hours. The optimization algorithm used was *AMSGrad* [19] with a plateau-based learning rate scheduler. The loss function used was the *cross-entropy* loss function. For the *FastSCNN* and *BiSeNetV2* networks a variation of *cross-entropy* was used as suggested in their respective papers. While training, we evaluated the models using the *pixel accuracy* and *mean intersection over union* metrics. The results are reported in Figures 1 and 2, respectively.

To select the best iteration for each model, we chose the one which maximized the average of the two collected metrics. The information about the best iterations are reported in Table III, alongside the inference speed measured in *FPS*. The speed metric was obtained by averaging the speed of inference on the images of the validation set.

As we can see from Figures 1 and 2, and from Table III, the best performing model, by far, is *BiSeNetV2* both in terms of *pixel accuracy* and *mIoU*. However, the *BiSeNetV2* model has the lowest inference speed at 21.68 *FPS*, which is less than half of the frame rate of *Touhou*. The fastest model, although second worst performing, is *FastSCNN*, which achieves and surpasses the 60 *FPS* target. The *ENet* model is good enough in both regards, achieving the second position in all categories. The *CGNet* model performs similar to *ENet*, albeit with lower inference speed. The *DFANet* is the worst performer, with an inference speed lower than *FastSCNN*.

| Model | Epoch | Accuracy (%) | mIoU (%) | FPS |
|---|---|---|---|---|
| CGNet | 45 | 98.82 | 90.64 | 28.88 |
| ENet | 66 | 98.97 | 91.18 | 35.79 |
| BiSeNetV2 | 37 | 99.31 | 94.03 | 21.68 |
| DFANet | 61 | 98.14 | 84.01 | 34.34 |
| FastSCNN | 21 | 98.18 | 85.44 | 82.24 |

TABLE III
BEST ITERATION FOR EACH MODEL

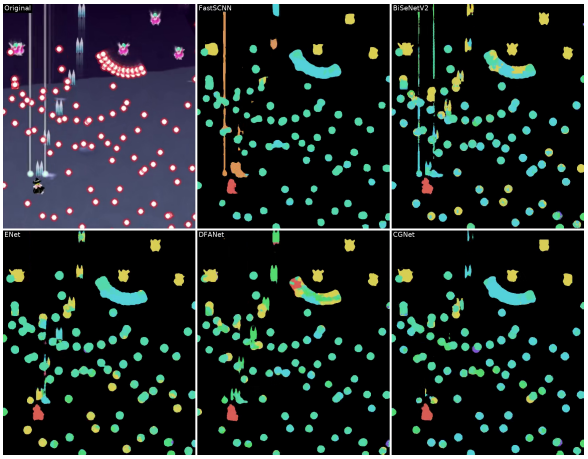


Fig. 3. Predicted labelings of a *Touhou* 6 game frame

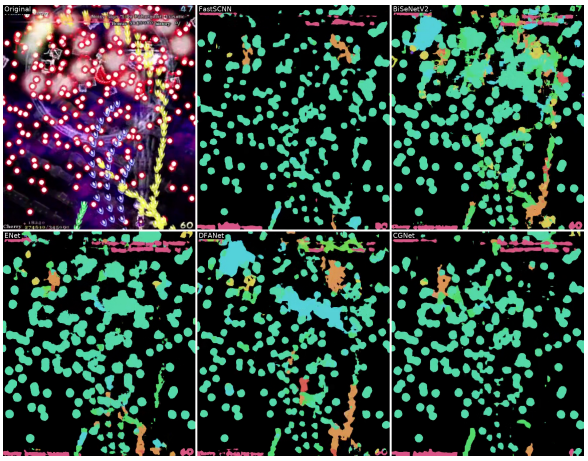


Fig. 4. Predicted labelings of a *Touhou* 7 game frame

In Table I are presented some images from the validation set, their label and the prediction made by the models. All models produce a reasonably good segmentation of the original image that closely resembles the ground truth. To assess the validity of these models on real game frames, we made this models label some video recordings of *Touhou* games and posted the results on *YouTube* [1]. Some sample frames are shown in Figures 3, 4 and 5. We can see from these videos that the performance of the models on real *Touhou* game frames drops, mainly due to the noise in the segmentation induced by the moving background, clusters of bullets and

Authorized licensed use limited to: UNIVERSITY OF BATH. Downloaded on April 16,2024 at 19:47:19 UTC from IEEE Xplore. Restrictions apply.


Fig. 5. Predicted labelings of a *Touhou* 8 game frame

visual special effects. Three models give promising results: *BiSeNetV2*, *DFANet* and *ENet*. The first two seem to be more sensitive to moving backgrounds and special effects, while *ENet* to clusters of bullets. For these reasons, the first two are the best performing for *Touhou* 6, while *ENet* does a better job with *Touhou* 7 and 8. The *FastSCNN* and *CGNet* models seem to perform not as well on real game data.

## V. Conclusions and Future Works

The research on AIs playing games from pixels is very active. Most of the best results come from the general AI research, but achieving better results for a specific game is still an easier way to go. In this paper we presented the AI-playing-from-pixels problem for *Touhou*, a bullet hell game that, albeit being massively popular in some niches, lacks an AI that can beat it at its hardest difficulty by just looking at pixels.

We proposed semantic segmentation as a first step to bridge the gap between pixel-looking AIs and internal-game-state looking AIs. To achieve a semantic segmentation model for this task, we created a dataset procedurally generated using manually labeled assets from a selected subset of *Touhou* games. We gave an overview of five possible real-time semantic segmentation models, selecting them for their speed and accuracy. We then trained these models with our dataset, and we showed that they perform well against the validation set, albeit not at the 60 *FPS* target, except for *FastSCNN*. We also evaluated these trained models on real game footage, which showed promising results, but with still too much classification noise.

In the future we plan to strengthen the dataset, by adding noise to the background, more clusters of bullets and assets from more *Touhou* games. We also want to investigate faster semantic segmentation models, to be able to reach the 60 *FPS* target while maintaining great classification accuracy.

## References

[1] Comparison on real *Touhou* videos. https://www.youtube.com/playlist?list=PL28XtHfG7kBVADdTifjlNOc9OVK-N4-5z.

[2] Touhou toolkit. https://github.com/thpatch/thtk.

[3] O. Ben-Kiki, C. Evans, and B. Ingerson. Yaml ain't markup language (yaml™) version 1.1. *Working Draft 2008-05*, 11, 2009.

[4] G. J. Brostow, J. Fauqueur, and R. Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recognition Letters*, 30(2):88–97, 2009.

[5] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *European conference on computer vision*, pages 44–57. Springer, 2008.

[6] H. Caesar, J. Uijlings, and V. Ferrari. Coco-stuff: Thing and stuff classes in context. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1209–1218, 2018.

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[8] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[10] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jaśkowski. Vizdoom: A doom-based ai research platform for visual reinforcement learning. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8. IEEE, 2016.

[11] N. Ketkar. Introduction to pytorch. In *Deep learning with python*, pages 195–208. Springer, 2017.

[12] F.-Y. Lam. Comic market: How the world's biggest amateur comic fair shaped japanese dōjinshi culture. *Mechademia*, 5(1):232–248, 2010.

[13] H. Li, P. Xiong, H. Fan, and J. Sun. Dfanet: Deep feature aggregation for real-time semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9522–9531, 2019.

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[15] N. D. Matsakis and F. S. Klock. The rust language. *ACM SIGAda Ada Letters*, 34(3):103–104, 2014.

[16] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[17] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016.

[18] R. P. Poudel, S. Liwicki, and R. Cipolla. Fast-scnn: fast semantic segmentation network. *arXiv preprint arXiv:1902.04502*, 2019.

[19] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. *arXiv preprint arXiv:1904.09237*, 2019.

[20] S. Risi and M. Preuss. From chess and atari to starcraft and beyond: How game ai is driving the world of ai. *KI-Künstliche Intelligenz*, 34(1):7–17, 2020.

[21] K. Sakai, Y. Okada, and Y. Muraoka. Developing ai for playing shooter games touhou kaeizuka. In *ICAI 2010: proceedings of the 2010 international conference on artificial intelligence (Las Vegas NV, July 12-15, 2010)*, pages 748–752, 2010.

[22] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.

[23] M. L. Waskom. Seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

[24] T. Wu, S. Tang, R. Zhang, J. Cao, and Y. Zhang. Cgnet: A light-weight context guided network for semantic segmentation. *IEEE Transactions on Image Processing*, 30:1169–1179, 2020.

[25] M. Wydmuch, M. Kempka, and W. Jaśkowski. Vizdoom competitions: Playing doom from pixels. *IEEE Transactions on Games*, 11(3):248–259, 2018.

[26] C. Yu, C. Gao, J. Wang, G. Yu, C. Shen, and N. Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *arXiv preprint arXiv:2004.02147*, 2020.

Authorized licensed use limited to: UNIVERSITY OF BATH. Downloaded on April 16,2024 at 19:47:19 UTC from IEEE Xplore. Restrictions apply.