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 strengthened 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, 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 games2. 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

1Frames Per Second
2The classic era refers to the games from Touhou 6 to Touhou 9.5
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 576×672 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, \(\textit{Boss, StandardEnemy, MajorEnemy, StaticEnemy, PlayerBullet, EnemyBullet, PlayerBomb, PowerItem, PointItem, Game-SpecificItem, LifeItem, BombItem, Text and Background.}\)

### 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 × 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] 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 × 512;
- CamVid: a mIoU of 72.4% with an inference speed of 124.5 FPS at a resolution of 960 × 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 × 640.

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

DFAter [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

\(^3\)Some backgrounds are rendered in 3D.
TABLE I
LABELING RESULTS OF THE BEST ITERATION FOR EACH MODEL

<table>
<thead>
<tr>
<th>Original</th>
<th>Ground truth</th>
<th>CGNet</th>
<th>ENet</th>
<th>BiSeNetV2</th>
<th>DFANet</th>
<th>FastSCNN</th>
</tr>
</thead>
</table>

--

The 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 × 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 × 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 × 1024.

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 × 672 and they are segmented...

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
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.

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Epoch</th>
<th>Accuracy (%)</th>
<th>mIoU (%)</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGNet</td>
<td>45</td>
<td>98.82</td>
<td>90.64</td>
<td>25.85</td>
</tr>
<tr>
<td>ENet</td>
<td>66</td>
<td>98.97</td>
<td>91.18</td>
<td>35.79</td>
</tr>
<tr>
<td>BiSeNetV2</td>
<td>37</td>
<td>99.31</td>
<td>94.03</td>
<td>21.68</td>
</tr>
<tr>
<td>DFANet</td>
<td>61</td>
<td>98.14</td>
<td>84.01</td>
<td>34.34</td>
</tr>
<tr>
<td>FastSCNN</td>
<td>21</td>
<td>98.18</td>
<td>85.44</td>
<td>82.24</td>
</tr>
</tbody>
</table>

TABLE III
BEST ITERATION FOR EACH MODEL
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