# Critic guided VAE for Crafter

A Project for Deep Learning for Datamining

Oliver Tautz

November 2, 2023

## 1 Task and Related Work

All work in this report is based heavily on the BA thesis of Lunay Cicek. He used a critic network [3] in combination with a Variational Autoencoder (VAE) [2] on *Minecraft* images. The critic supplies a reward signal, i.e. if a wood log is present or not, to the VAE. This way the VAE learns to model the presence of the wood log based on the critic's signal. After training, the model will hallucinate logs into pictures without them given a high critic value and remove present logs from images given a low one. One possible application explored in Lunay's thesis is to generate difference masks that highlight the location of the reward object.

In my project, I tried to apply the same technique to another game, called *Crafter* [1]. *Crafter* is similar to *Minecraft*. It is an open-world survival game where an agent must collect materials and craft objects. The biggest difference between *Crafter* and *Minecraft* is the perspective. *Minecraft* is a fully 3D game, while *crafter* uses a 2D top view.

I focused on a very basic task which is to collect wood from trees. For that, the agent has to walk up to the tree and hit a button. My goal is to have the VAE add or remove trees based on the reward signal given.

The full implementation can be found at
https://github.com/Oliver-Tautz/Critic-VAE. Fully working Google implementations and model weights are included.

## 2 Crafter Dataset

The first step was to find training examples. Thankfully [1] supplied a dataset of human expert playthroughs consisting of 100 episodes and over 18000 individual frames. It is richly annotated with reward, location, and inventory data. For the purpose of this project, I processed it into pairs $(x, y)$ such that for each frame $x$ there is a label of $y = 0$ or $y = 1$. $y$ is set to 1 if in the next frame, a tree is chopped and set to 0 otherwise.

In another preprocessing step, I removed frames that are too far away from the reward. Only the frames before a tree is chopped are used for training and testing, with a window size of 20. This is good for two reasons, first only reward-relevant frames are used and second, other biomes are removed making the task easier both for the critic and the VAE.

## 3 Training the Critic

Next, a suitable critic network had to be trained. I used the same architecture as Lunay and [3]. It is a simple Convolutional Neural Network (CNN) with fixed pooling layers and dropout in the last layers. The training was mostly successful.

As the reward class with $y = 1$ is heavily underrepresented, it is oversampled in the training set to match the number of samples with $y = 0$.

Training is done with binary cross-entropy loss and a batch size of 32 for 400 epochs. Convergence on the training set is good, but there are signs of overfitting as can be seen in Figure 1. The critic reached an accuracy of about 80% on the validation set, which is not great, but seems to be mostly sufficient for the purpose of this project.

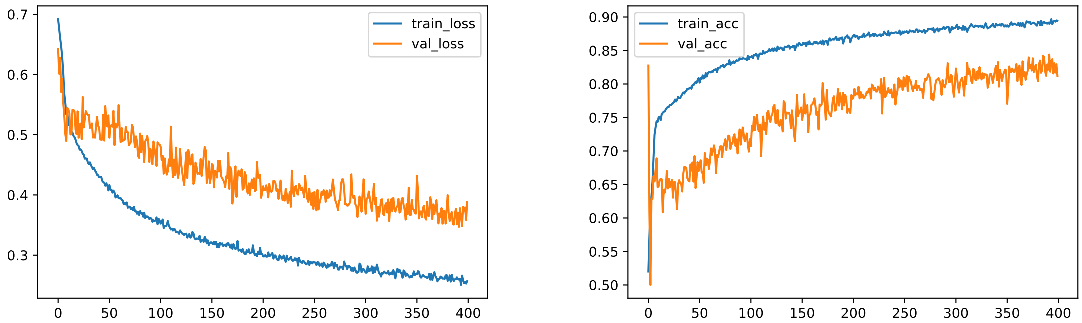


Figure 1: Training and validation loss of the critic model. It reaches a training accuracy of about 90% but shows signs of overfitting.

## 4 Training the VAE

To make the VAE work on *Crafter* more work needed to be done. The original model sampled the image down by a factor of 16, leaving only a small 4x4 latent image. This lead to very blurry reconstructions that lack detail. I removed two pooling layers from the encoder and the two corresponding upsampling layers from the decoder to bring the latent image size up to 16x16. Smaller image details are important in *Crafter* as the entities in the image, such as trees and cows are rather small. To make the new model computationally feasible on similar hardware the maximum amount of latent channels needed to be reduced from 256 to 64 in addition to halving the latent VAE dimension from 64 to 32. Because it worked well on *Minecraft*, the loss function remained identical using MSSIM with additional Kullback–Leibler divergence weighted 0.2. Another difference is that the raw logit of the critic is supplied to the VAE, while lunay used a value between 0 and 1.

For training, I use the same trick as lunay. The images are sampled by critic value. They are split into three classes, low($\leq 0.25$), mid(0.25-0.7) and high($> 0.7$). Afterward exactly one third of the dataset is sampled from each class. This is done to make the VAE see more examples of pictures with high or medium-high critic value. The VAE was trained for 100 epochs on 50000 samples per epoch.

Figure 2 indicates that the VAE seems to generalize well, as the validation loss does not diverge strongly from the training loss.

## 5 Results

All result figures you see are drawn from the validation set. Looking at Figure 3 reveals the training was a success. The VAE adds Trees to pictures if supplied with a high critic value, and removes most trees given a low critic value. While not perfect, this shows the VAE can learn the relevant shape and/or color of the trees from the reward signal alone. Figure 4 further strengthens the positive impression. The difference between a reconstruction with high and low critic value can highlight trees in the frame.

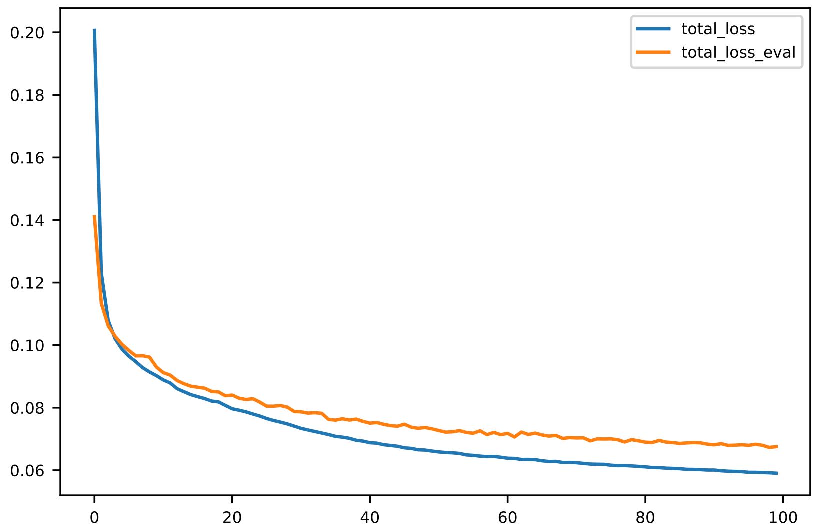


Figure 2: Loss curve of VAE training. Good convergence with no signs of overfitting

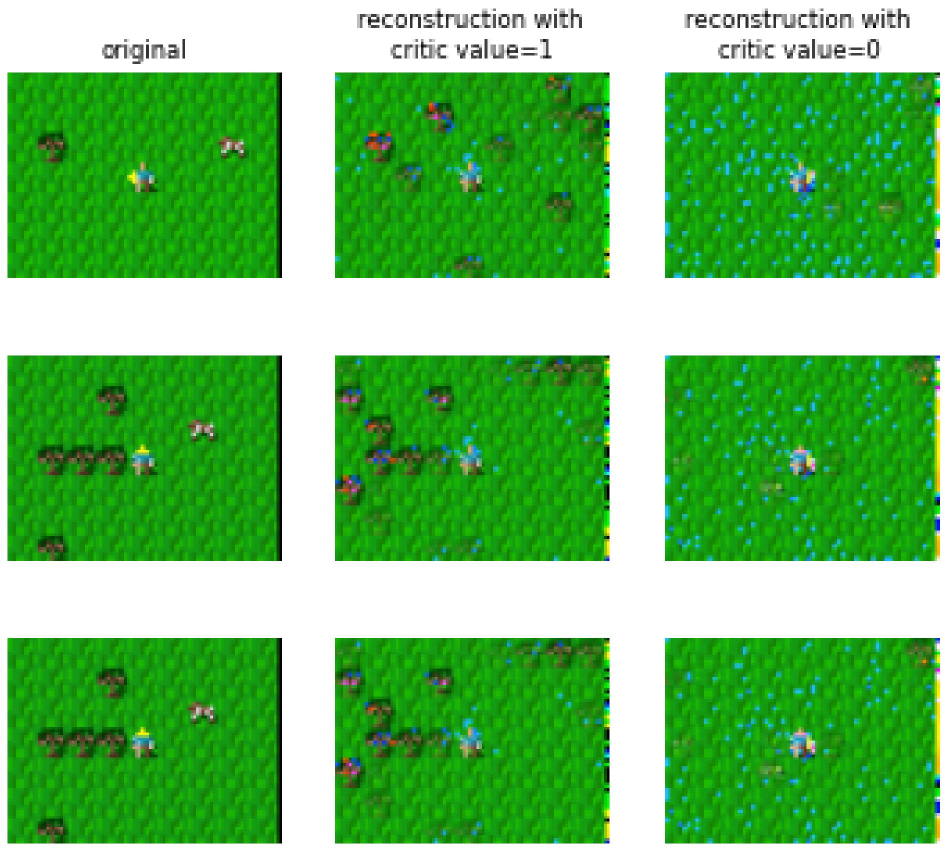


Figure 3: Example of reconstruction working as desired. Additional trees are hallucinated by the VAE given a high critic value. The trees are removed when using a low critic value.

While working well most of the time, the VAE still has some problems. Sometimes this does not work and imaginary trees or other parts of the picture can be masked. All reconstructions exhibit artifacts and errors, such as red instead of brown in the trees and blue instead of green for the ground. Also, other objects such as cows and crates are removed.

The additional Trees generated with a high critic value seem to be placed at random and not all trees originally present appear to be reconstructed. In some cases (see Figure 5) the VAE will hallucinate new Trees even with a low critic value.

Another big problem circumvented by sample selection is that of other biomes and the day/night cycle. *Crafter* features a diverse set of biomes and simulates the night by making the world look darker. If tested on other biomes or nighttime frames the VAE creates random images of known tiles not making much sense. For some reason, the reconstruction is much worse if given a low critic value (see Figure 6).

## 6 Discussion and Outlook

Generally, the VAE shows great promise in modeling the complex distribution of a reward in a video game. I showed that the approach used by Lunay on *Minecraft* can be applied to a visually very different environment with little adjustment, demonstrating its generality. Its biggest advantage is, that it can be applied as un- or self-supervised technique if a given environment supplies a stable and reliable reward signal.

But in this particular case, performance could be improved. The biggest problem seems to be the reward location. Trees can be all around the agent, but if they don't get chopped there is no reward. On the other hand, the agent can get a reward from just one tree on the screen. So the VAE seems to have difficulties finding the correct location of the reward. The window size was introduced to counter this problem, but it seems it is not sufficient.

Another problem is critic performance. 80% seems to be enough for the VAE to recognize the reward, but too many wrongly labeled samples remain, making the performance unstable. For the critic, the same location and causation problem of the reward remains.

I think the fastest approach to improve this model would be to generate a dataset where the critic can reach an accuracy $> 95\%$ and then train the VAE with the same setup. This dataset should include more samples with no trees present at all, as these examples are rare which might lead to the sometimes confusing results for low critic values. One possibility would be to take the given dataset and clean it manually. Another would be to have an algorithm e.g., find the forest biome frames and count the trees in it. That way a very clean dataset could be constructed.

For a game like *Crafter* it could be interesting to also try to generate and remove other objects such as rocks or maybe diamonds, especially the ones that are more sparse as they might provide a clearer signal for the models.

As an outlook, I thought about an ensemble of models. If there were a model for each item in all biomes and a predictor that could tell which biome is shown, the ensemble could supply data for the whole game. But to achieve this the results would need to be much more stable and first, the single cases need to be improved.

## References

[1] Danijar Hafner. *Benchmarking the Spectrum of Agent Capabilities*. 2021. URL: https://arxiv.org/abs/2109.06780.

[2] Diederik P Kingma and Max Welling. *Auto-Encoding Variational Bayes*. 2013. DOI: 10.48550/ARXIV.1312.6114. URL: https://arxiv.org/abs/1312.6114.

[3] Andrew Melnik et al. *Critic Guided Segmentation of Rewarding Objects in First-Person Views*. 2021. DOI: 10.48550/ARXIV.2107.09540. URL: https://arxiv.org/abs/2107.09540.

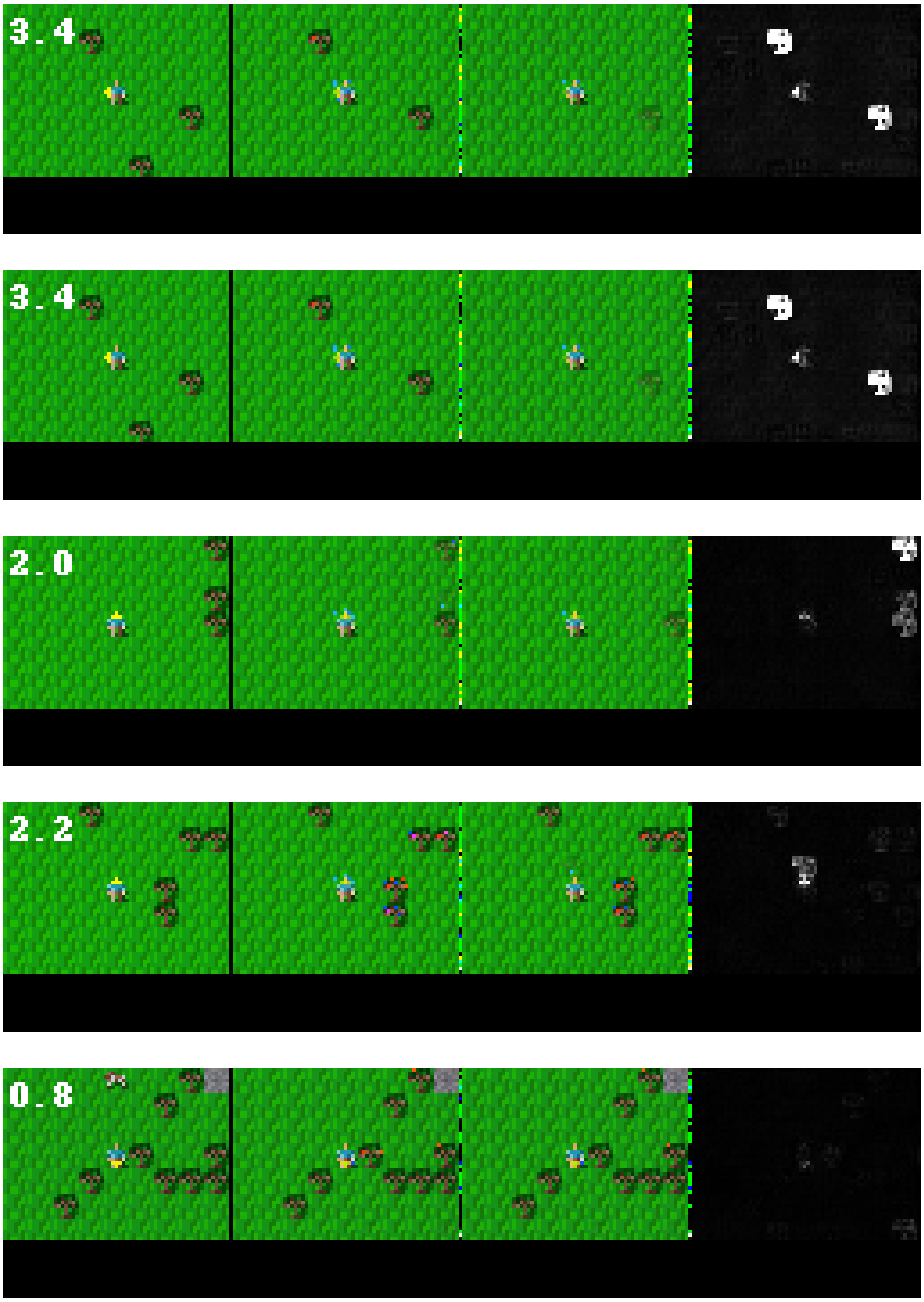


Figure 4: Examples of difference masks. The difference between high and low critic-value reconstruction highlights the trees. In the top left is the raw critic output. In some cases imaginary trees are highlighted and sometimes nothing is masked.

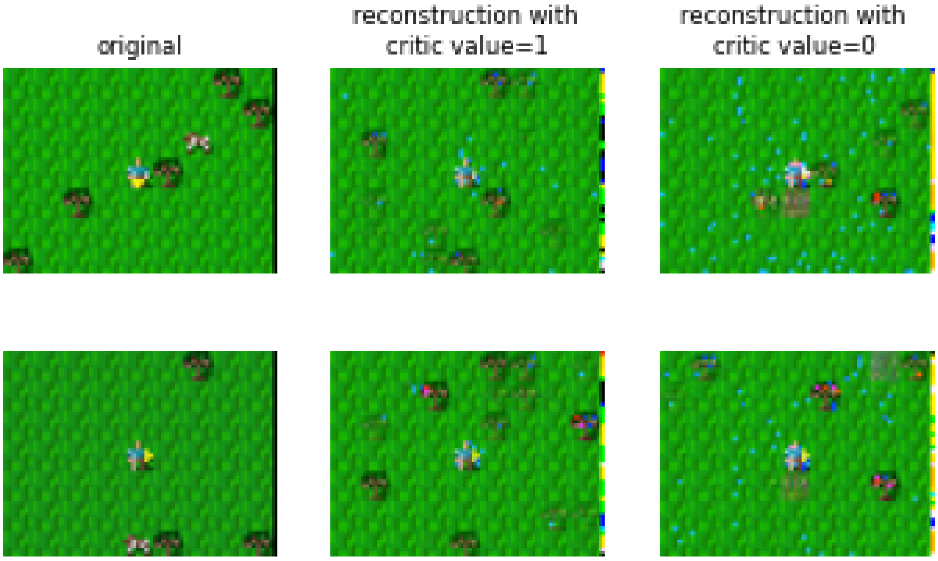


Figure 5: Examples of bad reconstruction, even with low critic value additional trees are hallucinated.

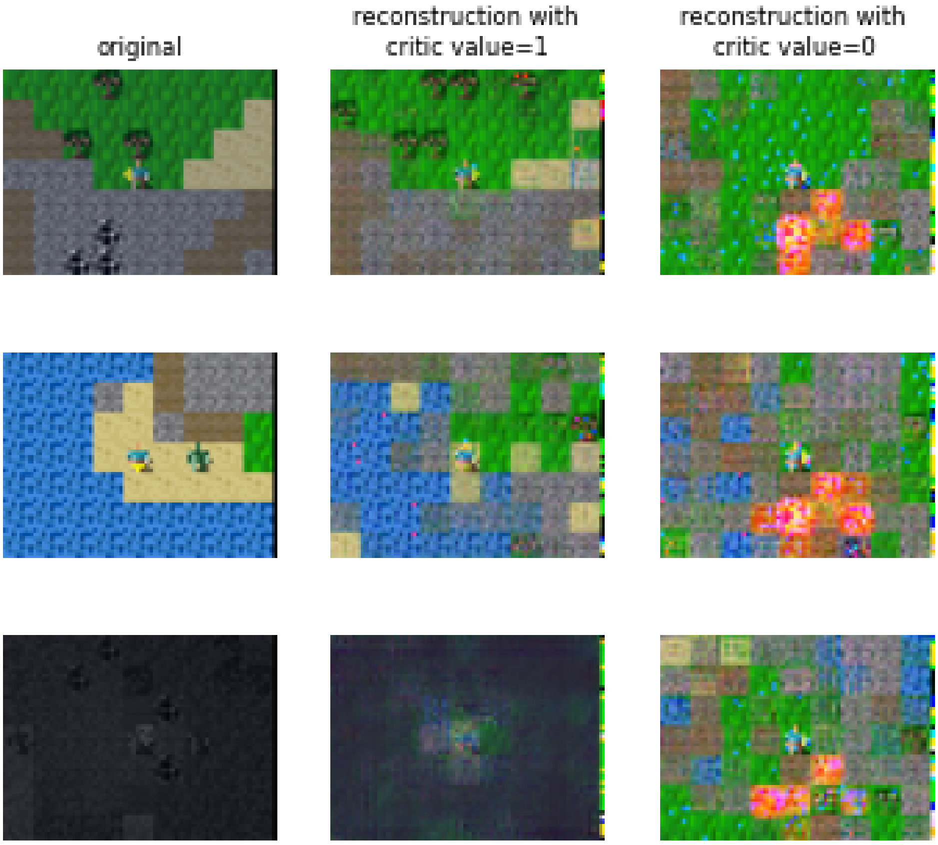


Figure 6: Examples of bad reconstruction in other biomes. As they are missing from the training set the VAE cannot reconstruct them properly.