Reinforcement Learning for Bomberman

Final Project for the lecture:
Fundamentals of Machine Learning

Daniel Gonzalez, Matrikel Nr.: 3112012
Maria Regina Lily, Matrikel Nr.: :)
Ferdinand Vanmaele, Matrikel Nr.: ;)
(Order alphebitcally, don't take it personal)

14.03.2019

1 Abstract

SOME TEXT - Look at the en in Conclusion for some useful links

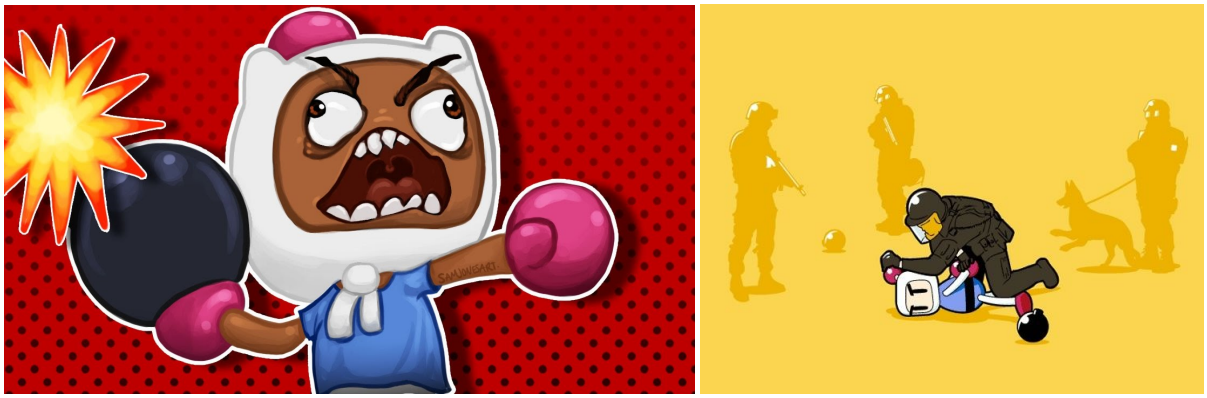


Figure 1: he acts as a dement (left) he is often seen as a terrorist(right)

2 Introduction

[TODO: some text]

3 Explaining the Framework

Features:

A1: Assumption:
1) BOMB -> others[i] -> agent (don't die)
2) BOMB -> BOMB -> agent (die)
3) BOMB -> crate -> agent (don't die)

C1: Consideration:
use s.bomb_power
C2: Consideration:
use bombs left

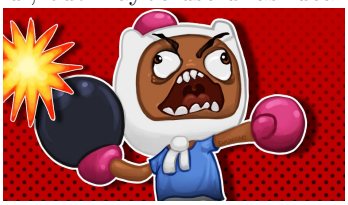
- FOR STEP 1:
  - **1.)** Reward the best possible action to a coin, if it is reachable F(s,a)=1 , otherwise F(s,a)=0. 'BOMB' and 'WAIT ' are always 0.
- FOR STEP 1 & 2:
  - **1.)**
    DONE
  - **2.)** Penalize if the action follows the agent to death (F,s)=1, F(s,a)=0. otherwise.
    DONE, it could be slightly improved
    REMARK: C1?
  - **3.)** Penalize if the action follows the agent into a "save":(Where the bomb won't at some point explode) position. F(s,a)=1 otherwise F(s,a) = 0. Bombs are always set to 0.
    TO BE DONE
  - **4.)** Reward the minimal distance that follows to safety if you are in a "Danger zone" (as defined in **3**.) F(s,a)=1 otherwise F(s,a) = 0. If you are not in a "Danger zone" then F(s,a) = 0 for all actions. For Bombs always set F(s,a) = 0.
    TO BE DONE: -> LILY
  - **5.)** Penalize invalid actions. F(s,a) = 1, otherwise F(s,a) = 0 .
    DONE
    REMARK: C2
  - **6.)** Reward when getting a coin F(s,a) = 1, otherwise F(s,a) = 0 .
    DONE
  - **7.)** Reward putting a bomb next to a block.
    F(s,a) = 0 for all actions and otherwise F(s,a) = 1 for a BOMB if we are next to a block.
    DONE
  - **8.)** Reward (if there are no blocks anymore ? and no coins?) the available movements F(s,a) = 1, otherwise F(s,a) = 0 . Bombs = 0, WAIT =1 ?
    TO BE DEFINE and DONE

Weitere Ideen:

- penalize distance between our agent and the nearest reachable crate.

TD(0) learning

[TODO: some text] Probably not useful, but maybe useful as idea for loading images



Some text


Some text

Figure 2: Some text

Gradient descent

[TODO: some text] Probably not useful, but maybe useful as idea for loading images

Figure 3: Some text

Bla blah

Other sub chapter

[SOME TEXT].

Figure 4: some text

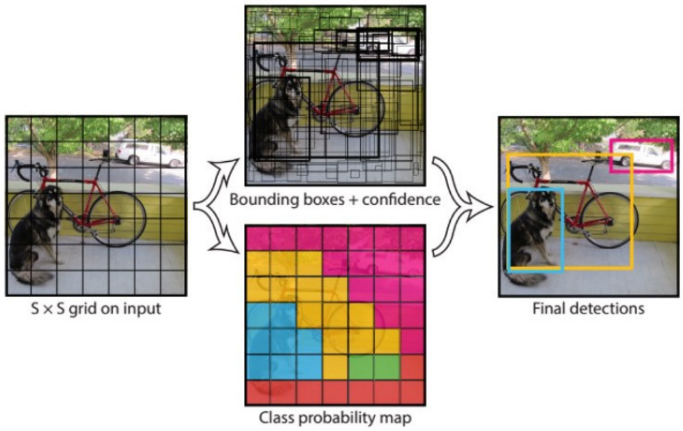


Figure 7: SOME TEXT.

4 Related work

[Some text]

Some text
In 2015, **SOME TEXT**, [9]) (see Figure 5).

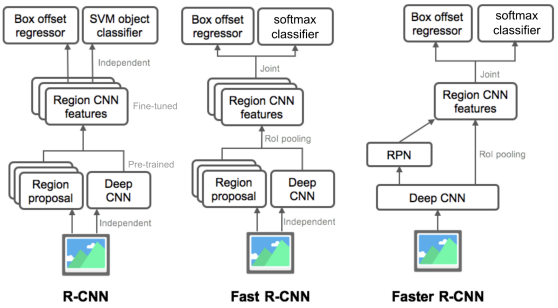


Figure 5: SOME TEXT

bla bla

some citation examples [4][3]

5 See figures above (Strategy to also use space and have more variants for the text

SOME TEXT

The YOLO approach to object detection

SOME TEXT

Anchor boxes

SOME TEXT

SUBSECTION

SOME TEXT

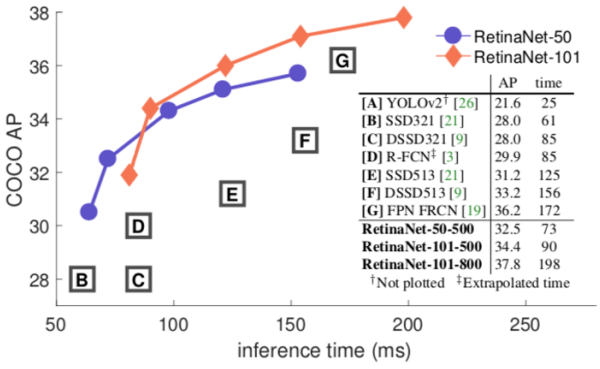


Figure 6: some text

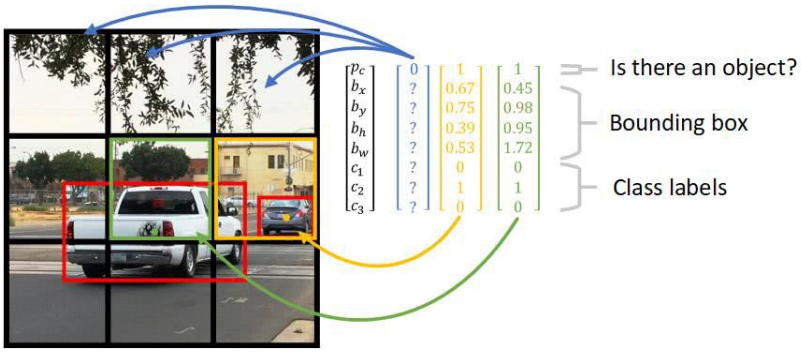


Figure 8: SOME TEXT

SUBSECTION

6 Feature extraction

Some text

7 Evaluation

Some text

8 Results: some Table & other way of loading images

some text

some table

Dataset situation				
name	description	precision	recall	mAP
1 - Simple	Paste cards on simple canvases *random rotations, brightness, blurring*	0.974	0.996	**0.991**
2 - Medium	Paste randomly scaled cards on simple canvases *random rotations, brightness, blurring*	0.946	0.988	**0.989**
3 - Elaborate	Paste randomly scaled cards on textures *random rotations, brightness, blurring*	0.937	0.978	**0.971**
4 - Hardest	Paste randomly scaled cards on textures *random rotations, brightness, blurring, less zoom*	0.940	0.983	**0.973**

Table 1: Precision and recall values have been calculated using a IOU threshold of 0.5. mAP values are based on averaged precision values over IOU thresholds of $[0.1, 0.2, \ldots 0.8, 0.9]$

success: classification:
Ad: 0.99995, **Ad:** 0.99997

success: classification
As: 0.99757, **As:** 0.99931

success: classification
Jd: 0.99967, **Jd:** 0.99992

Figure 9: Successful cases of detection of images that are pretty representative of the training distribution

Results of further work

Transfer learning

Webcam deployment

9 Discussion and Future Work [Frank & Daniel]

Overview

Training process

Deployment on a webcam

Future Work

10 Conclusion

`https://github.com/mlteam-ws2018/RL_boom`. SOME USEFUL LINKS (for the reportwriting):
[for motivation]: `https://www.youtube.com/watch?v=xMP-JqFQ_l4`
[gd, policy, q-learning]: `https://www.ias.informatik.tu-darmstadt.de/uploads/Theses/Sharma_BScThesis_2012.pdf`
[gd, policy, q-learning]: `https://repositorio-aberto.up.pt/bitstream/10216/91011/2/176444.pdf`

References

[1] Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., , and Vedaldi, A. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (2014).

[2] Everingham, M., Eslami, S. M. A., Van Gool, L., Williams, C. K. I., Winn, J., and Zisserman, A. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision 111*, 1 (Jan. 2015), 98–136.

[3] Lin, T., Dollár, P., Girshick, R. B., He, K., Hariharan, B., and Belongie, S. J. Feature pyramid networks for object detection. *CoRR abs/1612.03144* (2016).

[4] Lin, T., Goyal, P., Girshick, R. B., He, K., and Dollár, P. Focal loss for dense object detection. *CoRR abs/1708.02002* (2017).

[5] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C., and Berg, A. C. SSD: single shot multibox detector. *CoRR abs/1512.02325* (2015).

[6] Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. You only look once: Unified, real-time object detection. *CoRR abs/1506.02640* (2015).

[7] Redmon, J., and Farhadi, A. YOLO9000: better, faster, stronger. *CoRR abs/1612.08242* (2016).

[8] Redmon, J., and Farhadi, A. Yolov3: An incremental improvement. *CoRR abs/1804.02767* (2018).

[9] Ren, S., He, K., Girshick, R. B., and Sun, J. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR abs/1506.01497* (2015).