CNN Encoder to Reduce the Dimensionality of Data Image for Motion Planning

Janderson Ferreira[1], Agostinho A. F. Júnior[1], Yves M. Galvão[1]
Bruno J. T. Fernandes[1] and Pablo Barros[1,2] *

1- Universidade de Pernambuco - Escola Politécnica de Pernambuco
Rua Benfica 455, Recife/PE - Brazil

2- Cognitive Architecture for Collaborative Technologies (CONTACT) Unit
Istituto Italiano di Tecnologia, Genova, Italy

Abstract. Many real-world applications need path planning algorithms to solve tasks in different areas, such as social applications, autonomous cars, and tracking activities. And most importantly motion planning. Although the use of path planning is sufficient in most motion planning scenarios, they represent potential bottlenecks in large environments with dynamic changes. To tackle this problem, the number of possible routes could be reduced to make it easier for path planning algorithms to find the shortest path with less efforts. An traditional algorithm for path planning is the A*, it uses an heuristic to work faster than other solutions. In this work, we propose a CNN encoder capable of eliminating useless routes for motion planning problems, then we combine the proposed neural network output with A*. To measure the efficiency of our solution, we propose a database with different scenarios of motion planning problems. The evaluated metric is the number of the iterations to find the shortest path. The A* was compared with the CNN Encoder (proposal) with A*. In all evaluated scenarios, our solution reduced the number of iterations by more than 60%.

1   Introduction

Present in many daily applications, path planning algorithm helps to solve many navigation problems. Dijkstra proposed one of the first path planning algorithms in 1956 [1]. Although this algorithm was capable of finding the shortest path between two points, its use was limited to nodes with positive weights, and the computational cost is quadratic. To decrease the time complexity and allow negative weight between the nodes, others algorithms based on Dijkstra's algorithm were created, such as A* and Dijkstra's algorithm with Fibonacci heap [2].

Motion planning is a specific application of path planning regarding autonomous robots. Different types of approaches can be used in motion planning, like Grid-based search (which transform the environment in a grid-mesh) [3], Interval-based search (similar to grid-based search it but uses space data instead of a grid) [4] and Reward-based (similar to a reinforcement learning in deep learning) [5]. Commonly, path planning using Grid-based and Interval-based

presents some issues in a large or dynamic environment, due to the time complexity of these algorithms and the need to recalculate the route if a new object was found in an already mapped space [6].

An important reason for the time complexity problem in the motion planning is because there are usually a lot of possible ways to get to the destination. Regarding the topic of information reducing, some algorithms of decomposition are applied in big data problems: Principal component analysis (PCA) [7] and Truncated Singular value decomposition (TSVD) [8], Non-negative matrix factorization (NMF) [9], among others. Still, recently, these same solutions are used to reduce the number of possible ways to find the best route between two points [10, 11, 12]. However, many of the solutions proposed to compact data are not able to do that considering the non linearity of the data. In 2006 [13], Hinton, G. proposed the use of Autoencoders to reduce the dimensionality of data, he showed how it is possible to represent data with much less information, because his proposal considered the non-linearity data. Since then, Autoencoders have been used to compress data, helping to solve various machine learning problems [14].

In this paper, we investigate the application of a Convolutional Neuronal Network Encoder to reduce the number of routes. Therefore, the combination between our proposal and any algorithm of path planning must spend less iterations to find the best path than conventional A*. Moreover, we also propose a new dataset for evaluating the performance of the proposed method.

2   CNN Encoder

Convolutional Neural Networks have an excellent capability to extract high-level features. For this reason, currently, it is being used to solve many problems in deep learning and computational vision problems[15]. Simultaneously, Autoenconders are being used to code data information in unsupervised learning [16]. They are trained to reconstruct the input data using fewer data than the original input; this way, many times, they can eliminate useless information.

Thinking about the good results of CNNs and the capability of the Autoenconder to reduce the dimensionality of the data, we propose a CNN Encoder to eliminate useless routes from 2D maps.

2.1   Model architecture

The process of building the architecture was made interactively. After many combinations of parameters and applications of transfer learning of famous CNNs, like VGG16 and Resnet50, which gets a fewer number of iterations, can be seen in table 1.

3   Motion Planning Database

To check the efficiency of our solution and compare it to the conventional technique, we created an image database containing different scenarios, obstacles

Table 1: Architecture of the CNN Encoder

	Layer	Filters	Kernel Size	Activation	Batch Norm	Dropout
1	Image	-	-	-	-	-
2	Conv	64	3x3	ReLu	True	-
3	Conv	128	3x3	ReLu	False	30%
4	Max-Pool	-	3x3		-	-
5	Conv	256	3x3	ReLu	True	-
6	Conv	512	3x3	ReLu	False	30%
7	Dense	256	-	LeakyReLU	True	30%
8	Dense	512	-	LeakyReLU	True	30%
9	Dense	1024	-	LeakyReLU	True	30%
10	Dense	3600	-	Tanh	False	30%

and goals. Figure 1 shows instances of all proposed scenes. With our database, it is possible to evaluate the generalization capability of our model, applying our proposed approach in different situations and scenarios. Our database contains five different scenarios, each of them composed of 10000 scenes (RGB images with 60×60 pixels), where each one has some fixed obstacles and others that are arranged randomly over the rest of the scene. Also, for each of these, there is an image answer, representing the expected path, created based on grid search technique.
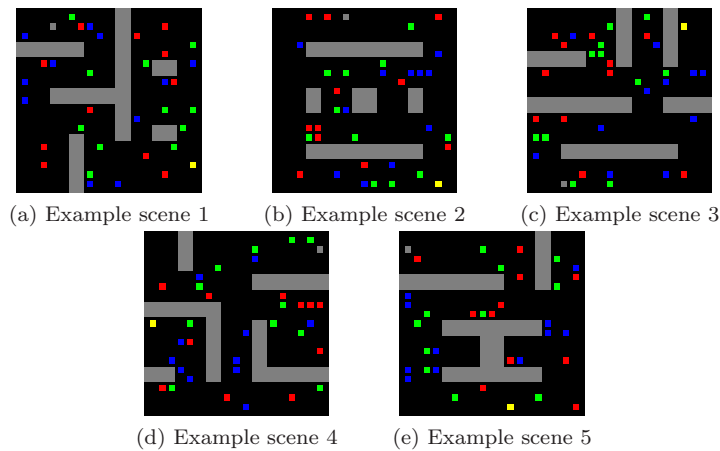

(a) Example scene 1   (b) Example scene 2   (c) Example scene 3

(d) Example scene 4   (e) Example scene 5

Fig. 1: The instances of proposed the database. The yellow and gray pixels, respectively, represents the start and the end of the path.

4 Experimental results

To evaluate our work, we compare the number of iterations of A*, and the CNN Encoder (our proposal) with A*. The best solution is the one which found the best route with the least number of iterations. The metrics that we used to

evaluate the obtained results, how we split the database, the architecture, and the output processing are as following:

- Metrics: Number of iterations. Represents the number of attempts until the algorithm found the shortest path.

- Split database: The database was split by percent split validation, 80% to train set, and 20% to test set.

- Output processing: The output processing has 5 steps, all of them can be seen in the Figure 2. They are:

  1. To Transform the output of the CNN Encoder (3600 values) to a Gray scale image with size 60x60.
  2. To Apply dilatation technique with kernel size equal to 3x3.
  3. To Binarize the image with threshold equal to 50.
  4. To Overlap the original scene and the processed output.
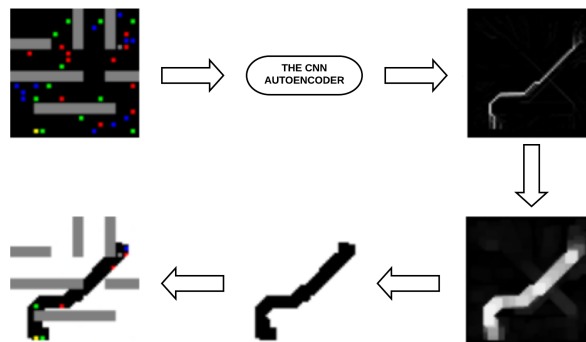 5. To Apply the A* to find the shortest path, and to count the number of iterations.

Fig. 2: The output processing of the CNN Encoder and comparison with the original input image. All black pixels in the first and last images represent the walkable paths.

Analyzing the results of Table 2, we can see that in all database scenarios, the number of iterations of the proposed solution presents an improvement of over 60% compared to the conventional A*, which means that the number of iterations decreased considerably.

Observing the Image 3 note which the A* many times try to find the shortest path using paths completely wrong, including going to the opposite side of the goal, it makes which the solution spend much time. On another hand, after the application of our proposal, the A* find the shortest path using fewer iterations. The CNN Encoder went almost directly to the goal and found that in much fewer iterations.

Table 2: Comparison between the number of iterations of A* and the number of iterations of CNN Encoder + A*

	Scene 1	Scene 2	Scene 3	Scene 4	Scene 5	Sum
A*	3401942	2756298	1797817	2382324	2478137	12816518
Proposal+A*	1006854	911499	855623	890303	915310	4579589
Difference	2395088	1844799	942194	1492021	1562827	8236929
Difference(%)	29,60%	33,07%	47,59%	37,37%	36,94%	35,73%
Improvement	70,40%	66,93%	52,41%	62,63%	63,06%	64,27%

(a) The paths taken by A* until found the shortest path

(b) The paths taken by The CNN Enconder and A* until found the shortest path

Fig. 3: These images illustrate output of each explored algorithm.

5   Conclusion

This work aimed to show the capabilities of a new algorithm based on the deep learning encoder to eliminate useless paths for motion planning algorithms.

From the obtained results, we can assume that the proposed CNN architecture was able to learn to avoid of fixed and dynamic obstacles regardless of the presented scenario. This way, it is possible for other researchers to use our solution and our methods to improve their results on autonomous navigation issues. As such, our contribution is the CNN Enconder architecture, the database, and database creation methods using random obstacles, which are useful for Encoder learning process to motion planning tasks.

As future works, we hope to compare the efficiency of our solution with other motion planning algorithms, and data reduction techniques combined with path planning. Furthermore, we intend to study the computational cost of the proposed solution and other solutions in the literature, because many robotic solutions have limited hardware.

References

[1] Thomas J Misa and Philip L Frana. An interview with edsger w. dijkstra. *Communications of the ACM*, 53(8):41–47, 2010.

[2] Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic deter-
mination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*,
4(2):100–107, 1968.

[3] David Šišlák, Premysl Volf, and Michal Pechoucek. Accelerated a* trajectory planning:
Grid-based path planning comparison. In *19th International Conference on Automated
Planning and Scheduling (ICAPS), Thessaloniki, Greece, Sept*, pages 19–23. Citeseer,
2009.

[4] Andre Gaschler, Ingmar Kessler, Ronald P. A. Petrick, and Alois Knoll. Extending the
knowledge of volumes approach to robot task planning with efficient geometric predicates.
In *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, may
2015.

[5] I. Jeong, W. Ko, G. Park, D. Kim, Y. Yoo, and J. Kim. Task intelligence of robots: Neural
model-based mechanism of thought and online motion planning. *IEEE Transactions on
Emerging Topics in Computational Intelligence*, 1(1):41–50, Feb 2017.

[6] Yu Fan Chen, Shih-Yuan Liu, Miao Liu, Justin Miller, and Jonathan P How. Motion
planning with diffusion maps. In *2016 IEEE/RSJ International Conference on Intelligent
Robots and Systems (IROS)*, pages 1423–1430. IEEE, 2016.

[7] Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space.
The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science,
2(11):559–572, 1901.

[8] Simon Du, Yining Wang, and Aarti Singh. On the power of truncated svd for general
high-rank matrix estimation problems. 02 2017.

[9] Daniel D. Lee and H. Sebastian Seung. Algorithms for non-negative matrix factorization.
In *Proceedings of the 13th International Conference on Neural Information Processing
Systems*, NIPS'00, pages 535–541, Cambridge, MA, USA, 2000. MIT Press.

[10] F. Carreira, J. Calado, C. Cardeira, and P. Oliveira. Navigation system for mobile robots
using pca-based localization from ceiling depth images: Experimental validation. In
*2018 13th APCA International Conference on Automatic Control and Soft Computing
(CONTROLO)*, pages 159–164, June 2018.

[11] Y. Xiang, W. Niu, J. Liu, T. Chen, and Z. Han. A pca-based model to predict adversarial
examples on q-learning of path finding. In *2018 IEEE Third International Conference
on Data Science in Cyberspace (DSC)*, pages 773–780, June 2018.

[12] Jianwei Wu, Deer Bin, Xiaobing Feng, Zhongpu Wen, and Yin Zhang. GA Based Adaptive
Singularity-Robust Path Planning of Space Robot for On-Orbit Detection. *Complexity*,
2018:11, 2018.

[13] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural
networks. *Science*, 313(5786):504–507, 2006.

[14] Dharin Parekh, Ankitesh Gupta, Shharrnam Chhatpar, Anmol Yash Kumar, and Manasi
Kulkarni. Lip reading using convolutional auto encoders as feature extractor, 2018.

[15] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436,
2015.

[16] Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural
networks. *AIChE Journal*, 37(2):233–243, 1991.