

# General Grid Recognition and Logic-Driven Gameplay for Intelligent Robots

Tayyab Ateeq<sup>1,\*</sup>

<sup>1</sup>Department of Mathematics and Computer Science, University of Calabria, Rende (CS), Italy

## Abstract

The recognition of complex visual patterns, such as grid structures, is a trivial task for humans. In contrast, enabling a robot to reliably detect and interpret such patterns remains a significant challenge in the field of autonomous perception. We recognise grid-like patterns in our daily life such as a grid of windows in high-rise buildings, grid-like arrangement of products on shelves in hypermarkets and superstores, grid design in fabrics and mobile games based on grid arrangement of objects such as match-3 and chess games. This research aims to develop a *Grid Detector* to identify a grid-like structure in a given image and then detect and classify objects having different shapes and colours within the cells of the grid. Such a framework could be used, for instance, to auto-detect and count the products placed on store shelves; it could also be used in recognizing grids in match-3 or chess games and become part of an iterative decision-making pipeline, allowing a delta robot to play such games. Furthermore, the goal of this research work is to design a general framework since we expect the architecture to be reusable on many different types of images such as buildings, store shelves and board games. This research currently aims to validate the proposed method by adapting an extensive dataset involving a variety of mobile games of different types and resolutions such as *Candy Crush Saga*, *Candy Fever*, *Chess*, *Checkers* and many more.

## Keywords

Computer Vision, Logic Programming, Object Recognition

## 1. Introduction

The notion of intelligent robotics fundamentally depends on advanced computer vision techniques that allow robots to perceive their environment and make autonomous decisions through reasoning. Within this context, this research investigates computer vision methods for the detection of grid-like structures and the recognition of objects arranged within them — a pattern commonly observed not only in mobile board games but also in diverse real-world settings such as store shelves, building façades, and industrial layouts. The core objective is to develop a robust, general-purpose grid detection framework capable of identifying grid-like structures in various image domains, regardless of specific screen layouts, resolutions, or visual styles. While current efforts focus on mobile games — including match-3 puzzles and chess-like games— the envisioned solution remains agnostic to device types and game-specific designs, making it broadly applicable across domains.

Beyond grid detection, this research aims to integrate the vision system with an Answer Set Programming (ASP) [1] reasoning module. Following grid and object detection, symbolic information representing game states and object positions can be extracted and translated into ASP facts. Declarative logic, powered by ASP, enables the system to autonomously infer valid actions, plan strategies, and adapt gameplay, effectively allowing a robot to participate in visual games without human intervention. The goal would be to integrate these proposed techniques in BrainyBot [2]. This framework enables robots to play puzzle games by detecting repeated colored shapes and feeding that data into a symbolic reasoning module. The main limitation can be found in the usage of Template Matching to detect those shapes: this results in the tedious job of retrieving such templates to be looked for within the image.

---

*Joint Proceedings of the Workshops and Doctoral Consortium of the 41st International Conference on Logic Programming, September 9–13, 2025, Rende, Italy*

\*Corresponding author.

✉ tayyab.ateeq@unical.it (T. Ateeq)

ORCID 0009-0009-4180-2564 (T. Ateeq)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

Moreover, this approach is really resolution-dependent, requiring different templates for different resolutions of the same game. Enriching the pipeline with the *Grid detector* would relieve researchers from the burden of collecting said templates.

The synergy between machine vision and symbolic reasoning demonstrates significant potential for solving complex, real-world problems where structured visual patterns appear — from automated product counting on store shelves to enabling intelligent agents to interact with grid-based games. This combination leverages the strengths of both paradigms: computer vision for environmental perception and ASP for diagnostic reasoning, high-level control, and decision-making.

A selection of grid-like game scenarios is illustrated in Figure 1, highlighting the broad applicability of the proposed approach to diverse visual patterns.



**Figure 1:** Sample game images with grid arrangement of objects

## 2. Related Work

Considerable work has been done regarding object detection for real-world images, but there are only a few works that are focused on recognition in game scenes. Furthermore, no considerable work has been done to detect grid-like structures in games or other images. However, there are some works to detect repeated objects in images that partially align with our problem. Furthermore, the reviewed studies also cover a wide range of methodologies for object detection, pattern recognition, and repetitive structure analysis, applied to domains including computer games, printed fabrics, biological images, and urban environments.

Yoon et al. [3] proposed a model focused on an educational competition based on the Angry Birds game, where the objective was to design AI agents capable of playing both seen and unseen levels. Though the organizers provided a built-in API for detecting game elements like birds and objects, it had limitations in accuracy and scope. The paper emphasised student learning rather than technical novelty and did not delve deeply into the vision techniques or AI methods employed. In contrast, Ge et al. [4] applied qualitative stability analysis to detect unknown objects based on interactions between gravity and stability of known objects, primarily within Angry Birds. Techniques such as Canny edge detection and modified K-means clustering were used. Their results achieved a recall of 0.79 and precision of 0.68 and were also tested on Candy Crush and Super Stack games. Another work [5] proposed a hybrid approach of Canny edge detection, Contour Detection, and Circular Hough Transform (CHT) to deal with counting overlapped circular objects. This method overcame the problems of overlapping circular shapes normally present in cases, such as medical or biological imagery.

Liu et al. [6] developed a method for detecting near-regular textures (NRT) using Generalized PatchMatch (GPM) and Markov Random Fields (MRF). Their pipeline rectified image geometry via homography, identified repeating patches, and expanded the lattice structure iteratively. Tested on datasets from the 2013 Symmetry Detection competition and NRT images, they reached a detection accuracy of 91.1%, outperforming previous methods. Grant et al. [7] presented a lattice detection approach for automatic geotagging. They used SIFT features and RANSAC to detect repeated patterns in images of man-made environments and then matched them with a 3D image database. While the method didn't prioritise exhaustive lattice detection, it provided enough pattern recognition to support geolocation. Canada et al. [8] presented a method to detect 2D lattice structures in zebrafish larva arrays. Using image alignment, morphological closing, and symmetry analysis, they achieved 100% accuracy on 19 out of 20 test images, though the method required 5 minutes per image. Doubek et al. [9] designed a method for detecting multiple repetitive patterns, involving blob detection, and SIFT descriptors. A cluster-based approach, combined with Hough Transform and cross-correlation, allowed the detection and completion of partially visible patterns. Final tiles were transformed into a shift-invariant representation using the Fourier Transform, enabling image retrieval.

### 3. Vision-Based Agnostic Detection of Grid

My proposed method consists of a sequential pipeline of classical image processing and computer vision techniques. Different phases of the proposed method are depicted in the following section.

**Isolating the Main Game Area from the Screenshot:** Given an input image  $IM$ , which is a mobile game screenshot—such as one from Candy Crush Saga, Candy Fever, Chess, Checkers, or any other grid-based board game—the goal of the first processing stage is to isolate the main game region from the rest of the image, which may contain UI components and background elements. The image  $IM$  is first converted to grayscale to reduce complexity and eliminate color information, which is not relevant at this stage. This grayscale image is then smoothed using a Gaussian blur filter to reduce noise and make contour detection more robust. Next, adaptive thresholding is applied to generate a binary image, which highlights potential structures such as grid lines by separating foreground content from the background based on local intensity variations. Edge detection is then performed using the Canny method, helping to identify clear boundaries in the binary image. These edges are further refined using morphological operations: first dilated to close small gaps, then eroded to clean up noise. From the resulting image, external contours are extracted, and the largest contour  $C_{\max}$  is selected under the assumption that it corresponds to the visible game grid. A binary mask is then created from this contour and applied to the original image  $IM$ , producing a new image  $S^1$ , which contains only the region inside the main game grid, with the rest of the image set to black.

**Cropping Extrusions and Noise from the Bottom and Sides:** In the second part of the algorithm, the output image  $S^1$  is further processed to remove noise and irrelevant content near the horizontal and vertical borders of the extracted region. This step begins by analyzing the image row-wise to count the number of non-black pixels in each row, scanning both from the bottom up and from the top down. The algorithm checks whether each row contains a sufficient amount of meaningful content. If a row contains fewer non-black pixels than a specified threshold, it is considered background or noise and is replaced entirely with black pixels. This process is repeated iteratively until a row with a significant number of non-black pixels is found, indicating the presence of relevant information in that row and its consecutive neighbors. The same procedure is then applied column-wise to clean up the vertical edges. This function produces the output image  $S^2$ , which offers a cleaner and more focused representation of the game grid for the next analysis phase.

**Grid Construction:** After getting  $S^2$ , we now have to find the grid lines underlying the board. This is achieved by detecting horizontal and vertical lines ( $H, V$ ) using the probabilistic Hough Transform

and filtering them by angle and continuity. These lines are clustered according to the distance between them using a tolerance value. By analyzing the extracted grid the missing lines are inserted to fill gaps left by some undetected lines. Using the completed sets of lines  $(H, V)$ , the algorithm constructs the full grid by pairing intersecting lines to form rectangular boxes and producing  $S^3$ . Let  $B$  be the set of all the grid cells in the game board; each box  $b_i \in B$  is annotated with its center  $(x_i, y_i)$ , width  $w^i$ , and height  $h^i$ .

**False Positives Removal:** To eliminate empty or invalid boxes that may be black or irrelevant, each box in  $B$  is evaluated based on pixel intensity as depicted below. If the ratio of black pixels in a box exceeds a threshold, it is removed.

$$\mathcal{B}_{\text{valid}} = \{b \in \mathcal{B} \mid \rho_b < \rho_{\text{max}}\} \quad (1)$$

where  $\rho_b$  is the black pixel ratio in box  $b$ , and  $\rho_{\text{max}}$  is a predefined threshold.

**Grid Validation by Identifying Boxes with Size Discrepancies:** Next, the algorithm identifies boxes with inappropriate dimensions, which are either significantly larger or smaller than the average cell size. This step is primarily related to the auto-tuning part of the algorithm, which will be discussed in the subsequent section. The area  $A_i$  of each box is calculated, and boxes outside a certain range around the mean area  $\bar{A}$  are considered invalid as follows.

$$\bar{A} = \frac{1}{|\mathcal{B}_{\text{valid}}|} \sum_{i \in \mathcal{B}_{\text{valid}}} A_i, \quad \text{Invalid box} = \left\{ b_i \mid A_i > \bar{A} \cdot \alpha \text{ or } A_i < \frac{\bar{A}}{\alpha} \right\} \quad (2)$$

where  $\alpha$  is a scaling factor.

**Clustering of Similar Boxes:** For identifying similar objects within grid cells, template matching technique is used. A small template is extracted from each unassigned box and matched against other boxes using normalized cross-correlation. If a match exceeds the threshold  $T$ , the box is assigned to the current template's group. The total number of unique object types  $G$  is equal to the number of unique groups formed.

**Genetic Algorithm for Parameter Tuning:** The techniques discussed above rely on multiple parameters whose values vary across various games. As already stated, the aim of the research is to be as agnostic as possible: in order to enable parameters optimization for each game without previous knowledge of the game itself, a genetic algorithm is proposed. As the name suggests, it uses a genetic framework, a population-based search heuristic inspired by the principles of natural selection and evolution. The core objective is to tune parameters such as edge detection thresholds and line detection constraints that yield the best performance in grid construction and object grouping, based on an evaluation function. The algorithm begins by identifying the set of parameter names to be optimized and creating a bounded range for each parameter based on the lower and upper limits provided in the input. A population of candidate parameter sets, also called individuals, is then randomly initialized. Each individual is a vector representing one possible configuration. The discussed algorithm is executed with each of these configurations and each individual is assigned to some fitness evaluation. The fitness function relies on: the total number of detected objects or boxes, the number of distinct object types, the number of vertical and horizontal lines added or removed, and the number of inappropriate-sized boxes. The fitness of a candidate is computed primarily based on maximizing the ratio = object / object types while minimizing invalid boxes, added lines, and removed lines. These metrics are prioritized hierarchically to compare fitness between candidates. The selection method being used is tournament selection, where a random sample of individuals are compared with each other, and the best of each group are chosen as a parent for reproduction. From these selected parents, crossover is performed, swapping segments of two parent solutions to create new offspring. It keeps the best

solution found so far and keep track of how many generations have passed without improvement. After a given number of generations with no improvement, the loop terminates, implying that no parameters are being adjusted anymore. All generations are then scored using the best parameter set, and the one with the highest score is returned as a final result. Such auto-tuning allows for the system to be scalable, robust and adaptable to new game environments or image styles without the requirement of manual calibration.

## 4. Conclusion and Future Work

This research proposes a method to detect grid layouts in given images using computer vision methods. In the simplest terms, this is a visual perception layer in which the raw screenshots are converted into structured, machine-readable data. In the case of grid-based games, such as Candy Crush Saga, Chess, or Checkers, the structured output produced by our pipeline, such as a matrix of objects with associated types and coordinates, is fed into the next logical step of logic and reasoning in which possible moves in the game are predicted. This vision system, structured yet adaptable, connects passive recognition to active decisions, allowing a robot to play games on its own by perceiving, abstracting, planning, and acting on its own, based entirely on visual information. After implementing and performing the validation on an extensive dataset of board-based mobile games, we intend to extend the work to other domains such as recognition of grid arrangement of store shelf products, grid formed by repeated windows in buildings, printed grid patterns on fabrics and many more.

## Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

## References

- [1] T. Eiter, G. Ianni, T. Krennwallner, Answer set programming: A primer, in: Reasoning Web., 2009.
- [2] D. Angilica, M. Avolio, G. Beraldi, G. Ianni, F. Pacenza, From vision to execution: Enabling knowledge representation and reasoning in hybrid intelligent robots playing mobile games, in: KR, 2023, pp. 44–54.
- [3] D.-M. Yoon, K.-J. Kim, Challenges and opportunities in game artificial intelligence education using angry birds, *Ieee Access* 3 (2015) 793–804.
- [4] X. Ge, J. Renz, P. Zhang, Visual detection of unknown objects in video games using qualitative stability analysis, *IEEE Transactions on Computational Intelligence and AI in Games* 8 (2015) 166–177.
- [5] J. Ni, Z. Khan, S. Wang, K. Wang, S. K. Haider, Automatic detection and counting of circular shaped overlapped objects using circular hough transform and contour detection, in: 2016 12th World Congress on Intelligent Control and Automation (WCICA), IEEE, 2016, pp. 2902–2906.
- [6] S. Liu, T.-T. Ng, K. Sunkavalli, M. N. Do, E. Shechtman, N. Carr, Patchmatch-based automatic lattice detection for near-regular textures, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 181–189.
- [7] G. Schindler, P. Krishnamurthy, R. Lublinerman, Y. Liu, F. Dellaert, Detecting and matching repeated patterns for automatic geo-tagging in urban environments, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–7.
- [8] B. A. Canada, G. K. Thomas, K. C. Cheng, J. Z. Wang, Y. Liu, Automatic lattice detection in near-regular histology array images, in: 2008 15th IEEE International Conference on Image Processing, IEEE, 2008, pp. 1452–1455.
- [9] P. Doubek, J. Matas, M. Perdoch, O. Chum, Image matching and retrieval by repetitive patterns, in: 2010 20th international conference on pattern recognition, IEEE, 2010, pp. 3195–3198.