Pixel-Based Deep Reinforcement Learning for Flappy Bird

Gajendra Varma Penmetsa, Surya Tej Pattem

Research Question and Objective

RQ: *Can multi-scale visual features (Inception) improve sample-efficiency and final performance of a pixel-only RL agent in a sparse-reward game?*

Objectives

1. Implement baseline Atari-CNN + QR-DQN (pixel input).
2. Design "Inception-CNN" extractor (novel).
3. Train both for equal budget (2 M steps).
4. Compare on rich metric set (pipes, AUC, TTFP, stability).

Methodology

- **Environment:** Custom PyGame → Gymnasium (84×84 gray, 4‑frame stack, +0.2 living bonus).
- **Algorithms:** Distributional Double DQN (QRDQN) in Stable‑Baselines3.
- **Architectures:**
  • *Baseline* – 3‑conv Atari extractor.
  • *Inception* – Conv → Inception‑A block (1×1|3×3|5×5) → Conv.
- **Training hyper‑params:** lr 5e‑4, buf 50 k, batch 32, γ 0.99, ε anneal 1→0.1, 2 M steps, A100 GPU.

Key Metrics & Tools

Category	Metric	Source
Proficiency	Avg / Best pipes	50-episode eval
Efficiency	TTFP (time-to-first-pipe)	Custom callback
Stability	Loss σ (10 k window)	TensorBoard
Learning	Area-under-curve	TensorBoard scalar
Robustness	Avg episode length	Eval

Preliminary Results

Model	Avg pipes	Best	Avg len (frames)
Atari-CNN	5.6	22	299
Inception-CNN	13.2	61	606

Improvements

- Avg pipes **+135 %**
- Best episode **+177 %**
- Avg survival **+103 %**
- Time-to-first-pipe: 0.78 M vs 1.30 M steps (-40 %).

Challenges & Next Steps

Challenges

- Sparse +1 reward → long warm-up
- GPU quota management (Colab)

Planned Work

1. Curriculum (wide gap → normal) to push scores further.
2. Rainbow-DQN with Inception extractor.
3. Saliency visualisation & write-up.