Deep Grade
A visual approach to grading student programming assignments
Lisa Yan (yanlisa@stanford.edu), Nick McKeown (nickm@stanford.edu), Chris Piech (piech@cs.stanford.edu)

Problem Statement

Grading at scale
Computer: unit tests
Human: expensive

Graphics assignments
Easy for learning
Hard for grading

What if we treated grading graphics animations as a multi-class computer vision classification problem?

Functionality
(Deep Grade)

Style
(Human grader)

Video

Basic sampling group observed labels in window

Multiple occurrences of label

Deletion of objects

#2

Model Setup

Per video
Downsample:
- Over time
- Over image

Per rubric item
1. Aggregate embeddings over time
2. Predict P(incorrect)

Loss function
\[ \sum_{rubric} CE(y_{rubric}, y_{pred}) \]

Open questions

Temporal preprocessing
Take every 30th frame.
- IDT, traditional algorithms difficult, since objects disappear forever
- Simple window averaging techniques?
- Desired: Combine frames to encode movement and state

Skewed dataset
High false positives.
- Despite being synthetic, has very low appearance of incorrect rubric items.
- Desired: weighted loss function

Temporal embedding
Time invariant CNN.
- Recurrent neural network instead?
- Static rubric items are no longer true over time.

Advanced Breakout player
Single playthrough from a basic player.
- Tricky rubric items: Sticky Paddle
- Reinforcement learning: “Play to grade”
- Desired: Multiple playthroughs for full picture

Zero-shot learning: Synthetic Dataset

Breakout Player

Input:
automatic mouse

Output:
video of animation observed rubrics

Dataset generation
1. Generate Breakout code
2. Computer plays Breakout
3. Record animation frames + rubrics triggered

Students
Real: 700 / 3 weeks
Synthetic: 10/8 / immediately

Breakout Grammar
Expert-constructed:
- Code snippet ↔ Rubric tag
- Control flow between snippets

Results

Ongoing results
Temporal rubric items: multi-frame dependent

Train set:
10k one-hot
40k general
Low precision.
Skewed dataset.

References