Real-Time Trading Card Recognition in Live Video

Kevin Culberg (kculberg@Stanford.edu)

Predicting

Tournaments for physical card games such as Magic: The Gathering attract thousands of online viewers. Despite this success, tournament livestreams are not easily accessible for newer players who have not memorized the many unique cards. I propose an object detection model based on the popular "You Only Look Once" (YOLO) model to detect cards from real time video of tournaments to improve viewer understanding.

Data

Dataset consists of hand-labeled full-color images with resolution 1920x1080 captured from live tournaments. Additional images were generated by artificially inserting cards into empty or partially empty frames.

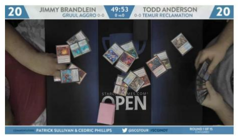
- Train: 20,000 images (62 real + 19,938 augmented)
- Dev: 120 real images
- Test: 120 real images

Labels are in YOLO format with bounding box information:

- Ex. <class_id> <x> <y> <width> <height>
- Three versions of the dataset were created with either 264, 51, or 1 unique class ids
- A single image can have between 0 and 20+ cards

Examples:

Hand-Labeled Image



Augmented Image

Features

Input Features:

- Resized Image: 608 x 608 x 3 pixels

Output Features:

- Feature Map: 38 x 38 x (5 + C), C = # unique classes
- Feature vector:
  - [x, y, width, height, p_obj, p_0, …, p_C]
    - x, y = object center x and y coordinate
    - width, height = size of the detected object
    - p_obj = probability this is an object
    - p_c = probability object is class c

Feature map is translated into list of detections by taking only those that meet: p_obj > threshold

Model

Model Structure:

	Start			Region 1		Region 2		Region 3	
Type	Conv2d	Conv2d	Res x1	Conv2d	Res x2	Conv2d	Res x8	Conv2d	Res x8
Filters	32	64	32/64	128	64/128	256	128/256	512	256/512
Size	3	3		3		3		3	
Stride	1	2		2		2		2	

	Head						Output
Type	Conv2d	Conv2d	Conv2d	Conv2d	Conv2d	Conv2d	Conv2d
Filters	256	512	256	512	256	512	5+C
Size	1	3	1	3	1	3	1
Stride	1	1	1	1	1	1	1

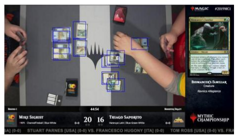
Loss function combines:

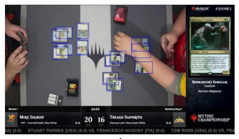
- Coord loss
  - Mean Squared
- Size loss
  - Mean Squared
- Objectness loss
  - Binary Cross Entopy
- Class loss
  - Softmax Cross Entropy

$$
\begin{aligned}
L = & \frac{\lambda_{\text{coord}}}{S^2} \sum_{i=0}^{S} \sum_{j=0}^{S} \mathbb{1}_{ij}^{\text{obj}} ((x_{ij} - \hat{x}_{ij})^2 + (y_{ij} - \hat{y}_{ij})^2) \\
& + \frac{\lambda_{\text{size}}}{S^2} \sum_{i=0}^{S} \sum_{j=0}^{S} \mathbb{1}_{ij}^{\text{obj}} ((w_{ij} - \hat{w}_{ij})^2 + (h_{ij} - \hat{h}_{ij})^2) \\
& - \frac{1}{S^2} \sum_{i=0}^{S} \sum_{j=0}^{S} \mathbb{1}_{ij}^{\text{obj}} d_{ij} \log \hat{d}_{ij} \\
& - \frac{\lambda_{\text{noobj}}}{S^2} \sum_{i=0}^{S} \sum_{j=0}^{S} \mathbb{1}_{ij}^{\text{noobj}} (1 - d_{ij}) \log (1 - \hat{d}_{ij}) \\
& - \frac{1}{S^2} \sum_{i=0}^{S} \sum_{j=0}^{S} \mathbb{1}_{ij}^{\text{obj}} \sum_{c \in \text{classes}} p_{ij}(c) \log \hat{p}_{ij}(c)
\end{aligned}
$$

Results

	Mean Average Precision (mAP)			Speed (ms)
	Data 264	Data 51	Data 1	
me	**5.31%**	**6.78%**	62.09%	**57.1**
YOLO	0.10%	2.20%	**86.08%**	67.4


Me / Data1


YOLO / Data1

Discussion

Detection speed for both models fast enough for live video. Performance of both models suffers greatly as the number of unique classes increases with my model being slightly better. YOLO outperforms my model when only a single class is present with much tighter bounding boxes. Adding classes requires significantly more training time and does not scale well. Model architecture is reasonable for bounding box detection, but alternate methods would be best for card id prediction.

Future

A good next step would be to adapt the current model architecture to act as a first stage in a larger pipeline for card recognition. By applying techniques from facial recognition it may be possible to scale the number of unique card classes to the thousands and combine that with the bounding box prediction from this model. The current running time of the model is fast enough that adding a second stage would still result in acceptable time.