PlayCDC - Playing Card Detection

Learning to detect suits and ranks of playing cards

Daniel Gonzalez & Frank Gabel
Ruprecht-Karls-Universität Heidelberg

Contact Information:
Daniel Gonzalez
Email: d.gonzalez@stud.uni-heidelberg.de

Frank Gabel
Email: ey143@uni-heidelberg.de

Abstract

With the capabilities of upcoming small video capturing devices in, for example, smart contact lenses with built-in camerass, whole new ways of cheating in certain cardgames emerge. In order to help facilitate these cheating endeavours, we implement an algorithm that detects the suits and ranks of playing cards in the field of view of a camera using the latest iteration of the YOLO object recognition algorithm.

Introduction

Main Objectives

The objectives of this project are summarized as follows:

Create a general dataset of a standard, 52-card deck of playing cards in different poses, brightness situations and blurring levels annotated with bounding boxes around the ranks and suits and corresponding class information.

Train an object detection algorithm on these synthesized data that performs bounding box localization and regression for classification. In particular, we train the latest iteration of the YOLO object detection algorithm [RF18] end-to-end.

Evaluate the algorithm on a hold-out validation dataset covering all classes. As a performance metric, mean Average Precision (mAP) is used.

Deploy the model on a smartphone camera as a proof of concept.

Methods

The dataset creation pipeline

The dataset was created in mainly 2 big steps.

Data Preparation: We took two photos for each card of a deck of 52 cards, manually crop the cards to a selection and rescale the result by 900×600 pixels using GIMP. After this, we detect the convex hulls of the cards suits and ranks using the SciPy python-library.

Data Generation: First, we paste the images in canvances provided by DTD [CMK+14], in order to have different textures arround the images. Then, we apply to each image linear transfomations, as well as blurring and sharping to generate a considerable amount of data for training. The latest step was performed using the imgaug python-library in order to keep track of the convex hulls.
We generate in total 450 training images for each card.

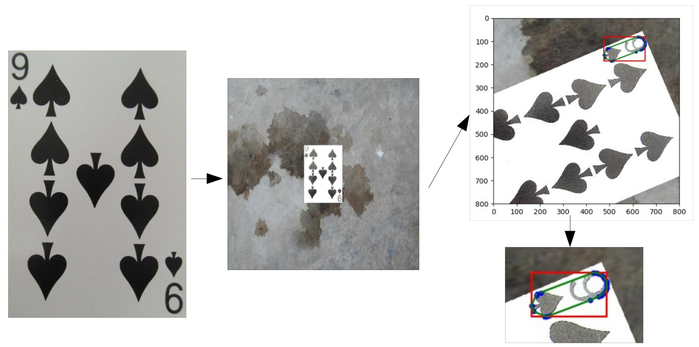

Figure 1: The dataset creation pipeline: paste photographs of images onto textures and apply random transformations to both images and corresponding convex hulls/bounding boxes

The YOLO approach to object detection

The general steps can be subsumed as follows:

Predict candidate bounding boxes

Class Prediction

Predictions Across Scales

Training using the Darknet-53 feature extractor

tiny-YOLO-v3

The particular architecture we used for training on our dataset is **tiny YOLOv3** which essentially is a smaller version of YOLO for constrained environments.

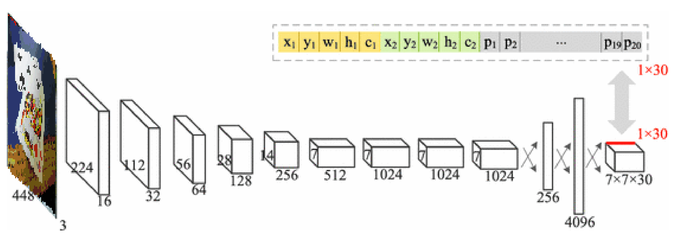

Figure 2: Figure caption

Evaluation strategy

For the evaluation, we use the Mean Average Precision (mAP).
We determinate True Positives and True Negatives using a Threshold of 0.5, i.e if $IoU > 0.5$ or $IoU < 0.5$, and taking together the False Negatives, we compute the Precision x Recall curve [see Figures]. The overall mean average precision was of 95.10%.

Vestibulum ac diam a odio tempus congue. Vivamus id enim nisi:

$$\cos\bar{\phi}_k Q_{j,k+1,t} + Q_{j,k+1,x} + \frac{\sin^2\bar{\phi}_k}{T\cos\bar{\phi}_k} Q_{j,k+1} = -\cos\phi_k Q_{j,k,t} + Q_{j,k,x} - \frac{\sin^2\phi_k}{T\cos\phi_k} Q_{j,k} \quad (1)$$

Results

sadddddddddddddddddddddddddddddddddddddddsdasadasdsfsadafadsfadsfdsfadsfdsafsadfdsafvc
example, smart contact lenses with built-in camerass, whole new ways of cheating in certain cardgames emerge. In order to help facilitate these cheating endeavours, we implement an algorithm that detects the suits and ranks of playing cards in the field of view of a camera using the latest iteration of the YOLO object recognition algorithm.example, smart contact lenses with built-in camerass, whole new ways of cheating in certain cardgames emerge.
sadddddddddddddddddddddddddddddddddddddddd
In order to help facilitate these cheating endeavours, we implement an algorithm that detects the suits and ranks of playing cards in the field of view of a camera using the latest iteration of the YOLO object recognition algorithm.

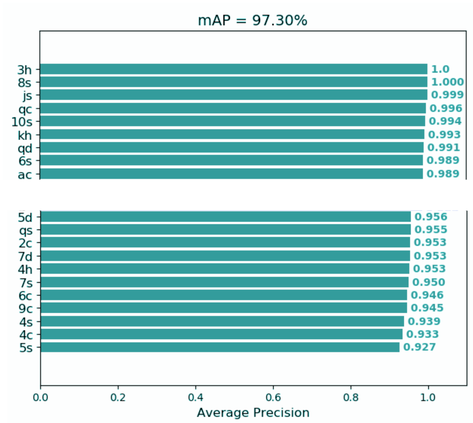

Figure 3: A gull

Conclusions

- Using an artificially created dataset, we achieve a mAP score of 95.10% on a holdout dataset.
- The task of object detection on ranks/suits of playing

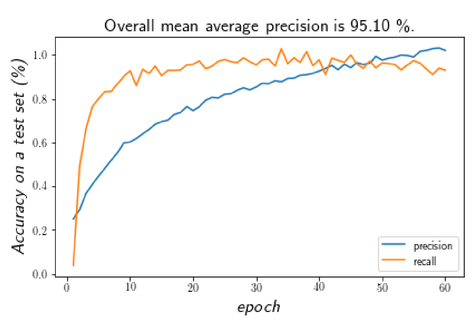

Figure 4: A gull

cards appears to be rather easy - it can be thought of being 2D rather than 3D.

- Deploying the model on a webcam results in 180 FPS on a 480x480 resolution, which is state-of-the-art fast.

References

[CMK+14] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[RF18] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.

Acknowledgements

Etiam fermentum, arcu ut gravida fringilla, dolor arcu laoreet justo, ut imperdiet urna arcu a arcu. Donec nec ante a dui tempus consectetur. Cras nisi turpis, dapibus sit amet mattis sed, laoreet.