# 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

A big topic in computer vision is the search of methods that are, given a query image, capable of answering questions like: What is present in an image? Is there a particular object in it? Where exactly in the image is this object located? Object detection deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Typically, object detection deals with two sub-taks: **object localization** using bounding boxes and **multiclass object classification** within said bounding boxes. In this project, we implement the YOLOv3 object detection algorithm.

## 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.
First, we took two photos for each card in a deck of 52 cards, crop the cards to a selection and rescale the result by 900×600 pixels. This was done manually using the selection/rotation/cropping/rescale-tools provided by GIMP.

Concluding this, we manage to detect the convex hulls of the cards suits and ranks using the SciPy library.
The next step was to generate a big amount of data for each card applying linear transfomations, as well as blurring and sharping on the image. First we paste the images in canvances using the Describable Textures Dataset [?], in order to have different textures arround the image. For the next, we use the imgaug library in order to perform the mentioned transformations, keeping track of the convex hulls.
For each of the 52 cards of the deck, we generate in total 450 training images.

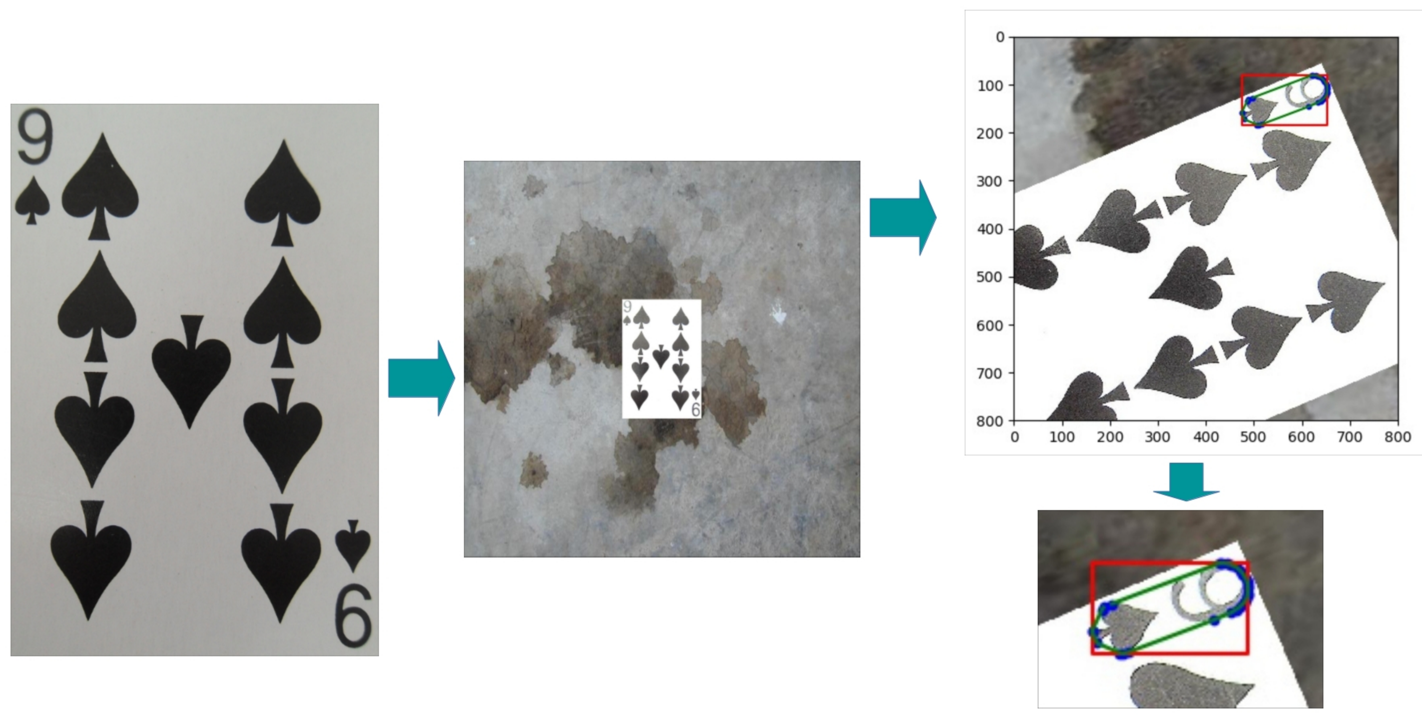

**Figure 1:** The dataset creation pipeline: paste photographs of images onto textures and apply random transformations to both images and corresponding convex hulls/bounding boxes

### The YOLO approach to object detection

Prior detection systems repurpose classifiers or localizers to perform detection. They apply the model to an image at multiple locations and scales. High scoring regions of the image are considered detections.
In YOLO, the approach is totally different in that a single neural network is applied to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. 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.

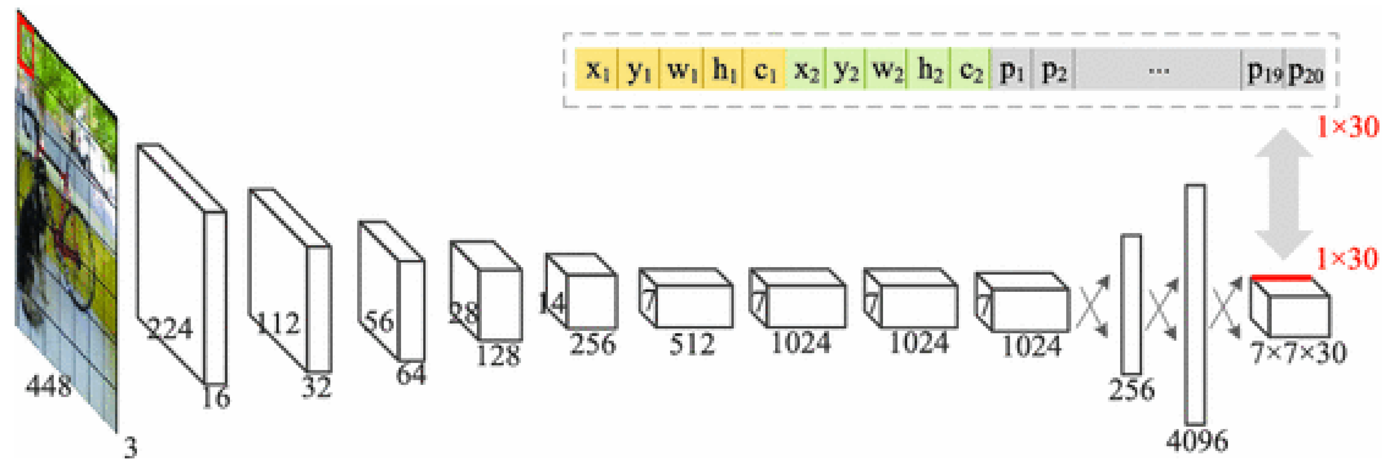

**Figure 2:** Figure caption

### Evaluation strategy

$$E = mc^2 \tag{1}$$

Curabitur mi sem, pulvinar quis aliquam rutrum. (1) edf (2) , $\Omega = [-1, 1]^3$, maecenas leo est, ornare at. $z = -1$ edf $z = 1$ sed interdum felis dapibus sem. $x$ set $y$ ytruem. Turpis $j$ amet accumsan enim $y$-lacina; ref $k$-viverra nec porttitor $x$-lacina.
Vestibulum ac diam a odio tempus congue. Vivamus id enim nisi:

$$\begin{aligned}\cos\bar{\phi}_k Q_{j,k+1,t} + Q_{j,k+1,x} + \frac{\sin^2\bar{\phi}_k}{T\cos\bar{\phi}_k} Q_{j,k+1} = \\ -\cos\phi_k Q_{j,k,t} + Q_{j,k,x} - \frac{\sin^2\phi_k}{T\cos\phi_k} Q_{j,k}\end{aligned} \tag{2}$$

and

$$\begin{aligned}\cos\bar{\phi}_j Q_{j+1,k,t} + Q_{j+1,k,y} + \frac{\sin^2\bar{\phi}_j}{T\cos\bar{\phi}_j} Q_{j+1,k} = \\ -\cos\phi_j Q_{j,k,t} + Q_{j,k,y} - \frac{\sin^2\phi_j}{T\cos\phi_j} Q_{j,k}.\end{aligned} \tag{3}$$

## Results

Donec faucibus purus at tortor egestas eu fermentum dolor facilisis. Maecenas tempor dui eu neque fringilla rutrum. Mauris *lobortis* nisl accumsan. Aenean vitae risus ante. Pellentesque condimentum dui. Etiam sagittis purus non tellus tempor volutpat. Donec et dui non massa tristique adipiscing.

| Treatments | Response 1 | Response 2 |
|---|---|---|
| Treatment 1 | 0.0003262 | 0.562 |
| Treatment 2 | 0.0015681 | 0.910 |
| Treatment 3 | 0.0009271 | 0.296 |

**Table 1:** Table caption

Phasellus imperdiet, tortor vitae congue bibendum, felis enim sagittis lorem, et volutpat ante orci sagittis mi. Morbi rutrum laoreet semper. Morbi accumsan enim nec tortor consectetur non commodo nisi sollicitudin. Proin sollicitudin. Pellentesque eget orci eros. Fusce ultricies, tellus et pellentesque fringilla, ante massa luctus libero, quis tristique purus urna nec nibh.

Nulla ut porttitor enim. Suspendisse venenatis dui eget eros gravida tempor. Mauris feugiat elit et augue placerat ultrices. Morbi accumsan enim nec tortor consectetur non commodo. Pellentesque condimentum dui. Etiam sagittis purus non tellus tempor volutpat. Donec et dui non massa tristique adipiscing. Quisque vestibulum eros eu. Phasellus imperdiet, tortor vitae congue bibendum, felis enim sagittis lorem, et volutpat ante orci sagittis mi. Morbi rutrum laoreet semper. Morbi accumsan enim nec tortor consectetur non commodo nisi sollicitudin.

In hac habitasse platea dictumst. Etiam placerat, risus ac.

Adipiscing lectus in magna blandit:

| Treatments | Response 1 | Response 2 |
|---|---|---|
| Treatment 1 | 0.0003262 | 0.562 |
| Treatment 2 | 0.0015681 | 0.910 |
| Treatment 3 | 0.0009271 | 0.296 |

**Table 2:** Table caption

Vivamus sed nibh ac metus tristique tristique a vitae ante. Sed lobortis mi ut arcu fringilla et adipiscing ligula rutrum. Aenean turpis velit, placerat eget tincidunt nec, ornare in nisl. In placerat.

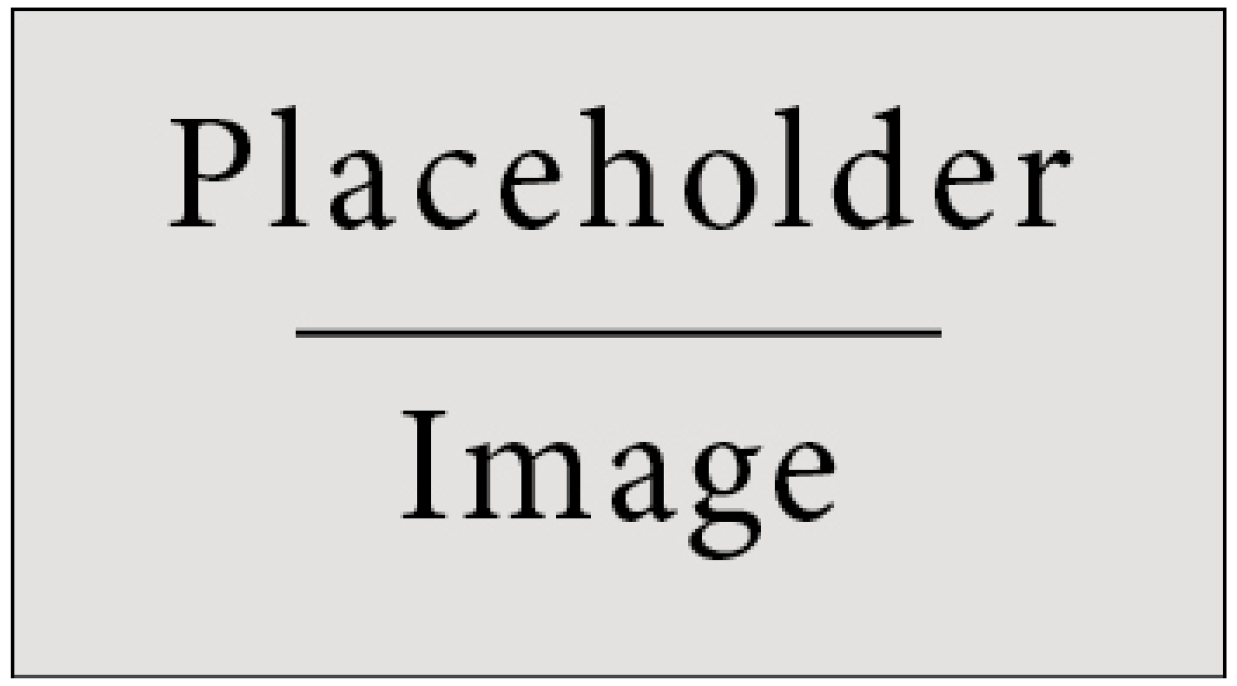


**Figure 3:** Figure caption

## 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 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.

## Forthcoming Research

Vivamus molestie, risus tempor vehicula mattis, libero arcu volutpat purus, sed blandit sem nibh eget turpis. Maecenas rutrum dui blandit lorem vulputate gravida. Praesent venenatis mi vel lorem tempor at varius diam sagittis. Nam eu leo id turpis interdum luctus a sed augue. Nam tellus.

## 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.