Bridge bidding via playing card detection using YOLO v3

Philip Xu
Brown University
69 Brown St, Providence RI
philipxjm1@gmail.com

Conant Raj Kumar
Brown University
69 Brown St, Providence RI
conantrkumar@gmail.com

Grace Bramley-Simmons
Brown University
69 Brown St, Providence RI
grace_bramley-simmons@gmail.com

Faculty Sponsor: James Tompkin
Brown University
69 Brown St, Providence RI
james_tompkin@brown.edu

Abstract

This project aimed to construct a mobile application capable of providing the user with the correct opening bridge bid after the user takes a photo of their hand. We used YOLO (You Only Look Once) object detection algorithm to perform the detection task. We generated our own training data set by using a custom script that combines playing card images with random background textures.

1. Problem Statement

We want to achieve precise detection of bridge hands (13 unique playing cards from a deck) from an image of the hand and accurately suggest the best bridge bid given the detections.

2. Related Works

Our approach builds off of YOLO Darknet object detection algorithm, which utilises a single neural network which predicts bounding boxes and class probabilities directly from full images. [1]

3. Data Generation

The steps followed for data generation are outlined here https://github.com/geaxgx/playing-card-detection/blob/master/creating_playing_cards_dataset.ipynb

3.1. Measured Data

The specific measurements of the card deck used were required to generate the dataset. The height and width of the card, as well as the height and width of the largest box containing the suit and value combination on the corners of the cards were required to produce usable images for test and training data. In our case, the card height and width were 88mm and 62mm respectively. The corner box started from 1.5mm and ended at 9mm in width, and 3mm - 22mm in height.

3.2. Card Image Generation

A video was taken of each of the 52 possible playing cards in the deck on a consistent green background with varied brightness. Frames of the video were then chosen based on the blurriness of the image to obtain the individual card photos.

3.3. Convex Hull Extraction

Once the cards have been extracted, a more specific hull is defined for each card to be more easily identified. Each card has a different hull according to the shape of the value and suit.

3.4. Scene Generation

Two types of scenes were used in the data set. The first type of scene is of two cards of varied size and rotation on a random background. The second type is of 3 fanned cards of varied rotation on a random background. This allows the model to identify cards which are almost entirely hidden behind other cards. 30,000 3 cards scenes were used in the training data, as well as 20,000 2 card scenes. 5,000 2 card scenes and 5,000 3 card scenes were used in the testing data.

Figure 1. Example Scene

4. Detection Algorithm

The steps followed for training YOLO Darknet are outlined in this git repository: `https://github.com/AlexeyAB/darknet`

4.1. Training Details

We used the following configurations for training our model:

1. Perform transfer learning with pre-trained weights: darknet53.conv.74
2. Hyperparameter modifications:
   (a) batch = 64
   (b) subdivision = 16
   (c) width = 608, height = 608
   (d) flip = 0
   (e) learning rate = 0.001 (0.0001) after 2000 iterations.
   (f) classes = 52 (52 playing cards)
3. Training / Testing split of 80/20.

After 5000 iterations, we reach mean average precision (mAP) of 99.98 percent. Training took 6 hours on RTX 2080.

4.2. Training Results

See figure 2.

4.3. Results Discussion

Mistakes are occasionally made by model when hand is not fanned enough or placed in front of a complex background. This is likely due to our training data being artificially generated, so cards are generally relatively fanned out and the backgrounds are relatively uniform.

Some mistakes are made between distinguishing similar looking classes such as Aces with Fours, and Spades with Clubs; we believe that this can be alleviated by generating more training data from a more diverse set of card decks and more videos.

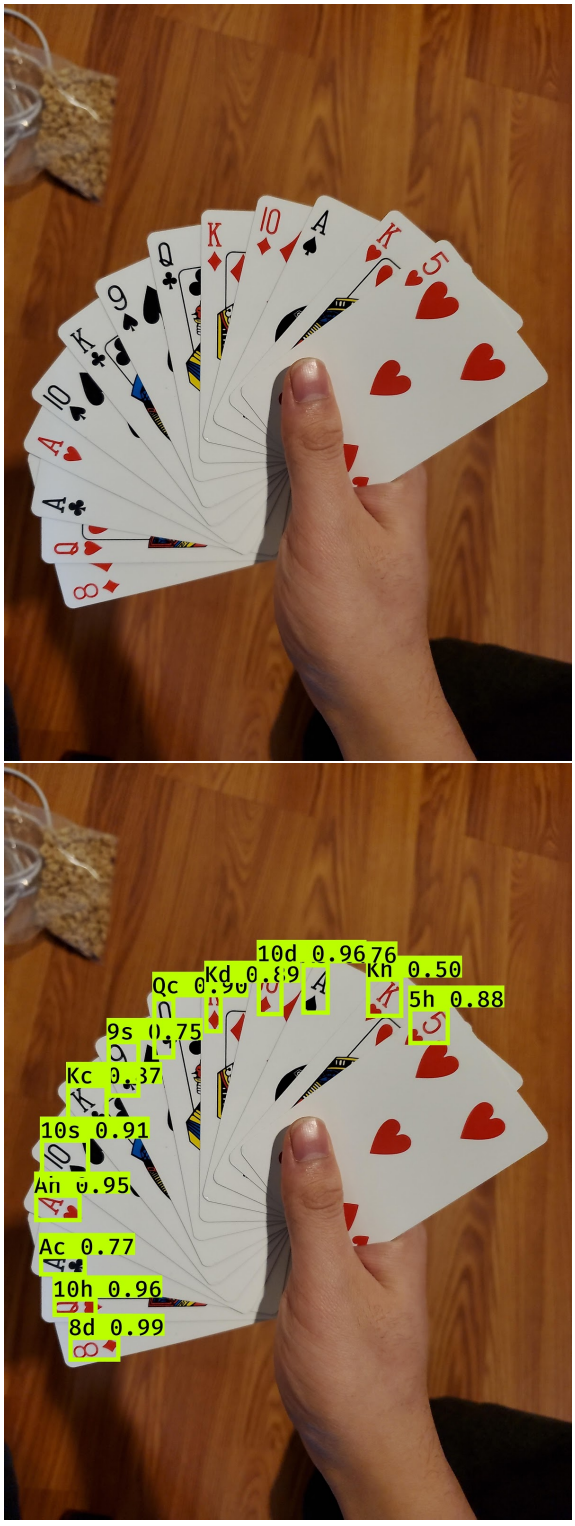

Figure 2. Example Hand Detection

5. Bidding Algorithm

The bidding algorithm was written with the Standard American Yellow Card convention opening bids in mind. These bids are based on values related to the number of ”points” in each hand determined by the number of face cards and values of those face cards. The bid is also determined by suit distribution.

6. Conclusion

This project successfully completed its goal of creating an app which can give the user an accurate opening bridge bid based on a photo of their hand. This app is useful to users who are learning the game and would like to check their intuition with the correct bid. In the future we may extend it to more bids than just opening bids.

7. Appendix

1. Raj was responsible for generating the dataset and bridge algorithm.

2. Philip was responsible for model training and integration of model into backend server.

3. Grace was responsible for developing a mobile application using React Native and a backend server using Flask that allows a user to take a photo of a hand of cards, and receive a summary of the cards in their hand and a recommended bridge bid.

4. The videos used for scene generation as well as the tens of thousands of scenes generated are too large to be included in the github repository but a link to a google drive containing them is given here. The training weights used in the YOLO model are also stored here. `https://drive.google.com/drive/u/0/folders/1d2eO6PkvG82aVSon6ONEgAEqyYfAuD38`

References

[1] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.