Playing Cards and Bidding Calls Detection For Automatic Registration of a Duplicate Bridge Game

Piotr Wzorek[1[0000-0003-3885-600X]] and Tomasz Kryjak[1[0000-0001-6798-4444]]

Embedded Vision Systems Group, Computer Vision Laboratory,
Department of Automatic Control and Robotics,
AGH University of Science and Technology, Krakow, Poland
{pwzorek,tomasz.kryjak}@agh.edu.pl

Abstract. In this work, the implementation of a playing cards and bidding calls detection system for the automatic registration of a duplicate bridge game is presented. For this purpose, two YOLOv4 deep convolutional neural networks were used. During the training process, which was carried out using an automatically generated dataset, we obtained a detector characterised by more than 99.9% accuracy on the test set. The prepared system has been evaluated on videos collected during bridge competitions and is characterised by high accuracy. The application has been implemented and evaluated on a CPU and an embedded GPU.

Keywords: Deep neural networks · YOLOv4 · Object detection · Duplicate bridge · embedded GPU

1 Introduction

Duplicate bridge is a logic game played with 52 classical cards. It involves 4 players in two pairs and consists of two consecutive phases: auction and play. During the first of these, the players communicate using the so-called bidding calls. In the second phase, the players play one card each in turn (Figure 1).

During the most prestigious competitions, live broadcasts are performed manually by operators. This is a low-complex and repetitive task, so eliminating human involvement can reduce the broadcasting costs and have a positive impact on its accuracy. Therefore, in this paper, we propose an embedded vision system that can be used to automate this process. For this purpose, we used a solution based on the YOLOv4 deep neural network. To be applied in practise, the solution should enable live transmission, that is, process and analyse the video stream in real time, possibly on an energy efficient and compact platform. With regard to that, we have implemented the detector on an embedded GPU (NVIDIA AGX Xavier).

The main contribution of this paper is the proof that the transmission of bridge games can be automated with the use of modern computer vision algorithms. To the best of our knowledge, this is the first scientific publication on this matter.

Fig. 1: Examples of auction (left) and the play phase (right).

The remainder of this paper is organised as follows. In Section 2 the implementation of cards and bid calls detection is described. Optimisation of the detector for an embedded GPU is explained in Section 3. The evaluation of the solution is described in Section 4. The publication ends with a summary and plans for further work on the application.

2 Cards and Bidding Calls Detection

The system should recognise 52 different cards and 38 unique bidding calls. There have been several classical playing card detection methods published so far. The paper [1] from 2010 proposed a system based on a "classical" approach – the Hough transform was used to locate objects. Classification of the proposed areas was carried out using probabilistic methods to compare a set of symbols based on the orientation of their edges or pattern matching and computing correlations.

Later work described solutions based partially or fully on Deep Neural Networks (DNNs). [2] proposes a solution based on the *EfficentDet* network [3]. In the problem of cards detection during poker games considered, the authors' motivation to choose this architecture was the size of the objects relative to the size of the analysed image (0.7% of the dimension of the image).

In this paper, we have decided to use two YOLOv4 networks [4] – one for card detection, the other for bidding calls. After an initial analysis of the problem, we have selected this architecture, as it can be considered as state-of-the-art due to its detection performance (analogous to the possible alternatives), while characterised by a short inference time. The use of the two networks does not have a negative impact on the speed of the application, because at any stage of the game it is not necessary to recognise these elements at the same time.

The database of cards and bidding calls used during the training process was automatically generated and includes cases specific to the game of bridge – we have described the process in a separate publication [5]. We have generated 39300 images of cards and 30000 images of bidding calls using the described method.

We conducted the training process of the YOLOv4 network on 4 *Tesla V100 GPUs* with an input size of $640x640x3$ and a batch size of 64. For the YOLOv4

Table 1: Achieved detection accuracy and throughput for NVIDIA AGX Xavier.

Used model	accuracy [mAP@0.5]	throughput [fps]
YOLOv4 FP32 - cards	99.96%	9
YOLOv4 TensorTR FP16 - cards	99.3%	20
YOLOv4 TensorTR INT8 - cards	97.2%	28
YOLOv4 FP32 - bidding calls	99.98%	9
YOLOv4 TensorTR FP16 - bidding calls	99.6%	20
YOLOv4 TensorTR INT8 - bidding calls	99.3%	28

model, the Mish activation function, the $CIoU$ loss function, and regularisation using DropBlok and Class label smoothing mechanisms were selected. Furthermore, MOSAIC augmentation was used for the input images from the training dataset. In this research, we used the Prometheus computing cluster available in the *Academic Computer Centre CYFRONET AGH*.

We applied the transfer learning technique by using network weights adapted to the detection of objects from the *MS COCO* [6] dataset. For both detectors, we obtained high detection performance – 99.96% mAP@0.5 for card recognition and 99.98% mAP@0.5 for bidding calls recognition.

The obtained playing cards detector has an accuracy higher than the alternative proposals in the literature. For example, the solution based on the *EfficientDet-S* network described in [2] has a detection efficiency of 96.9% mAP@0.5 on the test set. However, significant differences between the case considered in the aforementioned publication and the one described should be taken into account – the network is adapted to perform card detection on other datasets.

3 Implementation on Embedded GPU

In order for the detector to be used for real-time game transmission, the trained models were optimised for the embedded GPU using TensorRT tools [7]. For both networks, 16 bit floating-point (FP16) and 8 bit integer (INT8) representations were prepared. Each model was evaluated on a test dataset (Table 1). As the precision of the weight representation in the network model is reduced, a significant decrease in inference time can be observed with only a slight decrease in precision.

4 Towards Automatic Registration of a Duplicate Bridge Game

To evaluate the prepared detector, we recorded a video during a bridge competition. It involved 15 bidding calls (8 of them unique) and 52 cards. We ran three tests. In the first, each frame was subjected to detection using YOLOv4 without optimisation. All bidding calls were correctly identified. However, the correct estimate of the number of reported passes by the same player proved to be troublesome. This problem was solved by assuming that if a call other than a pass was not recognised in that players' area – that the player passed. The FP16 and INT8 versions also allowed to recognise all bidding calls correctly.

When analysing the play phase, we have obtained the following results:

- FP32 – 46 out of 52 detected correctly (88.5%),
- FP16 – 45 out of 52 detected correctly (86.5%),
- INT8 – 14 out of 52 detected correctly (26.9%).

They indicate that even a very good detector (97-99% accuracy), trained on a large and varied database, does not necessarily perform as reliably in real conditions. This is particularly noticeable with the INT8 version.

5 Summary

The developed playing cards and bidding calls detection system is characterised by high efficiency. The use of an embedded GPU with an optimised FP16 model enables to obtain a real-time solution. This is possible due to the relatively slow pace of the bridge play. Applying some modifications could make it possible to use the system for preparing live broadcasts of bridge competitions.

We plan to continue the development of the system. As part of further work, we intend to thoroughly analyse the detection system, update the training dataset with "problematic" cases and continue the training process, which could significantly improve the quality of detection of elements used during the game. Additionally, supporting the recognition process with game rules analysis should improve the effectiveness of the solution. For example, the obligation to follow suit could be taken into account when recognising cards.

Acknowledgement. The work presented in this paper was supported by the AGH University of Science and Technology project no. 16.16.120.773.

References

1. Pimentel, J. and Bernardino, A.: A Comparison of Methods for Detection and Recognitionof Playing Cards"
2. Chen, Q.: Poker Watcher: Playing Card Detection Based on EfficientDet and SandglassBlock, 2020 11th International Conference on Awareness Science and Technology (iCAST).IEEE. 2020, s. 1–6
3. Tan,Pang R. and Le, Q. V.: EfficientDet: Scalable and Efficient Object Detection, 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10778-10787, doi: 10.1109/CVPR42600.2020.01079.
4. Bochkovskiy, A., Wang, C. and Liao, H.M.: YOLOv4: Optimal Speed andAccuracy of Object Detection, CoRRabs/2004.10934 (2020). arXiv:2004.1093
5. Wzorek, P. and Kryjak, T.: Training dataset generation for bridge game registration, Zeszyty Studenckiego Towarzystwa Naukowego, ISSN 1732-0925. arXiv:2109.11861
6. Lin, T.: Microsoft COCO: Common Objects in Context, CoRR abs/1405.0312(2014). arXiv:1405.031
7. The TensorRT SDK, https://developer.nvidia.com/tensorrt, Last accessed 7 March 2022