Soccer Player Recognition using Artificial Intelligence and Computer Vision

Charles-Alexandre Diop, Baptiste Pelloux, Xinrui Yu, Won-Jae Yi and Jafar Saniie
Embedded Computing and Signal Processing Research Laboratory (http://ecasp.ece.iit.edu/)
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago, IL, U.S.A.

***Abstract*— Groundbreaking features and functionalities are available in sports broadcasting programs, specifically in soccer games, such as post-game analysis, tracking of players, tracking of the ball, and associated teams' and players' statistical information. However, viewers don't have control to choose which features to view on-demand and aren't able to interact with any of these functionalities. Our system aims for the fundamental system of an on-demand viewer-driven application that would be able to track players on the field, identify formation changes, follow team strategy changes, and gather all statistical information of the player and the team on a single screen. In order to realize such an application, we focus on developing a method to track players on the field using multiple Artificial Intelligence (AI) and computer vision techniques where our application employs facial recognition and jersey number recognition algorithms. As a live soccer game broadcast would have various camera views of the players, our custom-made database contains captured images from different scenarios of camera views such as different angles and zooms of the player and is used to train the model using Convolutional Neural Network (CNN). Our system is scalable to different types of sports, such as basketball, baseball, volleyball, and more where players have their numbers on their jerseys and would be feasible to apply our system since soccer involves more players on the field than other sports games.**

I. Introduction

The notion of sports is as old as human history. Even before the era of live or recorded broadcasting sports events, sports games were an activity that had the power to create unity among people. In 1936, Berlin Olympics was the first televised sporting event soon after the invention of the television [1]. The method of sports games broadcasting has evolved with interactive feedback to the viewers to provide more information related to the game, players involved, and the team associated with the game and/or the league. In the recent advancement of Artificial Intelligence (AI) and machine learning coupled with Big Data, interactive and real-time experience of learning statistics and predictions of possible movements by the players have made users more engaged in the games [2]. However, most of these new features to the sports broadcasting programs lack direct interaction with the viewers, they should be able to select the information that needs to be displayed on the screen. In this paper, in the effort to realize such a system, we explore methods of tracking players on the field, specifically in soccer games, by identifying their faces and jersey numbers.

II. System Design

Our system design is divided into two main parts: detection and recognition. Detection is involved with the players on the field, the players' faces, and jersey numbers on the players. Our system design will be utilizing YOLOv3 [3] to detect players on the field, use a customized face dataset and recognize the players' faces using the Dlib toolkit [4], and detect and recognize players' jersey numbers by MNIST [5] and Kaggle datasets [6] coupled with Convolutional Neural Network (CNN) algorithm. The overview of the system flowchart is shown in Fig. 1. In this section, we will describe the detection algorithms first where the recognition algorithms will be utilizing the results obtained from the detection algorithms.

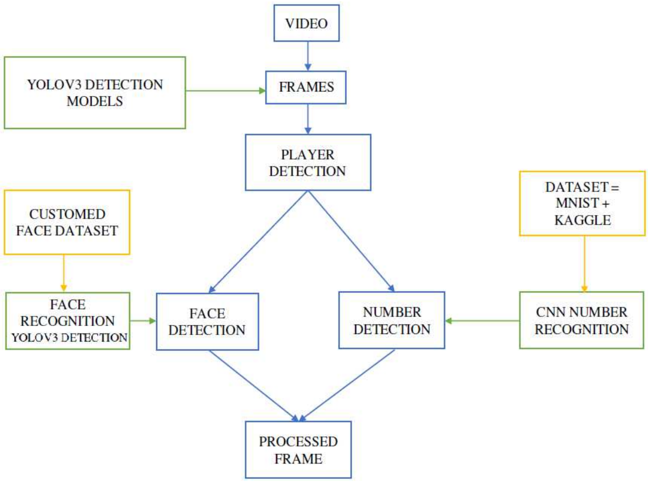


Fig. 1. Overview of the System Design

A. *Detection Algorithms*

The first step required to achieve the goal of this study is to detect players on the field before proceeding to recognition mechanisms. As soccer games are dynamic in terms of players' movement on the field, there are many cameras installed in the stadium to observe the play from various angles and zooms. Thus, our system is designed to identify a player by either facial or jersey number detection algorithm. These two algorithms will support each other in case of misinterpretations or missed detections. For facial detection, we have investigated the feasibility of utilizing several algorithms. First, the Dlib is a well-known toolkit with machine learning algorithms used in robotics, embedded systems, and high computing environments [4]. Although this toolkit's facial detection library performed adequately to detect players' faces but was only effective on zoomed-in camera feeds and was unsuccessful to detect any when in an aerial view mode (see Fig. 2).

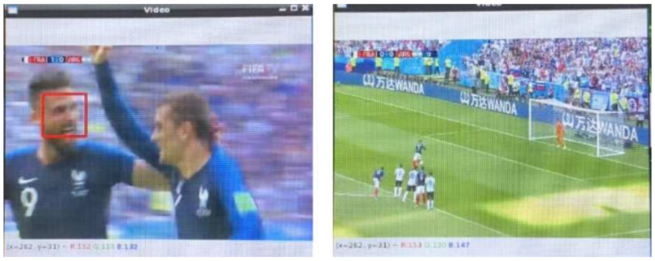


Fig. 2. Dlib Face Detection Results of Zoomed-In View (left) and Aerial View (right)

Another algorithm that we have explored is the Histogram of Oriented Gradients (HOG) in OpenCV [7] where HOG is widely used in computer vision and image processing for object detection purposes. This algorithm counts the occurrences of gradient orientation in localized portions of an image. It is often associated with a linear Support Vector Machine (SVM) algorithm to perform the classification of a person. Our test results showed that this algorithm too was only able to detect players when they were zoomed in the camera feed, but not when in the aerial view mode (see Fig. 3).

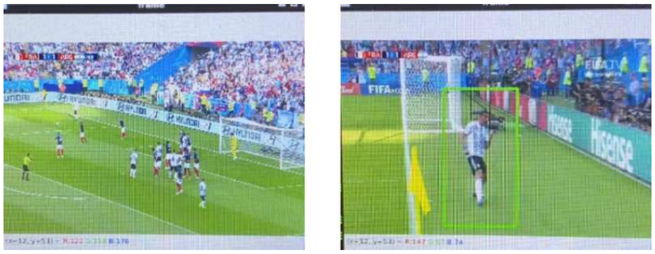
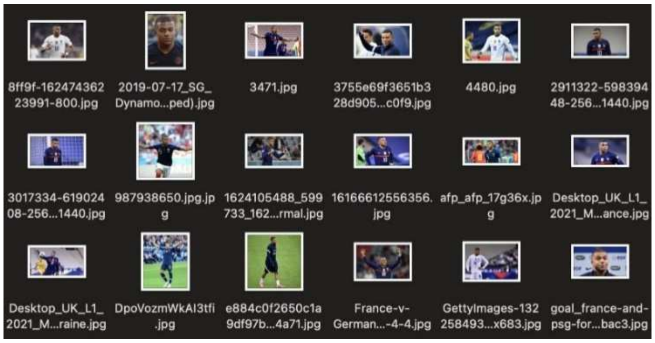


Fig. 2. HOG with SVM Algorithm Player Detection Result

As the above methods were not sufficient for our system design, we have explored a deep learning framework that would not interfere with the detection results depending on the view of the camera (aerial or zoomed-in). We have chosen to use Darknet [8], which is an open-source neural network framework for deep learning that provides several CNN models for object detection including AlexNet [9], ResNet [10], VGG [11], and YOLO [3]. In this paper, we decided to use the YOLO model. YOLO, "You Only Look Once", is a powerful, fast, and accurate deep CNN for object detections. It uses a single neural network that divides the image into regions and predicts the bounding boxes and probabilities for each region. We have utilized YOLOv3 for detecting the players and their faces in our system design. Section III describes the associated algorithms and their approaches in detail.

B. Recognition Algorithms

After the system has successfully detected the players, their faces, and jersey numbers, it needs to recognize the detected results to use them for profiling, game analysis, statistical database, and more. In this section, we specifically chose a soccer match, France Vs Argentina in the 2018 World Cup, to test the feasibility of the design approach. For both teams, 22 players on the field, we have gathered more than 3,000 images from the Internet and added them to our custom dataset, where each player's folder contains 50 different pictures of the player from different scenes (see Fig. 3)



8ff9f-162474362 23991-800.jpg
2019-07-17_SG_ Dynamo...ped).jpg
3471.jpg
3755e69f3651b3 28d905...c0f9.jpg
4480.jpg
2911322-598394 48-256...1440.jpg
3017334-619024 08-256...1440.jpg
987938650.jpg.jp g
1624105488_599 733_162...rmal.jpg
16166612556356. jpg
afp_afp_17g36x.jp g
Desktop_UK_L1_ 2021_M...ance.jpg
Desktop_UK_L1_ 2021_M...raine.jpg
DpoVozmWkAI3tfi .jpg
e884c0f2650c1a 9df97b...4a71.jpg
France-v- German...-4-4.jpg
GettyImages-132 258493...x683.jpg
goal_france-and- psg-for...bac3.jpg

Fig. 3. Illustration of Sample Images Collected for Kylian Mbappe (France)

From the gathered images, it is necessary to extract players' faces for creating our customized dataset. We have applied Haar Cascades Classifier's face detection algorithm [12] to all gathered images to extract faces from the image. Fig. 4 is the extracted faces of Kylian Mbappe as an example.

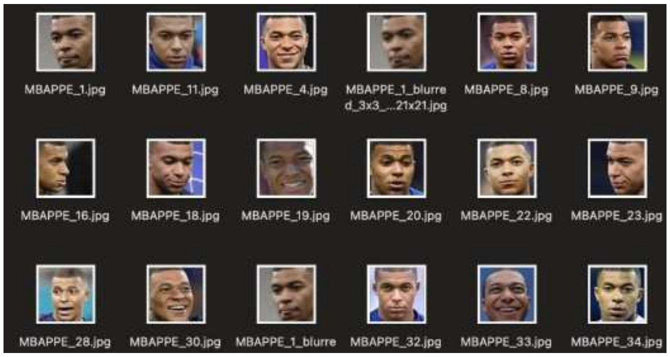


Fig. 4. Face Extraction Results from Gathered Image

Our system also applies a Gaussian filter with different kernel sizes to best simulate the blur on the captured image caused by the movement of the players. Fig. 5 is an example of different images generated after applying filters. As mentioned in the previous subsection, face detection in the soccer match would be only feasible if the camera is zoomed into the player and does not detect their faces when in the aerial view mode. Although Dlib is not the optimal solution to detect players' faces when in aerial view mode, our system still keeps this functionality in case of the zoomed-in view mode. For the aerial view mode, our system will utilize the jersey number recognition system to identify the players on the field.

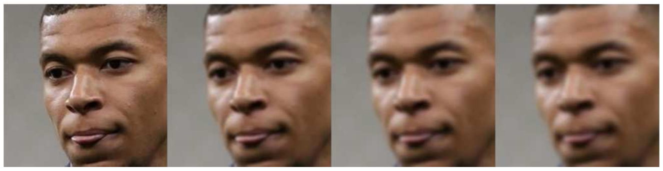


Fig. 5. Illustration of images generated after applying a gaussian Blurring filter with different kernels

For jersey number recognition, our system focuses on utilizing a customized dataset that comprises the MNIST database and Kaggle dataset. MNIST database of handwritten digits is a large dataset containing 60,000 images of digits and 10,000 test images. We chose this dataset as the fonts of the number on the jersey would differ from team to team. The Kaggle dataset containing printed digits was also included in our customized dataset. Our system uses the following specifications to train the model using CNN (see Table I).

TABLE I. THE ARCHITECTURE OF THE USED CNN FOR THE NUMBER RECOGNITION

Layer type	Parameters	Output feature map size
Convolutional layer	Filters = 8, kernel = 5x5, strides = 1, activation: relu	28x28x8
Max pooling layer	Pool_size = 2x2	14x14x8
Convolutional layer	Filters = 16, kernel = 3x3, strides = 1, activation: relu	14x14x16
Flatten layer	N/A	1x3136
Fully connected layer	Units = 32, activation: relu	1x32
Fully connecter layer	Units = 10, activation: softmax	1x10

III. IMPLEMENTATION AND RESULTS

A. Detection Algorithms

As mentioned in the previous section, YOLOv3 is used to detect the players on the field. Fig. 6 shows the result of detection of the players on the field, as well as detection of the ball used in the game. All players are identified correctly in green boxes, as well as the ball which can be used for ball movement analysis.

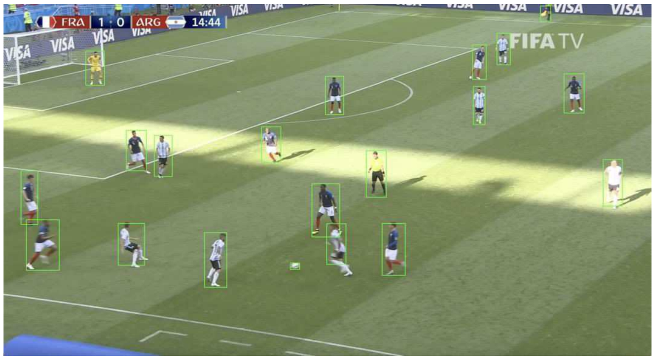


Fig. 6. Result of Player and Ball Detection using YOLOv3

As for the facial detection using YOLOv3, we have trained our customized YOLO model by utilizing LabelImg software [13] where it is possible to train the model by manually identifying a player on the region of interest. Fig. 7 (aerial view mode) and Fig. 8 (zoomed-in view mode) are the results of utilizing this algorithm applied to the camera feed where only the faces are correctly identified in blue boxes.

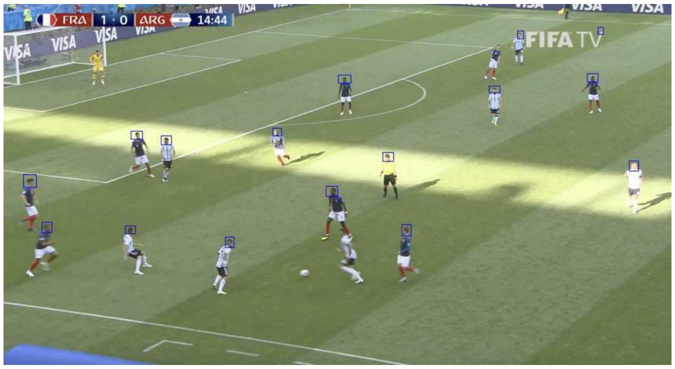


Fig. 7. Results using YOLOv3 Face Detection (Aerial View Mode)

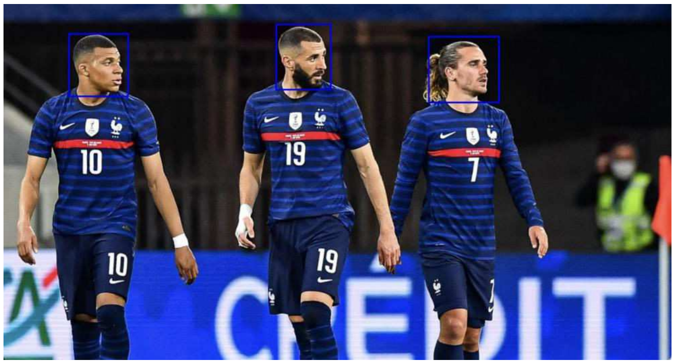


Fig. 8. Results using YOLOv3 Face Detection (Zoomed-in View Mode)

B. Recognition Algorithms

As mentioned in the previous section, recognizing players by their faces in the aerial view mode is challenging, and our system utilizes facial recognition and/or jersey number recognition to identify the players. For zoomed-in captured images, our system utilizes the Dlib toolkit to recognize which player is currently on the screen. Fig. 9 shows multiple recognized players in multiple scenes using the Dlib toolkit based on our customized dataset.

For jersey number recognition, as mentioned in the previous section, we created our customized dataset by combining the MNIST dataset with another dataset from Kaggle containing printed digits. The model was compiled with a categorical cross-entropy loss to minimize the accuracy and an Adam optimizer [14]. By using the parameters defined in Table I, the CNN training resulted in approximately 92% accuracy. To test our model, the input of the test images must have the same structure as the training data. To do so, it is necessary to convert captured jersey number images to grayscale to reduce the complexity of the image for recognition. We applied a Gaussian blur filter to remove false edges and smooth the image. Finally, we applied adaptive thresholding to emphasize the distinction between the background and the foreground as shown in Fig. 10. The quality of the recognition does not only depend on the performance of the model but also the quality of the input image for the inference. To use the aerial view of the camera to identify the jersey number of the player is difficult and would require pre-processing of the captured image before using it as the input to the created model. We have compared two thresholding algorithms to maximize the in-between class variance in the captured image, specifically for the jersey number recognition. Thresholding is an image segmentation processing technique used to make a clear distinction between classes in an image. Adaptive thresholding [15] is a simple method where the user inputs a threshold and every pixel value superior to that threshold is converted to black or white depending on the user's indications. The drawback of this method is that the quality of the thresholding depends on the user's input and that threshold won't be efficient in all captured images. Alternatively, Otsu Thresholding Method [16] computes a threshold that maximizes the in-between class variance in the picture, and it considers each pixel's intensity value, which means that each image will have a different threshold to provide accurate results. Fig. 11 illustrates the comparison between the Adaptive Thresholding and Otsu Threshold methods for jersey number recognition.

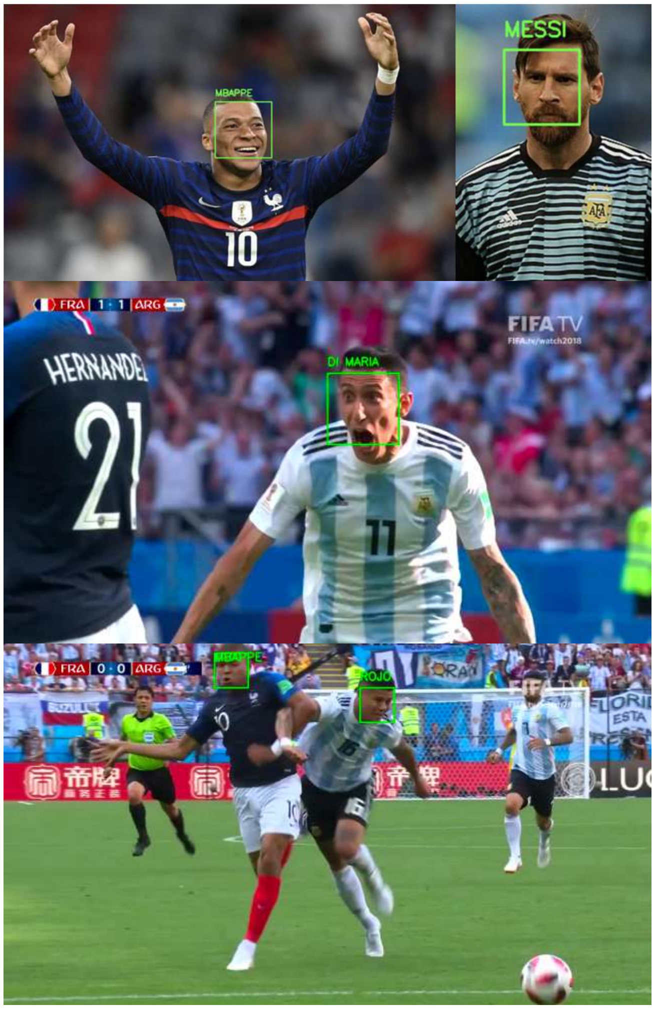
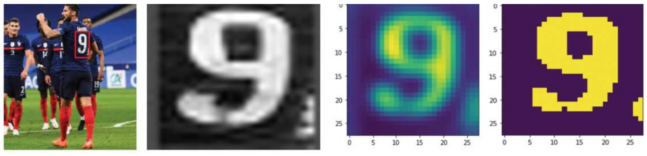


Fig. 9. Results obtained for face recognition using Dlib (Zoomed View)



Number Detection Convert to Grayscale Gaussian Blur Adaptative Thresholding

Fig. 10. Illustration of the pre-processing steps for number recognition

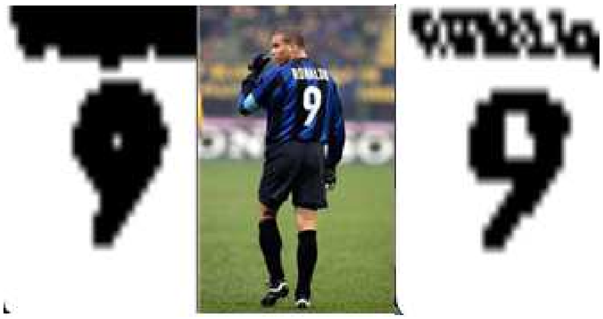


Fig. 11. Comparison of Adaptive Thresholding (left) and Otsu Thresholding Methods (right)

Fig. 12 illustrates the outcome of the jersey number recognition in zoomed-in camera view mode, and Fig. 13 illustrates the outcome of the jersey number recognition in the aerial camera view mode.

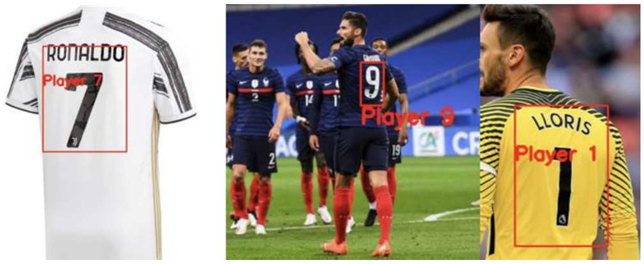


Fig. 12. Results of Jersey Number Recognition (Zoomed-In Mode)

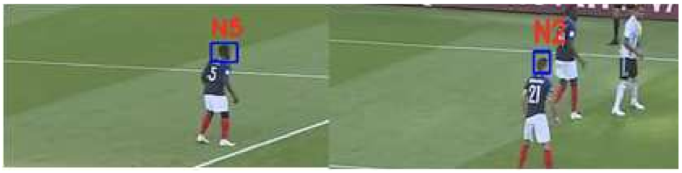


Fig. 13. Results of Jersey Number Recognition (Aerial View Mode)

As shown in Fig. 13, the obtained result is not always accurate as a single-digit number was correctly recognized but only recognized one of the digits when the jersey number was two digits. This is due to the quality of the captured image and the conditions of the lighting which can be improved for future studies.

IV. Conclusion

In this paper, we have explored AI and computer vision methods in the recognition of soccer players on the field using facial recognition and jersey number recognition. As soccer game broadcasts mainly view the game in aerial camera view mode, adopting detection and recognition methods were challenged throughout the study. Image processing results from the captured image of the zoomed-in view mode had accurate detection and recognition rates, whereas the image processing results from the captured image of the aerial view had difficulty in detecting and recognizing the correct faces and numbers. However, the obtained results show the potential of improvement of the overall system performance with more images to be included in the dataset training, and with improved quality of the captured image from the broadcast feeds.

REFERENCES

[1] G. Sundem, "What was the first televised sporting event?”. [Online] Available: https://people.howstuffworks.com/culture-traditions/tv-and-culture/10-tv-shows-that-have-gained-global-audience.htm [Last Accessed: Mar. 6 2022].

[2] AWS Machine Learning, "Calculating new stats in Major League Baseball with Amazon SageMaker". [Online] Available: https://aws.amazon.com/blogs/machine-learning/calculating-new-stats-in-major-league-baseball-with-amazon-sagemaker/ [Last Accessed: Mar. 6 2022].

[3] J. Redmon, A. Farhadi, "YOLOv3: An Incremental Improvement", 2018. arXiv:1804.02767.

[4] Dlib C++ Library – Algorithms. 2021. [Online] Available: http://dlib.net/algorithms.html [Last Accessed: Mar. 6 2022].

[5] Y. LeCun, C. Cortes, C. Burges, "MNIST handwritten digit database", 2021. [Online] Available: http://yann.lecun.com/exdb/mnist/ [Last Accessed: Mar 6. 2022].

[6] Kaggle Datasets. 2022. [Online] Available: https://www.kaggle.com/datasets [Last Accessed: Mar 6. 2022].

[7] S. Mallick, "Histogram of Oriented Gradients explained using OpenCV", LearnOpenCV, 2016. [Online] Available: https://learnopencv.com/histogram-of-oriented-gradients/ [Last Accessed: Mar 6. 2022]

[8] J. Redmon, "Darknet: Open Source Neural Networks in C", 2016. [Online] Available: https://pjreddie.com/darknet/ [Last Accessed: Mar 6. 2022].

[9] A. Krizhevsky, I. Sutskever, G. Hinton, "ImageNet classification with deep convolutional neural networks", *Communications of the ACM*, vol. 60, no. 6, pp. 84-90, 2017.

[10] K. He, X. Zhang, S. Ren, J. Sun, "Deep Residual Learning for Image Recognition", 2015. arXiv:1512.03385

[11] O. Parkhi, A. Vedaldi, A. Zisserman, "Deep Face Recognition", *British Machine Vision Conference*, 2015.

[12] Cascade Classifier, OpenCV, 2022. [Online] Available: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html [Last Accessed: Mar. 6 2022].

[13] Github, LabelImg. 2022. [Online] Available: https://github.com/tzutalin/labelImg [Last Accessed: Mar 6. 2022]

[14] D. Kingma, J. Ba, "Adam: A Method for Stochastic Optimization", 2017. arXiv:1412.6980.

[15] S. Abdullah, K. Omar, A. Zaini, M. Petrou, M. Khalid, "Determining adaptive thresholds for image segmentation for a license plate recognition system", *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 6, pp. 510-523, 2016.

[16] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62-66, 1979.