Underwater Hand Gesture Localization and Classification Using YOLOv5 and Other CNNs

Or De-Goede Matan Topel

Github link: https://github.com/OrDG/CADDY_Gesture_Classification_DL_Project

Introduction

Background:
CADDY is a project focused on developing an assistant diving robot that interacts with a diver and performs tasks, such as taking pictures of the area or moving the diver's boat around.
To communicate with the robot, the diver uses an invented sign language CADDIAN [9] that the robot should see with a stereo camera and understand. Therefore, a system that can "translate" images of the diver hand gesturing is critical for the robot's functionality. Now, although this kind of classifying system is common in a lot of complicated projects like this one, the challenges this system faces are definitely unique: the images of the diver are taken underwater where the lighting and visibility can change drastically between different environments or even different hours at the same location. Also, the diver's hands in the image might be unclear because of different objects in the frame, even the rest of the diver's body could potentially confuse the system. The translating system will need to work under all of these image shooting conditions to be able to help the diver at all times. As CNNs are very capable of classification of images[1], and are able to produce surprisingly good results even when the conditions are harshly set against them, we thought a classifying CNN would make a great solution to the challenge the CADDY project faced at this front. And although the project team have succeeded at creating a similar solution that used additional data, we decided to take their open access dataset of images taken for this cause, and try to create our own learning system that can be a part of such a special project.

project goal:
Creating a high accuracy CNN classifier of a diver's gestures from CADDIAN, using stereo images taken underwater in different water conditions.

Figure 1: The diver hand signing to the robot at all the different scenarios

Motivation:
Convolutional Neural Networks (CNNs) are deep neural networks that contain layers of stacked convolution layers or filters. It is known that CNNs are one of the methods of choice for processing images and supervised image classification [1]. Because of that, we believe that a system using CNNs can do a good job in classifying a diver's hand gestures.

A big motivation for our project was a system proposed in [2], which separated the problem of hand gesture classification into two simpler problems, localization of the hand and then classification of the hand gesture. Moreover, the proposed project used YOLOv3 as a solution for localization, which led us to research on more recent proven architectures of localization networks, such as YOLOv4 [3] and detectron 2 [4].

Finally we chose YOLOv5 [5] as the architecture for Localization in our project, mainly because of the ease of implementation using pytorch and its similarities to YOLOv4 in terms of accuracy and real-time capabilities (which we believe can be helpful for future work).

previous work:
We found two different articles which used the same dataset and had the goal as ours.

In [7], four different CNNs were trained and compared on the same dataset we used (AlexNet, VggNet, ResNet, GoogLeNe).

In [8], traditional computer vision techniques and deep learning methods (a classifier using Res-Net-50 backbone) were compared on the same dataset we used.

Method

Our network contains a localization part - YOLOv5s and a classification part - our own CNN.

YOLOv5:
There is not any published paper on YOLOv5, so we will explain to you the architecture of YOLOv4, because it has many similarities to YOLOv5.

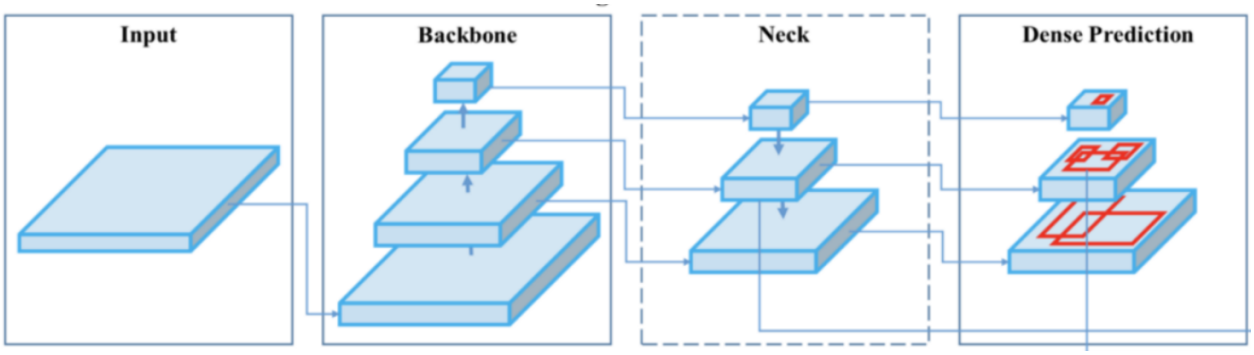

Figure 2: A typical object detection architecture of one stage detector:

- Backbone - A feature extraction network that can enhance the learning capabilities. YOLOv4 uses CSPDarknet53 [12].
- Neck - Collects feature maps from different stages of the Backbone . Usually, a neck is composed of several bottom-up paths and several top-down paths. In our case, there are two main parts:
  (1) A spatial pyramid pooling block over the Backbone that increases the receptive field and separates out the significant context features.
  (2) PANet [13] is used as a method for parameter aggregation from the detector levels.
- Dense prediction(Head) - Predicts classes and bounding boxes of objects. In our case, the system uses the same process as in YOLOv3 [14]. The network detects bounding boxes coordinates [x, y, width, height] as well as the confidence score for the detection.

YOLOv5 has 4 different kinds of networks, with different number of parameters:

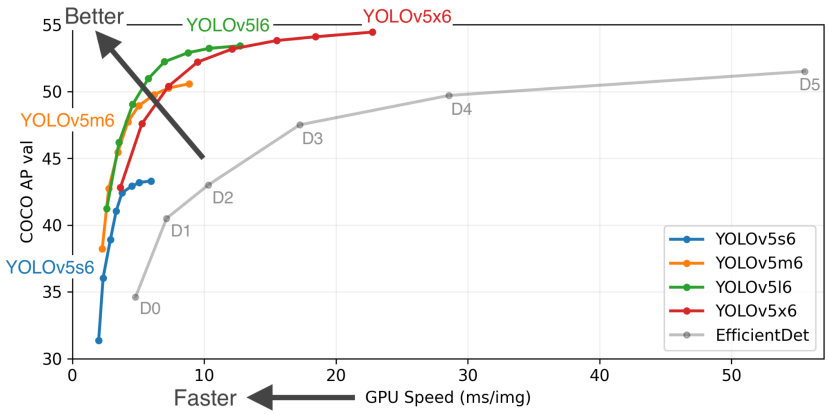

Figure 3: Performances of different versions of YOLO

We chose YOLOv5s as our favorable architecture for the task because we only predict one class of object, and also because we trained our network on Google Colab.

While modifying YOLOv5s to our system we tried to maximize the combination of Precision and Recall quality indexes:

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = 1 - \frac{FP}{TP+FN}$$

Where TP (True Positives), FP (False Positives), and FN (False Negatives) are areas of the image - Positives are pixels inside the rectangle containing the hand according to the coordinates given in the dataset, Negative are all the other pixels, and the True and False refer to the rectangle predicted by the network.[6]

Our CNN:
Our CNN architecture is pretty conventional. The network is built from 3 connected Convolution blocks, 3 fully connected layers after that, and finally a softmax layer that gives the signs probabilities as the output. Each conv block is structured from a convolution layer, BatchNorm layer, RelU, another convolution layer, another RelU and a max pooling layer. Each convolution layer increases the number of channels and decreases the size of each channel. As pointed out we used RelU activations and Adam optimizer. Overall, our CNN had 5,852,170 weights.

Experiments and results

The Dataset:
The open access dataset [10] we worked on included images taken by a stereo camera of the diver doing a sign from the hand sign language in a certain environment. The stereo camera is a type of camera with lenses for simulating human binocular vision. This means that for each time the diver was shot, there are two images of him, slightly different in their angles of shooting. The dataset included images of 16 different signs taken at 8 different environmental scenarios. In all of the images the diver is wearing unique gloves on both of his hands with different colors on his palm and digits. These colors are meant to help to distinguish the hands and its features from the rest of the black garment of the diver's suit and the features from each other. The coordinates of the diver's hands in each image are also included in the dataset.

Figure 4: Diver gloves and RAL-K7 chart

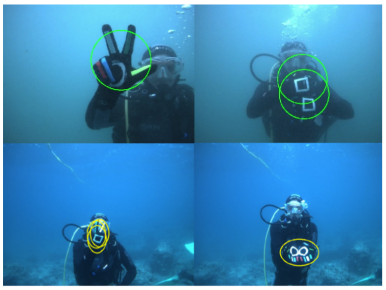

Figure 5: Detected gestures, from the top left, clockwise: take a photo, carry equipment, start communication, and go to the boat.

The CADDY crew also used sonar images which are not included in the open access dataset but played a big part in their classification ability.

Training, validation and testing:
First we trained our CNN on a cutted, resized and normalized trainset, and after some babysitting (Hyperparameters changed - epocs, learning rate, optimizer) we got 98.458% accuracy on the validset.

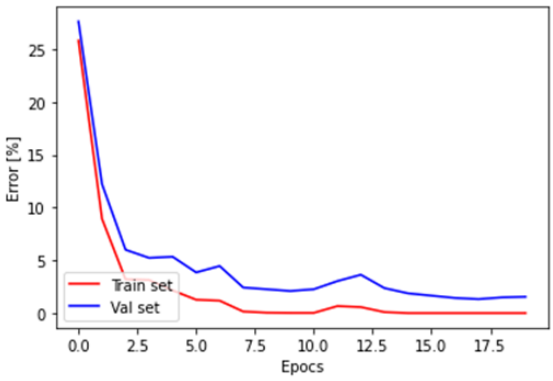

Figure 6: Error during training of our CNN

After that, we trained the YOLOv5s with the data on the cache of the GPU, to get faster training time. To do that, we had to train the network only on ¼ of the trainset (~3K images), but it proved to be enough, because after we tuned the hyper parameters on the valid set, the results on the testset were - precision: 0.997, recall - 0.986.

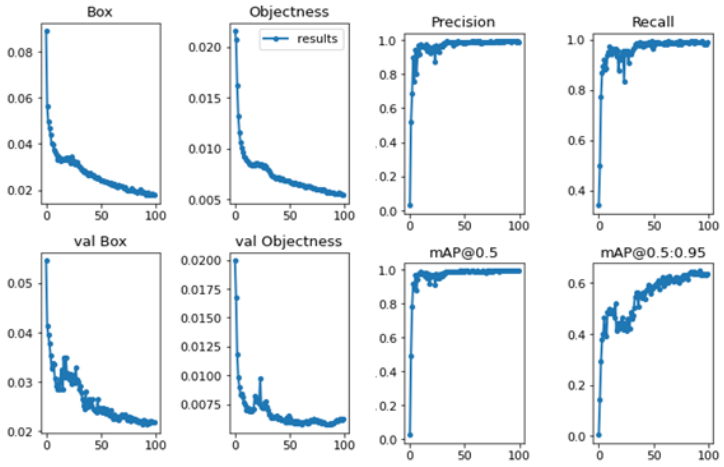

Figure 7: Quality indexes during training

When we ran our trained end-to-end network, we got 97.85% on the full stereo testset (~1.8K stereo images). Each network was trained in a little less than one hour.

Results of previous method/s:
In [7], using VggNet, the network got 95.00% accuracy on the testset (including true negatives).

In [8], using a classifier with a ResNet-50 backbone, the network got 97.06% accuracy on the testset (including true negatives).

Future work

While we worked on the project we constantly thought of ideas of additional implementations and functionalities our system could have if we could have worked on more data for more time.

The dataset we worked on did include a class of “true negatives” - images where the diver is not making an intended sign at the robot, meaning that the robot should not falsely interpret any of the words in the language. We thought about two possible solutions to this challenge.One option we could consider is putting some learned threshold on the output of the softmax layer calculating the signs probability. The threshold will determine whether there is enough certainty for the predicted sign. If there is no one sign with enough certainty we can say that there is probably no sign intended by the diver at all, and set the output to “true negative”. Another option is to create a new class called ‘none’, and train our CNN on the dataset of the full 17 classes.

If we want to make our system viable in real-time, we need to deal with another type of false negatives, where the diver is in frame but not necessarily facing the camera. For this goal, we can use another part of the CADDY project [11] which is meant to identify the angle the diver is facing. Using this information the network can determine the probability the diver is focusing on the robot, trying to communicate with it.

Dealing with “true negatives” is an important step toward real time operation of the system, which is a big and important challenge on its own. To work in real time our system will firstly need to work under time limitations made to ensure the robot can respond sufficiently on the field. Given our system's low classification time and the YOLOv5s [5] real-time capabilities we believe our system can already work under those time limitations. To improve our system even more, we can modify it using recurrent network techniques to use previous translated words as context for the current classification.

Conclusion

The CADDY's hand gesture recognition challenge is certainly an interesting one that involves a lot of factors such as light conditions, the background scenery, etc. An option for overcoming those factors is splitting the task to one part of localizing the diver's hands in the image, and a second part of classifying the image of only the hand, to the class fitting the sign's meaning. The separation of the hands from the background makes sure it does not affect the feature learning process done by the classification system. We decided to try this method in our attempts to solve CADDY's challenge. At the end of our work, we have succeeded in reaching the goal we have set for ourselves. We planned and built a classifying system, based on deep learning ideas, that can classify the stereo images of the diver correctly to their meant word in high accuracy. Even though we did not classify true negatives, our system used a small number of parameters (in comparison to previous works), and got a final accuracy of 97.85% on the testset. We also proposed viable solutions for the true negative dataset on future work.

The combined network stands as the end-to-end system produced to answer the problem we tried to resolve and after its training it is ready for work and is fully functional.

There is potential for future work on the system, especially for fitting it to real time operation and its applications such as identifying when no gesture is made and the use of previous context for classification, possibly by using recurrent network or transformer techniques.

Refrences:

[1] Y. LeCun, et al. “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.

[2] S. Sharma, H. Pallab Jyoti Dutta, M. K. Bhuyan and R. H. Laskar, "Hand Gesture Localization and Classification by Deep Neural Network for Online Text Entry," 2020 IEEE Applied Signal Processing Conference (ASPCON), 2020.

[3] Bochkovskiy, A., Wang, C. Y., Liao, H. Y. M., 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint arXiv:2004.10934.

[4] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, “Detectron2,” https://github.com/facebookresearch/detectron2, 2019.

[5] Jocher, G., , et al: YOLOv5 (2020). https://github.com/ult ralytics/yolov5.

[6] Han, M., Qin, S., & Zhang, N. (Eds.). (2020). *Advances in Neural Networks − ISNN 2020. Lecture Notes in Computer Science, page 241, 2020*.

[7] J. Yang, J. P. Wilson and S. Gupta, "Diver Gesture Recognition using Deep Learning for Underwater Human-Robot Interaction," OCEANS 2019 MTS/IEEE SEATTLE, 2019.

[8] Mygel Andrei M. Martija, et al. “Underwater Gesture Recognition Using Classical Computer Vision and Deep Learning Techniques”, Journal of Image and Graphics, Vol. 8, No. 1, March 2020

[9] D. Chiarella, et al. “A Novel Gesture-Based Language for Underwater Human–Robot Interaction”, Journal of Marine Science and Engineering, August 2018

[10] A. G. Chavez, et al, “CADDY Underwater Stereo-Vision Dataset for Human–Robot Interaction (HRI) in the Context of Diver Activities ”, Journal of Marine Science and Engineering, August 2018

[11] A. G. Chavez, et al, "Stereo-vision based diver pose estimation using LSTM recurrent neural networks for AUV navigation guidance," OCEANS 2017 - Aberdeen, 2017.

[12] Chien-Yao Wang, et al. CSPNet: A new backbone that can enhance learning capability of cnn. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPR Workshop), 2020. 2, 7

[13] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 8759–8768, 2018. 1, 2, 7

[14] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” CoRR, vol. abs/1804.02767, 2018