

Smart Arms Detection System Using YOLO Algorithm and OpenCV Libraries

Rayan KHALAF¹, Yaman AKBULUT^{2*}

^{1,2}Department of Software Engineering, Technology Faculty, Firat University, Elazig, Turkey
rayan.s.alhaddad@gmail.com, yamanakbulut@firat.edu.tr

(Geliş/Received: 04/02/2021;

Kabul/Accepted: 15/02/2021)

Abstract: The continuous development that the world is on must be pursued into human services, among these services provided by the modern world are the ability to detect faces, controlling and monitoring crowded places, the roads, auto-drive cars, and other smart systems that we can see through our daily life, these systems are known as object detection systems. This branch is getting development continuously and it gets big attention and support from international institutes, military institutions, and security companies. Object detection is a big branch and is used in many industries like medical, food production, plastic industries, controlling and monitoring roads, and more. In this paper we will explain information about object detection, its types, the YOLO algorithm, and a small idea that we made for controlling and monitoring arms showed by monitoring cameras or images that contain arms, we made it using the YOLO algorithm with the help of OpenCV and python libraries.

Keywords: YOLO, object detection, arms detection, monitoring, digital.

YOLO Algoritmasını ve Açık Kütüphanelerini Kullanan Akıllı Silah Algılama Sistemi

Öz: Dünyanın üzerinde bulunduğu sürekli gelişme, insan hizmetlerine dönüştürülmelidir, modern dünyanın sunduğu bu hizmetler arasında yüzleri algılama, kalabalık yerleri, yolları, otomatik sürüş arabaları ve diğer akıllı sistemleri kontrol etme ve izleme yeteneği vardır. günlük hayatımızda görebildiğimiz bu sistemler, nesne tespit sistemleri ile biliniyor. Bu şube sürekli gelişmekte ve uluslararası enstitülerden, askeri kurumlardan ve güvenlik şirketlerinden büyük ilgi ve destek almaktadır. Nesne algılama büyük bir daldır ve tıp, gıda üretimi, plastik endüstrisi, yolları kontrol etme ve izleme ve daha fazlası gibi birçok endüstride kullanılmaktadır. Bu çalışmada nesne algılama, türleri, YOLO algoritması ve el kollarını içeren kamera veya görüntüleri izleyerek gösterilen el kollarını kontrol etmek ve izlemek için yaptığımız küçük bir fikri açıklayacağız, YOLO algoritmasını kullanarak yaptık. opencv ve python kütüphanelerinin yardımı.

Anahtar kelimeler: YOLO, nesne algılama, silah tespiti, izleme, dijital.

1. Introduction:

The news circulating the world speaks of the constant struggle between development and ignorance, between reaction and backwardness, between light and darkness. And what the world is witnessing in terms of development in the field of science must be exploited by the leaders of the civilized world to make the world safer so that researchers and people, in general, can practice their normal lives without fear. From this standpoint, we found that devoting technology to serve humanity through smart systems that can monitor and control all vital, crowded places and state institutions is one of the responsibilities of the leaders of the civilized world. We began to search for a technology that can give a service to the community in a modern and smart way and we found it in computer vision.

Computer vision is a branch that is growing so fast and every day we find an invention or enhancement in its branches. It is a process that takes images as an array of digits, analyzes them depending on filters, and extracts their features. Then, it classifies them into a specific class or group [1]. Some problems appear in computer vision such as an image is 2D while the real world is 3D, the difference in point of view, the difference in size (near or far), the difference in color (like different clothes), the resolution, and the size of the image, the noise that appears in images because of the surrounding circumstances, etc. [1, 2].

A type of computer vision is object detection, one of the most modern and updated branches inside computer vision, also we can say is the most spread in computer vision. It differs or enhanced algorithm from classification types because in object detection we can detect more than one variable in the same image [3]. This type of computer vision has become widespread in multiple branches like medicine, industry, surveillance, and

* Corresponding author: yamanakbulut@firat.edu.tr. ORCID Number of authors: ¹ 0000-0002-9991-8844, ² 0000-0002-4760-4843

more others. The basic idea is to detect more objects in one image or frame especially in real-time, and our research will be in this branch (object detection in real-time), exactly in the YOLO algorithm.

Terrorist incidents are among the most common incidents that threaten the security of society. We found the idea of using technology as a weapon in the face of such incidents is a good idea and has great goals, one of which is to reduce the risk of a terrorist attack for any institution and also to support expertise and competencies to work intensively for more intelligent inventions. From this point, the idea of making this system we will talk about is coming. The system is an arms detector in a smart way built on a YOLO algorithm with OpenCV and python libraries

2. Deep Learning

Deep learning is a technology to analyze and extract features from data like images, frames, and sounds, this branch developed in fast steps to be more spread all around the world and it is divided into branches exploited in most of our daily life [15]. The name (deep) came from the structure of the algorithm layers, it is referred to the hidden layers of convolution used inside its structure [17]. One of the biggest deep learning branches is object detection, in the object detection branch, we are talking about voices, images, and frames and how this technology deals with them. The most popular technology used in this branch is CNN because of its characteristics [15]. The increase came to use in more branches like medicine, industry, and other branches after the growth of datasets, CPUs, and intelligent machines in general, this growth led to the use of deep learning to be responsible for making rapid and correct decisions depending on the dataset and an algorithm [16, 17].

2.1. Convolutional neural network

CNN One of the most advanced areas in machine learning and image processing, the name is taken from the analogy of human nerves and how the human brain works from convolutional neural networks [4]. Its history dates back to the 1990s when LeCun et al. used a gradient-based machine learning algorithm on CNN and he got successful outputs of classifying digits wrote by hands [5]. As the work of this technique is very like the work of the nerves of the human brain of complexity, accuracy, and the speed of processing and, above the human brain in some operations, new techniques in this field have proven superior to the human brain through the speed and accuracy of the human brain in parts of a second [4]. As it is a branch of NNs it gave good features to this branch, because it fixes many problems that appeared in NNs like the big number of parameters, the location of the specific object, besides the speed is an improvement [18]. This technique proved its success through more global applications like GoogLeNet, DeepFace, Go game, and others [4]. CNN's processing consists of convolution, pooling, and fully connected layers that are specialized to deal with different input dimensions with different layers as images, video frames, and voices [6, 19]. CNN's work on extracting Images features on many levels to give the processor enough information to detect the object in a fast and accurate way depending on its tools like loss function, convolution, and pooling [19]. There is a function Deep Convolutional Neural Networks (DCNN) uses to process faster than the traditional CNN's multiple times that is So-Called Rectified Linear Unit (SCRLU) [6]. This success goes to the invention of high-speed CPUs, GPUs, big datasets, and improved algorithms as we mentioned latterly, we will explain some of the big datasets and improved algorithms used lastly.

2.2. Object detection

This branch was invented in the 1990s, but its applications spread after that maybe in the 2000s. That is because in the 1990s we did not have big data sets, good algorithms; good funds specified to do AI studies like provided today from companies like Google, Amazon, Facebook, etc., and good CPUs and GPUs [7]. Generally, object detection is divided into two branches, one is classifying the image or frame in two stages like R-CNN, another one is taking the image in one step for classifying like YOLO [20]. We discovered how datasets are made and what are the contents of the dataset are; let's take an overview of the most popular algorithms used in these databases to achieve object detection and classification. The most popular object detection algorithms are:

- R-CNN
- Fast R-CNN

- Faster R-CNN
- Mask R-CNN
- Single Shot Detection
- You Only Look Once

2.3. You Only Look Once (YOLO)

Born of a new object detection approach after multiple improvements of object detection that uses bounding boxes of object detection instead of classifying whole image pixels (that have been used in object detection algorithms before), also is a real-time object detection because of the improvement in speed of recognition, the previous approaches did not provide it, they were slower than YOLO so it cannot detect real-time videos while YOLO did that, in YOLO algorithm takes a single look to the image and extract the objects that made YOLO much faster but this effects the accuracy too, the table below defines object detection approaches according to its born with its properties [8].

Table 1.1 Comparing YOLO with other object detection approaches [8].

Detection frameworks	Train	mAP	fps
Faster R-CNN	VOC 2007+2012	70.0	0.5
Faster R-CNN VGG16	VOC 2007+2012	73.2	7
Faster R-CNN Resnet	VOC 2007+2012	76.4	5
YOLO	VOC 2007+2012	63.4	45
YOLOv2 288 × 288	VOC 2007+2012	69.0	91
YOLOv2 352 × 352	VOC 2007+2012	73.7	81
YOLOv2 416 × 416	VOC 2007+2012	76.8	67
YOLOv2 480 × 480	VOC 2007+2012	77.8	59
YOLOv2 544 × 544	VOC 2007+2012	78.6	40

YOLO idea is cleverer than previous approaches like Deformable Part-based Method (DPM) because it takes region proposals inside an image with multiple class probabilities and processes it simultaneously to get faster (this is the idea from the name YOU ONLY LOOK ONCE). It takes several steps to do the best [8].

- Dividing an image into a grid of 13×13 cells
- Use bounded boxes with probable classes to predict objects
- Use a single CNN to optimize the RP's
- Loss function
- Non-maximal suppression and fast processing get to 45 fps.

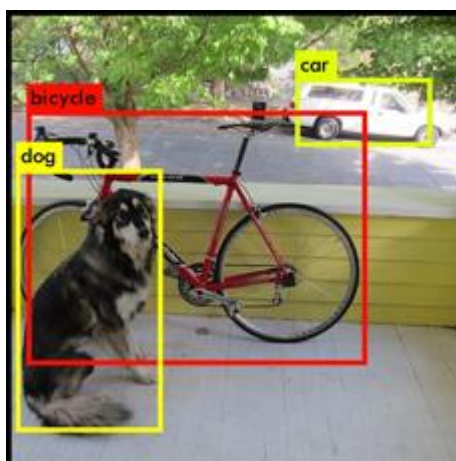


Figure 1. Regions remained that contained objects [8].

2.4. YOLO's vision to the image

YOLO's first step is dividing an image into a grid with 13×13 cells; cells that are the centers of the objects are responsible for detecting that object. This cell will predict the bounding of five cells [8].

So we have $13 \times 13 = 169$ cells, and every cell predicting its five bounding cells that lead to having 845 bounding cells. That is a big number of cells that need big time to process, but in YOLO it takes just that boxes having 30% or more of similarity (programmer can change this value depending on the accuracy needed) [8].

2.5. Loss Function

Loss function works on taking the bounding boxes that most, contain an object. As we can see in Figure 1 there are only three boxes that contain an object while the algorithm produces a big number of possibilities, but only detect three boxes that pass the percentage. This predicted boundary gives that what size and form of an object are beside of giving an exact bounded box (object size), in general, it defines objects through a formula that [14]:

$$pr(object) * iou_{pred}^{truth} \quad (1)$$

Where Pr denotes the proposal region and IOU denotes the intersection of a union. This formula gives the decision if there is an object or not, if there is no predicted object confidence score will be zero. If there is, it should take the IOU between our predicted box with a ground truth box [8, 13, and 14].

2.6. YOLO V3 the latest version

At the end of our research, we will talk about the last version of YOLO, which is YOLOV3 that is much faster and more accurate than the other versions. In this version, we will find small changes but it makes the detection operation more accurate [9]. Firstly the bounding boxes are the same as versions before, using anchors for making bounding boxes, the decision of having an object inside the RP or not is taken depending on assigning bounding box for each object, if not assigned it will be discarded [9]. In the prediction of class, they used independent logistic classification instead of softmax prediction that makes it faster than previous versions [9]. In the state of prediction they used three different scales of boxes, they tried it on the COCO dataset with the Darknet platform [9].

3. ARMS DETECTION EXPERIMENT

In this section, we will explain how YOLO is working with custom object detection, and we took a gun named Automatic refill Kalashnikov Modernized (AKM-47) as an example for gun detection.

Windows is a good platform to apply our idea of guns detection, but it needs some tools to be adaptive with the YOLO algorithm, as:

- Git tool: which is an open-source tool designed to help and handle small and big projects [10].
- Cygwin: this is a tool providing an environment like UNIX but on windows, also named GNU. This tool provides some functions that we need to build Darknet environment like (make, ssh, ssl, gcc-core, vim editor) [11].
- Installing Darknet: this is an open-source framework that supports neural networks and YOLO as a part of NN's, written in CUDA and C. supporting CPU and GPU too [12].

Now the environment is ready to install the YOLO algorithm.

3.1. Installing YOLO algorithm

First, we must download YOLO pre-trained weights that weights are provided by YOLO. If the installation is completed successfully we should test it before going to build our custom detection system. The most popular picture is below to know the system is installed successfully:

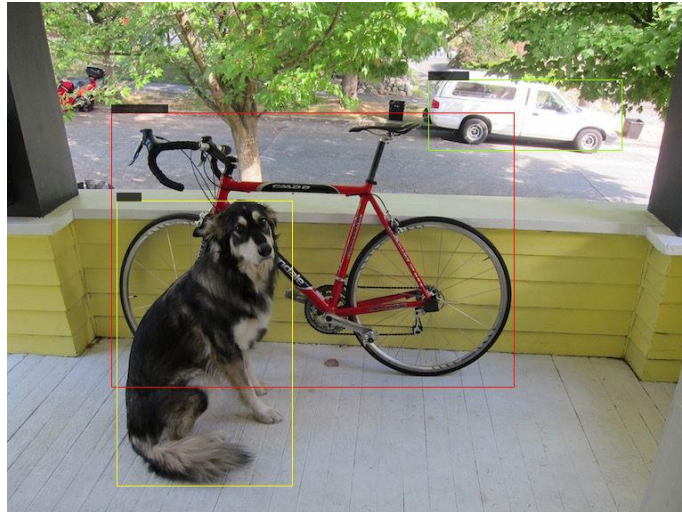


Figure 1. YOLO detection

YOLO is working now, the next step we will build our specific detection system (AKM-47).

3.2. Collecting images for AKM-47

The most important step when we decide to build a detection system on a specific object is collecting enough images about that object because it is the basic stone of building that system. In our experiment, we searched for images of AKM-47 on Google and we collected 100 different images, but this number of pictures is not enough for our system because of the bad resolution and the similarity. We decided to build our dataset by hand; we took 825 images of the object and prepared it to be ready to be a dataset. As we can see below:



Figure 2. AKM-47 dataset

We depended on the factors of resolution and the variety of angles in collecting and taking images, to make our dataset include most of the object angles and position sides. After collecting we need to resize all the images in the same size to make it through the processing. All images are resized to 180×180 pixels, the size we have chosen is to make the processor faster through the process of the weight. For resizing images to the specific size we used paint application to prepare it. All images are then collected in a folder named images to make it ready for the next step.

This step will take a long time depending on the CPU it works on, and if GPU is available or not. It takes 14 days if we are using CPU only, as we did in this project. The most important step is loss function and controlling how it works because detection depends on it. As below:

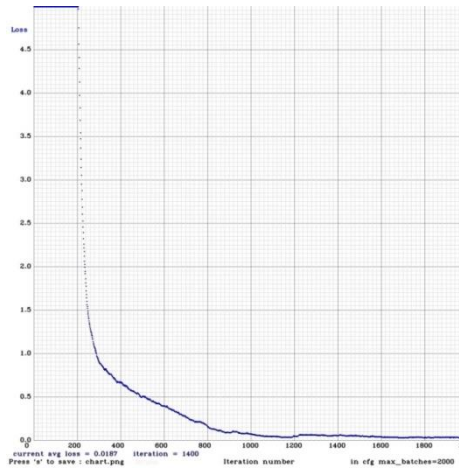


Figure 5. Loss function

The final step is testing our work using our weights instead of YOLO weights and test if it is detecting or not. As below:

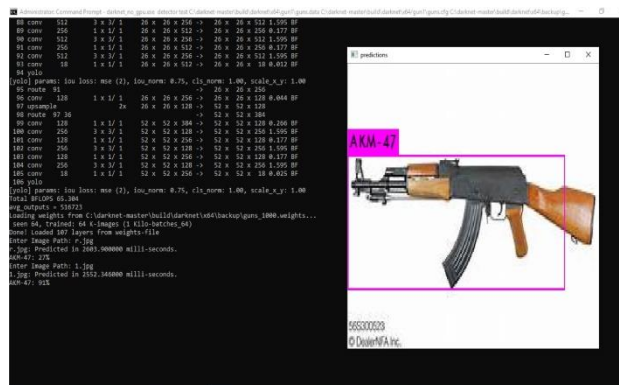


Figure 6. Testing the weights

As we note above the system detects AKM-47. Accuracy and speed depend on dataset and CPU, GPU, in our experiment the accuracy is reaching 90% in image detection within 2.5 seconds, and in the video, the accuracy is between 80% and 95% with a rate of 0.3 frames per second (fps).

4. Conclusion

Object detection with neural networks works very well inside controlling and surveillance systems, everything we can make with this technology means we can build monitoring systems for controlling vehicles, civil cars, persons, guns, etc. especially in military institutions because of its sensitivity. Our idea is simple but at the same time, it is a good idea if it is taken by institutions, government departments, or security companies, with a group of programmers, good computers, and technician persons.

There is another good idea for building a smart security system if it gets good care and support, the idea is connecting an X-ray to scan places and transfer the video to a guns detection system, in this manner the security will be 100% and the system will be smart enough to detect guns at all places, giving a piece of good information to the security managers.

There are new ideas always growing up, we hope in future we will perform smarter systems with better detection and more intelligence. We hope in future we can make an advanced system, contains more guns and

objects with good computers and bigger dataset, offer it to all military and public areas to control it from any terrorist attacks. Finally, we thank this modern technology for helping humans to be safer.

References

- [1] Kaehler, A., & Bradski, G. (2016). *Learning OpenCV 3: computer vision in C++ with the OpenCV library*. " O'Reilly Media, Inc."
- [2] Szeliski, R. (2010). *Computer vision: algorithms and applications*. Springer Science & Business Media.
- [3] Zafar, I., Tzanidou, G., Burton, R., Patel, N., & Araujo, L. (2018). *Hands-on convolutional neural networks with TensorFlow: Solve computer vision problems with modeling in TensorFlow and Python*. Packt Publishing Ltd.
- [4] Stuart, J., & Norvig, P. (2010). *Artificial Intelligence: a Modern Approach* Prentice-Hall. In *A Simon & Schuster Company Englewood Cliffs*.
- [5] Alom, M. Z., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M. S., ... & Asari, V. K. (2018). The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*.
- [6] Runia, T. F. H. (2015). *High-Speed Object Detection: Design, Study and Implementation of a Detection Framework using Channel Features and Boosting*, Master Thesis.
- [7] Ketkar, N., & Santana, E. (2017). *Deep learning with python* (Vol. 1). Berkeley, CA: Apress
- [8] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).
- [9] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- [10] <https://git-scm.com>, access: 2 February 2020.
- [11] <https://www.cygwin.com>, access: 2 February 2020.
- [12] Redmon, J.. *Darknet: Open source neural networks in c*. 2013.
- [13] Atienza, R. (2020). *Advanced Deep Learning with TensorFlow 2 and Keras: Apply DL, GANs, VAEs, deep RL, unsupervised learning, object detection and segmentation, and more*. Packt Publishing Ltd.
- [14] <https://machinethink.net/blog/object-detection-with-yolo>, access: 18 April 2020.
- [15] Z. Zhang, P. Cui and W. Zhu, "Deep Learning on Graphs: A Survey," in *IEEE Transactions on Knowledge and Data Engineering*, doi: 10.1109/TKDE.2020.2981333.
- [16] Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., ... & Dean, J. (2019). A guide to deep learning in healthcare. *Nature medicine*, 25(1), 24-29.
- [17] X. X. Zhu *et al.*, "Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources," in *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8-36, Dec. 2017, doi: 10.1109/MGRS.2017.2762307.
- [18] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017, August). Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)* (pp. 1-6). Ieee.
- [19] T. Guo, J. Dong, H. Li and Y. Gao, "Simple convolutional neural network on image classification," 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), Beijing, 2017, pp. 721-724, doi: 10.1109/ICBDA.2017.8078730.
- [20] Z. Zhao, P. Zheng, S. Xu and X. Wu, "Object Detection With Deep Learning: A Review," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3212-3232, Nov. 2019, doi: 10.1109/TNNLS.2018.2876865.