

**King Fahd University of Petroleum and Minerals**

*College of Engineering & Physics*

*Control & Instrumentation Engineering Department*

**CISE 483**

**AI & ML for Robotics**

**Personal Protective Equipment Detection Using YOLOv3**

**Group members:**

**ABDULELAH BUKHAMSEEN 201646300**

**MUHANNAD ALSHEHAB 201670280**

**Outline:**

|  |  |
| --- | --- |
|  | Page |
| Introduction | 3 |
| Computer Vision | 3 |
| Convolutional Neural Networks | 4 |
| You Only Look Once | 6 |
| Helmet Detection Model |  |
| Conclusion | 12 |
| References | 13 |

**Introduction:**

Safety is one of the key factors that contribute to the success of any organization, especially factories where accidents could cause threatening damages to labors or operators, such damages mostly occur due to the absence of Personal Protective Equipment (PPE). PPE has proven to be effective and essential in providing the needed safety measures that prevents such damages. Unfortunately, some workers have leniency to comply to PPE as it could be uncomfortable and make the job harder, such behaviors has been the cause of many fatal accidents. As technology evolve and flourish, we now have the capability to detect helmets using Computer Vision and Machine Learning techniques. In this report, we will explain how Computer Vision enables the computer to see, recognize, and localize helmets by using trained Convolutional Neural Networks.

**Computer Vision:**

Computer vision is a field of artificial intelligence that enables computers and systems to derive meaningful information from digital images, videos, and other visual inputs, computer vision also helps computers take actions or make recommendations based on that information. If AI enables computers to think, computer vision enables them to see, observe and understand. The most used techniques to achieve Computer Vision are convolutional networks. The model used to detect helmets consist of a deep convolutional network.

**Convolutional Neural Network CNN:**

To understand what a Convolutional Neural Network is, one must understand the concept of a Deep Neural Network DNN.

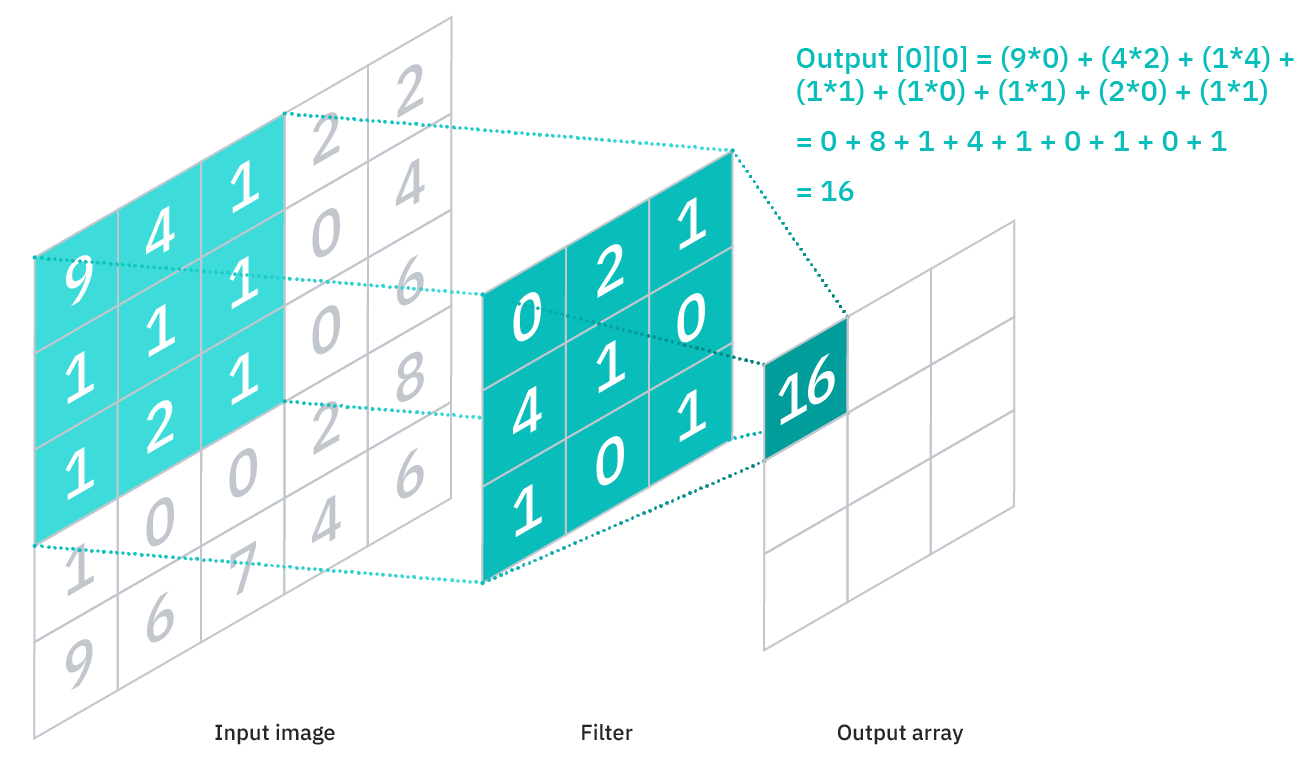
DNN is an artificial neural network that consist of multiple layers between input and output layers, layers consist of nodes each carrying an activation function that connects to the next layer through a connection that carry a certain weight. Usually, a DNN is used to derive high-level functions by transforming the data into a more creative and abstract components.

There are multiple of DNN architectures that leverage the structure of the network to achieve a certain object, for example the Convolutional Neural Networks CNN.

Convolutional Neural Networks CNN is one kind of DNN that has one or more layers of convolutional units.

Diagram

Description automatically generatedThe role of the CNN is to filter the images into a form which is easier to process, without losing features which are critical for getting a good prediction.   
A convolution layer is the core of a CNN where most of the computation happens. It takes image matrixes as an input and give a simplified (smaller) matrix as an output; the simplification process is done through filtering the images to extract the important values (features). The number of filters and their sizes are predetermined and will not be affected by the training, just like the number of layers and nodes in a regular neural network. The training will affect the filter weights only and this will change the how the network extract features from images.



**You Only Look Once (YOLO):**

YOLO is a short name for “You Only Look Once” which is an algorithm used for object detection and recognition in computer vision. Basically, it answers two questions:

1- What is the object? This question for identifying the object in a specific image.

2- Where is it? This question for establishing the exact location of the object within the image.

Object detection consists of various approaches such as fast R-CNN, Retina-Net, and Single-Shot MultiBox Detector (SSD). But all the previous methods could not detect many objects in a single algorithm run. However, YOLO algorithm considers a development of them, and it has gained popularity because of its exceptional performance over the object detection techniques. Object detection in YOLO is done by using the theory of regression in problem and provides classification probabilities of a specific image.

YOLO algorithm applies CNN to detect objects in real-time. the algorithm involves only a single forward propagation through a neural network to detect objects (single run).

Why YOLO? There are many factors for chosen a particular algorithm based on your objective. We choose YOLO for three main factors. First factor is the speed. The speed of the detection can predict the objects in real-time. Second factor is the high accuracy. The algorithm provides accurate results within minimal background errors. The last factor is the learning capabilities. Which make it great in learning the representations of objects.

There are various variants for YOLO algorithm: YOLOv1, YOLOv2, and YOLOv3. In our project we used YOLOv3.

How YOLOv3 works:

YOLO v3 is based on a Darknet variation that used to have a 53-layer network trained on ImageNet. For detection, 53 more layers are put on top of it, giving YOLO v3 a 106-layer fully convolutional underlying architecture.

A screenshot of a video game

Description automatically generated with medium confidence

Three-Scale Detection

Residual skip connections and upsampling are part of the modern design. The most notable feature of version 3 is that it detects on three distinct scales. YOLO is a fully convolutional network with a 1 x 1 kernel applied to a feature map as the final output. Detection in YOLO v3 is accomplished by using 1 x 1 detection kernels on feature maps of three different sizes at three different locations in the network.

Graphical user interface

Description automatically generated1 x 1 x (B x (5 + C) is the shape of the detection kernel. B stands for the number of bounding boxes a feature map cell can predict, "5" for the four bounding box attributes and one object confidence, and C stands for the number of classes.

I’d like to point out that stride of the network, or a layer is defined as the ratio by which it downsamples the input. In the following examples, I will assume we have an input image of size 416 x 416.

YOLO v3 makes prediction at three scales, which are precisely given by downsampling the dimensions of the input image by 32, 16 and 8 respectively. The 82nd layer is the first to detect something. The image is down sampled by the network for the first 81 layers, with the 81st layer having a stride of 32. If we start with a 416 x 416 image, the resulting feature map will be 13 x 13. The 1 x 1 detection kernel is used to make one detection, yielding a detection feature map of 13 x 13 x 255. The feature map from layer 79 is then passed through a few convolutional layers before being up sampled by 2x to 26 x 26 dimensions. The feature map from layer 61 is then depth concatenated with this one. After that, a few 1 x 1 convolutional layers are applied to the merged feature maps to fuse the features from the previous layer (61). The 94th layer then performs the second detection, resulting in a detection feature map with dimensions of 26 x 26 x 255. The feature map from layer 91 is treated to a few convolutional layers before being depth concatenated with a feature map from layer 36, following a similar approach. Following that, a couple 1 x 1 convolutional layers merge the information from the preceding layer, as previously (36). The final of the three is created at the 106th layer, providing a feature map with dimensions of 52 x 52 x 255.

A picture containing diagram

Description automatically generatedYOLO v3 employs a total of nine anchor boxes. There are three for each scale. If you're using your own dataset to train YOLO, you should use K-Means clustering to generate 9 anchors. Then, in descending order of dimension, place the anchors. Assign the first scale's three largest anchors, the second scale's next three, and the third scale's last three.

**Helmet Detection Model:**

We found a YOLOv3 trained model that is used to detect helmets. The model takes images with size 416x416 and color channels RBG as an input and gives 4 outputs. Outputs are x, y, box width, and box height. First two outputs, (x, y), are coordinates of two opposite corners of the box that will localize the helmet, the rest are the height and width of the rectangle (box). To test the model in Google Colab, we used a code that builds the network and feed the image to the network, at the end we print the output on the original image. Specifications will be included in the code.

**Conclusion:**

All in all, by using the Convolutional Neural Network in the YOLOv3 model we were able to implement Computer Vision by detecting and localizing helmets in the inserted images. The same model could be used to detect other PPEs.

References:

* <https://www.ibm.com/cloud/learn/convolutional-neural-networks>
* <https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b>
* <https://github.com/rafiuddinkhan/Yolo-Training-GoogleColab>