SEMINAR REPORT

ON

Real Time Object Detection Using You Only Look Once (YOLO)

Submitted by

Alan Anto (SJC20AD006)

to

the APJ Abdul Kalam Technological University

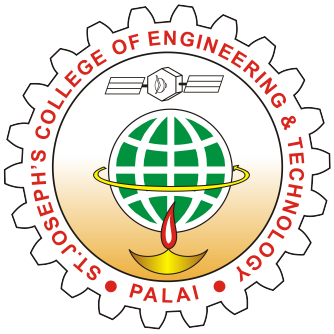
in partial fulfillment of the requirements for the award of the degree

of

Bachelor of Technology

in

Artificial Intelligence and Data Science



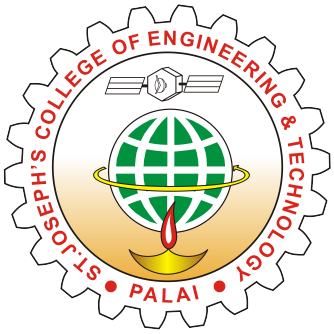
Department of Artificial Intelligence and Data Science

St. Joseph's College of Engineering and Technology, Palai

DECEMBER : 2023

ST. JOSEPH'S COLLEGE OF ENGINEERING AND TECHNOLOGY, PALAI

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE



CERTIFICATE

This is to certify that the seminar report entitled **Real Time Object Detection Using YOLO** submitted by **Alan Anto (SJC20AD002)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Artificial Intelligence and Data Science is a bonafide record of the seminar carried out by him under my guidance and supervision.

Seminar Guide
Lakshmi G
Assistant Professor
Department of AD

Seminar Coordinator
Mr.Jacob Thomas
Assistant Professor
Department of AD

Head of Department
Dr. Deepa V
Associate Professor
Department of AD

Place: Choondacherry
Date: 01-12-2023

Acknowledgement

I wish to record our indebtedness and thankfulness to all who helped me to complete this seminar titled "Real Time Object Detection Using YOLO". I would like to convey a special gratitude to Dr. V.P. Devassia, Principal, SJCET, Palai, for the facilities. I express my sincere thankfulness to Dr. Deepa. V, Head of the department, Department of Artificial Intelligence & Data Science for her cooperation and valuable suggestions. Also, I express my sincere thanks to the seminar coordinator Mr. Jacob Thomas for his helpful feedback and timely assistance. I am especially thankful to my guide,Lakshmi G, Assistant professor, Department of Artificial Intelligence & Data Science for giving me valuable suggestions and critical inputs through guidance and support. I also extend my thanks to college lab technicians, my friends, and others who directly or indirectly helped me during this seminar work.

Alan Anto

Abstract

You Only Look Once (YOLO), which is a popular and influential computer vision algorithm used in object detection tasks. YOLO revolutionized the field of object detection by introducing a real-time approach that can rapidly identify and locate multiple objects within an image or video frame. Unlike traditional methods that use multiple stages and complex operations, YOLO takes a different approach by dividing the image into a grid and predicting bounding boxes and class probabilities for objects directly within each grid cell. This single-pass architecture makes YOLO incredibly fast and efficient, making it suitable for applications like autonomous vehicles, surveillance, and more. However, it may have limitations in detecting small or closely spaced objects compared to other methods. Despite these trade-offs, YOLO remains a widely adopted and essential tool in the computer vision toolbox.Additionally, the seminar highlights a conference paper presenting a YOLO-based approach for human action recognition and localization. The paper addresses an approach to detect, localize and recognize actions of interest in almost real-time from frames obtained by a continuous stream of video data that can be captured from a surveillance camera. The model takes input frames after a specified period and is able to give action label based on a single frame. Combining results over specific time we predicted the action label for the stream of video. We demonstrate that YOLO is effective method and comparatively fast for recognition and localization in Liris Human Activities dataset.

Table of Contents

List of Abbreviations

CNN Convolution Neural Network

DP Discussion between two or more people

ER Enter/leave a room (pass through a door)

EU Try to enter a room (unsuccessfully)

FPS Frames Per Second

GO Give an object to another person

HS Handshaking

LB Luggage left unattended

TC Mobile/ Telephone conversation

TK Keyboard typing

TO Put/take an object into/from a box/desk

UR Unlock and enter (or leave) a room

YOLO You Only Look Once

List of Figures

List of Tables

Chapter 1

Introduction

Over the past few years, there have been significant strides in Computer Vision[8], and Convolutional Neural Networks (CNNs)[9] have been instrumental in transforming image recognition and object detection. Within this domain, a particularly notable advancement is the "You Only Look Once" (YOLO) algorithm, which has revolutionized real-time object detection. Created by Joseph Redmon and Santosh Divvala, YOLO[1] presents a fresh approach to the complexities of object detection, emphasizing a unified framework that achieves both speed and accuracy.

YOLO signifies a fundamental change by presenting a unified model that can predict both bounding boxes and class probabilities simultaneously for numerous objects[7] within an image. In contrast to conventional two-stage detectors that incorporate region proposal networks, YOLO completes object detection in a single pass through the neural network, resulting in exceptional speed and suitability for real-time applications. A leading detection method, Fast R-CNN[14], tends to misidentify background patches as objects due to its limited ability to grasp the larger context. YOLO exhibits fewer background errors, making less than half the number compared to Fast R-CNN

In the ever-evolving landscape of manufacturing, the relentless pursuit of operational excellence hinges on upholding the highest standards of product quality. Central to this system is the YOLO (You Only Look Once) algorithm, a groundbreaking method for object detection. YOLO distinguishes itself by swiftly and accurately identifying objects within an image in a single pass. This efficiency is particularly critical in manufacturing, where real-time detection of defects is vital for maintaining elevated quality standards.

Thanks to its distinctive architecture, the YOLO algorithm facilitates rapid information processing, rendering it an optimal choice for integration into manufacturing facilities and production lines.

1.1 Motivation

In an era dominated by rapid technological advancements, maintaining a keen awareness of the latest developments is paramount. The exploration of "You Only Look Once" (YOLO) extends beyond a surface-level fascination with emerging technologies; it demands a profound curiosity and a commitment to staying ahead in the ever-evolving domain of computer vision. As our digital landscape undergoes continuous transformation, YOLO[4][5][6], an innovative real-time object detection system, distinguishes itself by offering a groundbreaking approach that ensures both efficiency and precision in identifying and categorizing objects within images and videos. Delving into the intricacies of YOLO, understanding its underlying principles, and exploring its practical applications not only provides intellectual stimulation but also aligns seamlessly with the pursuit of knowledge that propels advancements in artificial intelligence.

YOLO (You Only Look Once) emerges as a product of a genuine interest in advancing one's knowledge and skills in computer vision and object detection. This cutting-edge algorithm is fueled by a deep motivation to remain at the forefront of technological progress. YOLO represents a revolutionary method for real-time object detection, facilitating the identification and localization of multiple objects in images or video frames in a single pass. The commitment to mastering YOLO underscores a proactive engagement with the evolving landscape of computer vision, reflecting a dedication to staying informed and adept in the face of ever-changing technological paradigms[5].

As individuals embrace the study of YOLO, they position themselves at the nexus of innovation, contributing to the collective effort to push the boundaries of what is possible in the realm of artificial intelligence. By immersing oneself in the intricacies of YOLO, practitioners not only expand their technical expertise but also contribute to the broader trajectory of advancements that shape the future of computer vision and object detection.

1.2 Background

The You Only Look Once (YOLO) model represents a significant advancement in the field of computer vision and object detection. Conceived by Joseph Redmon and Santosh Divvala, the YOLO model was introduced to the research community through a series of papers, with the seminal work being "You Only Look Once: Unified, Real-Time Object Detection" in 2016.

Prior to YOLO, [5]traditional object detection methods often involved multi-stage processes, where an image was first analyzed to propose potential regions of interest, and subsequently, these regions were classified to identify objects. This two-step approach was computationally intensive and posed challenges in achieving real-time performance, especially for applications requiring rapid and continuous analysis of visual data.

The YOLO model introduced a paradigm shift by unifying the region proposal and classification tasks into a single neural network. This approach significantly streamlined the object detection process, allowing for real-time inference on images and videos. The key innovation of YOLO is its ability to predict bounding box coordinates and class probabilities directly from the entire image in one forward pass through the neural network. This grid-based approach divides the input image into a grid and predicts bounding boxes and class probabilities within each grid cell.

The advantages of YOLO include its simplicity, speed, and efficiency. The model is capable of detecting and classifying multiple objects in an image simultaneously, providing a holistic understanding of the visual scene. YOLO has undergone several iterations, with each version introducing improvements in terms of accuracy, speed, and the ability to handle different object scales and aspect ratios.

Over time, YOLO has become a widely adopted and influential model in computer vision applications, ranging from surveillance and autonomous vehicles to robotics and augmented reality. Its impact is not only seen in the technical advancements it introduced but also in shaping the direction of research in object detection and real-time image analysis. The YOLO model has inspired subsequent models and frameworks, contributing to

the ongoing evolution of object detection methodologies in the field of computer vision.

1.3 Outline

Chapter 1 lays the foundation for this thesis, introducing the study, elucidating the motivation behind delving into YOLO (You Only Look Once), and providing a comprehensive background analysis. Within this chapter, the context is meticulously established for subsequent discussions on YOLO and its diverse applications. As we transition to Chapter 2, an extensive literature survey unfolds, with a specific focus on different research studies using different yolo models. The chapter critically dissects these models, illuminating their respective strengths and weaknesses. A concise summary at the end of the chapter serves as a stepping stone, guiding readers toward a nuanced understanding of the forthcoming YOLO-centric discussions. The core of the thesis, Chapter 3, offers an exhaustive exploration of YOLO. Theoretical explanations, the intricacies of the loss function, and the evolutionary journey of YOLO are thoroughly examined, providing readers with a profound insight into the inner workings of this groundbreaking model. Furthermore, this chapter sheds light on the labeling process, enhancing the reader's comprehension of the fundamental elements that underpin YOLO's functionality. Chapter 4 shifts the focus from theoretical exploration to practical application, centering on the real-time action recognition and labelling . This chapter meticulously dissects a pertinent application paper, offering valuable insights into the implementation and performance of YOLO within this specific context. Finally, Chapter 5 serves as the culmination of the thesis, encapsulating the entire journey with a conclusive summary. This chapter not only recaps the key findings but also provides a holistic perspective, tying together the theoretical foundations, critical analyses, and practical applications discussed throughout the thesis. In doing so, it offers a comprehensive synthesis of the research, inviting readers to reflect on the broader implications and contributions of the study to the field.

Chapter 2

Literature Review

2.1 A Survey on Object Detection

Most object detectors contain two important components: a feature extractor and an object classifier. The feature extractor has rapidly evolved with significant research efforts leading to better deep convolutional architectures. The object classifier, however, has not received much attention and many recent systems (like SPPnet and Fast/Faster R-CNN) use simple multi-layer perceptrons. This paper demonstrates that carefully designing deep networks for object classification is just as important.

This paper introduces Networks on Convolutional feature maps (NoCs), a new type of object classifier that uses shared, region-independent convolutional features. This paper shows that NoCs are more accurate than simple multi-layer perceptrons for object detection. This paper shows that NoCs are a key component of the winning entries in the ImageNet and MS COCO challenges 2015.

NoCs achieve state-of-the-art results on both the PASCAL VOC and COCO datasets. On PASCAL VOC 2010, NoCs achieve a mean average precision (mAP) of 65.6, which is better than the previous state-of-the-art of 60.4. On COCO 2015, NoCs achieve an mAP of 39.0, which is also better than the previous state-of-the-art of 34.8.

2.2 Review On Object Detection Using Yolo

[1]In the field of object detection, the paper "You Only Look Once: Unified, Real-Time Object Detection" presented a groundbreaking approach that transformed real-time detection. Unlike conventional methods that involved distinct stages for region proposal and classification, YOLO integrated these tasks into a singular neural network, leading to markedly increased detection speeds. This groundbreaking method facilitated real-time object detection on edge devices, creating opportunities for applications in robotics, autonomous vehicles, and surveillance systems.

YOLO V1's pivotal innovation is its capacity to execute object detection in a single forward pass through the neural network, eliminating the necessity for region proposals and subsequent refinement steps. This characteristic renders YOLO notably faster than conventional methods, enabling real-time processing of video streams and images. The paper introduced a comprehensive architecture that simultaneously forecasts bounding box coordinates and class probabilities for multiple objects in a grid-based manner.

The architecture of YOLO V1 involves partitioning the input image into a grid and assigning the task of object detection to individual grid cells. Within each cell, predictions are made for bounding box parameters (x, y, width, height) and class probabilities related to the objects present. This grid-centric methodology not only improves the efficiency of the algorithm but also establishes a cohesive framework for the detection of objects.

Redmon's paper not only presented the YOLO algorithm but also conducted a thorough assessment of its performance in comparison to existing methods. The findings demonstrated that YOLO not only exhibited competitive accuracy but also outperformed other approaches in terms of speed, establishing it as the preferred choice for real-time applications.

The YOLO V1 paper has served as a catalyst for numerous iterations and enhancements within the YOLO family, each version leveraging and expanding upon the strengths of its forerunners. The pioneering characteristics of YOLO V1 have enduringly shaped the landscape of computer vision, leaving an indelible mark on subsequent research in object detection and real-time image processing. The comprehensive methodology outlined in the paper has evolved into a standard for developing object detection systems that are both efficient and accurate[27].

YOLOv4 stands as a pinnacle in the realm of object detection algorithms, showcasing its supremacy through remarkable achievements on the MS COCO dataset and other benchmarks. With a staggering 43.5 Average Precision (AP) and an impressive 65.7 AP50, YOLOv4 attains these scores while maintaining a real-time speed of approximately 65 frames per second on the powerful Tesla V100 hardware. This exceptional performance underscores its practical applicability in scenarios requiring swift and accurate object detection.

A key innovation in YOLOv4 is the adoption of the CSPDarkNet53 backbone network. This architecture, derived from Darknet-53, incorporates a novel Cross Stage Partial (CSP) connection, enhancing accuracy while concurrently reducing computational costs. Additionally, YOLOv4 introduces the Mish activation function, an efficient alternative to Swish, along with the CioU loss function, specifically crafted to enhance model convergence and mitigate localization errors. The integration of CmBN (Cross mini-Batch Normalization) further contributes to the model's generalization capabilities, ensuring robust performance across diverse datasets.

Further fortifying its capabilities, YOLOv4 incorporates DropBlock, a state-of-the-art regularization technique designed to bolster the model's resilience against adversarial attacks. These innovations collectively contribute to YOLOv4's impressive results, as evidenced by its state-of-the-art performance on benchmark datasets such as MS COCO, VOC2007, and VOC2012. In particular, YOLOv4 achieves an 84.6 mAP on VOC2007 and an outstanding 89.0 mAP on VOC2012, both accomplished at a commendable speed of 40 frames per second. The amalgamation of speed, accuracy, and innovative architectural enhancements solidifies YOLOv4 as a groundbreaking object detection algorithm, setting new standards in the field of computer vision[6].

In the continuous evolution of YOLO (You Only Look Once), a series of refinements and enhancements have been introduced to elevate its performance and accuracy. These updates encompass subtle yet impactful design changes aimed at optimizing the algorithm. Notably, a new network has been meticulously trained, boasting improved accuracy while maintaining the signature speed that YOLO is renowned for.

Despite the increase in size, the new iteration remains impressively swift. At a resolution of 320 × 320, YOLOv3 demonstrates remarkable efficiency, executing in a mere 22 milliseconds with an impressive mean Average Precision (mAP) of 28.2. This puts it on par with the accuracy of Single Shot Multibox Detector (SSD) while operating three times faster. The commitment to speed and accuracy is further underscored by its performance on the .5 Intersection over Union (IOU) mAP detection metric. YOLOv3 excels, achieving a noteworthy 57.9 AP50 in a mere 51 milliseconds on a Titan X. This is in stark contrast to the 198 milliseconds required by RetinaNet, showcasing a similar level of performance but at a remarkable 3.8 times the speed.

These advancements reaffirm YOLO's position as a cutting-edge object detection algorithm, striking an impressive balance between accuracy and computational efficiency. The commitment to open-source principles is evident, as the entire codebase is made available online, fostering transparency and collaboration within the wider community. The iterative refinement of YOLO reflects a dedication to pushing the boundaries of real-time object detection, making it a versatile and impactful tool in the realm of computer vision[5].

M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge:The paper provides a review of the PASCAL Visual Object Classes (VOC) challenge, an annual competition and workshop that focuses on object classification, detection, segmentation, action classification, and person layout. The authors introduce a number of novel evaluation methods to analyze the performance of submitted algorithms on the VOC datasets. These methods include:A bootstrapping method for determining whether differences in the performance of two algorithms are significant or not A normalized average precision so that performance can be compared across classes with different proportions of positive instances A clustering method for visualizing the performance across multiple algorithms so that the hard and easy images can be identified The use of a joint classifier over the submitted algorithms in order to measure their complementarity and combined performance The authors also analyze the community's progress through time using the methods of Hoiem et al. (2012) to identify the types of occurring errors. They conclude the paper with an appraisal of

the aspects of the challenge that worked well, and those that could be improved in future challenges. Key findings: The PASCAL VOC challenge has been a valuable resource for the computer vision community, providing a standardized dataset, evaluation software, and a forum for researchers to compare their algorithms. The challenge has helped to identify the strengths and weaknesses of the current generation of algorithms, and has led to the development of new methods for object classification, detection, and segmentation. The community's progress on the challenge has been significant, but there are still many challenges that remain[3].

Evaluating the performance of computer vision algorithms is classically done by reporting classification error or accuracy, if the problem at hand is the classification of an object in an image, the recognition of an activity in a video or the categorization and labeling of the image or video. If in addition the detection of an item in an image or a video, and/or its localization are required, frequently used metrics are Recall and Precision, as well as ROC curves. These metrics give quantitative performance values which are easy to understand and to interpret even by non-experts.
However, an inherent problem is the dependency of quantitative performance measures on the quality constraints that we need impose on the detection algorithm. In particular, an important quality parameter of these measures is the spatial or spatio-temporal overlap between a ground-truth item and a detected item, and this needs to be taken into account when interpreting the results. We propose a new performance metric addressing and unifying the qualitative and quantitative aspects of the performance measures. The performance of a detection and recognition algorithm is illustrated intuitively by performance graphs which present quantitative performance values, like Recall, Precision and F-Score, depending on quality constraints of the detection.
In order to compare the performance of different computer vision algorithms, a representative single performance measure is computed from the graphs, by integrating out all quality parameters. The evaluation method can be applied to different types of activity detection and recognition algorithms. The performance metric has been tested on several activity recognition algorithms participating in the ICPR 2012 HARL competition[28].

2.3 Summary

Following an exhaustive examination of the literature surrounding Faster R-CNN and YOLOv1, it is evident that both frameworks have made significant contributions to the progress of computer vision applications. Faster R-CNN, introduced by Shaoqing Ren et al., innovatively incorporates region proposal networks (RPNs), streamlining object detection by unifying region proposal generation and object classification. This integration has resulted in noteworthy enhancements in accuracy and efficiency compared to earlier methodologies.

In contrast, YOLOv1 (You Only Look Once), presented by Joseph Redmon et al., revolutionized object detection by adopting a single-pass approach, eliminating the need for region proposal networks. YOLOv1 achieves real-time object detection by dividing the input image into a grid and directly predicting bounding boxes and class probabilities. While YOLOv1 boasts impressive speed, it may encounter challenges in accurately localizing small objects due to its coarse grid structure.

The literature survey underscores that both Faster R-CNN and YOLOv1 exhibit distinct strengths and weaknesses, and the selection between them hinges on specific application requirements. Faster R-CNN excels in accuracy, making it suitable for scenarios where precision is crucial, albeit at the expense of slightly reduced speed. YOLOv1, with its real-time capabilities, is ideal for applications requiring rapid processing, establishing it as the preferred choice for real-time object detection in certain contexts.

Recent strides in the field have given rise to subsequent versions of both Faster R-CNN and YOLO, each addressing limitations and introducing further optimizations. The choice between these frameworks should consider factors such as speed, accuracy, and the particular demands of the application. Additionally, emerging techniques and hybrid approaches present promising avenues for future exploration in the realm of object detection. The synthesis of findings from the literature survey sets the stage for a comprehensive understanding of the landscape, laying the foundation for informed decisions and potential innovations in object detection methodologies.

Chapter 3

Theoretical Aspects

3.1 Real Time Object Detection Using YOLO

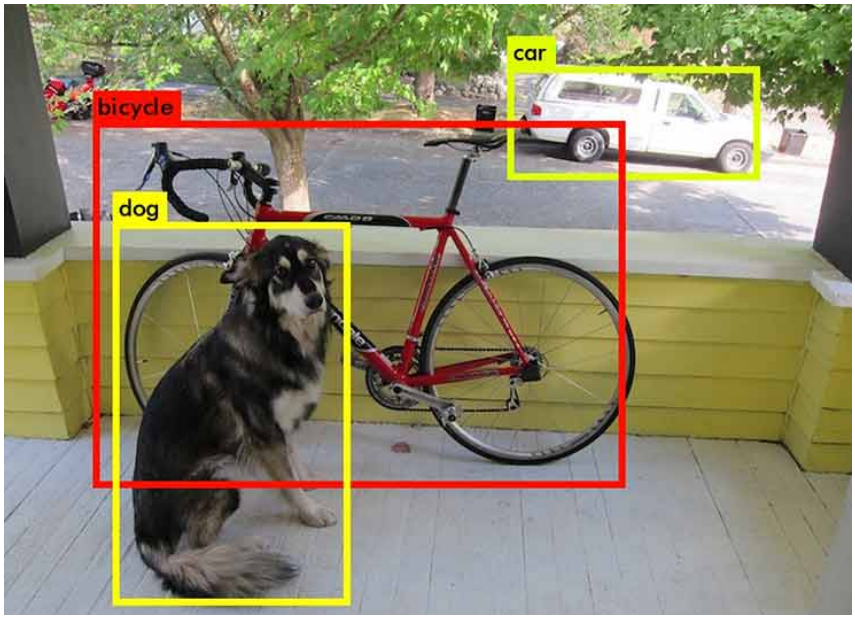


Figure 3.1: Bounding Box Annotation

Addressing the challenge of precisely identifying object locations in images introduces a groundbreaking solution known as "You Only Look Once" (YOLO). This inventive approach surpasses traditional methods by diverging from the conventional approach of analyzing different image segments separately. YOLO, instead, takes in the entire image at once, dividing it into a grid and swiftly predicting the location and identity of objects in a single pass. This enables it to confidently declare, for instance, 'That's a cat, situated

in the bottom-left corner of the picture.' Such capability proves vital for applications like self-driving cars or security cameras, where real-time knowledge of object locations is crucial for swift and accurate decision-making. YOLO's efficiency and speed mark it as a transformative force in the realm of computer vision. The model not only predicts bounding boxes but also assigns class probabilities to each detected object, such as "dog" or "bicycle," along with corresponding probability scores. A high score reflects strong confidence in the prediction, while a lower score may indicate uncertainty or a less robust prediction.

3.2 Theoretical Explanation

Within the YOLO (You Only Look Once) object detection framework, the initial step involves resizing the input image to dimensions of 448x448 pixels. Subsequently, the image is partitioned into a grid of cells, denoted as S[1], with S set to 7. Each of these grid cells takes on the responsibility of predicting the presence of objects within its defined boundaries. When dealing with a specific ground truth object, for instance, one belonging to the person class with bounding box coordinates (200, 311, 142, 250), the target values for prediction are computed in relation to the respective grid cell[1].

$$\Delta x = \frac{200 - 192}{64} \approx 0.13 \tag{3.2.1}$$

$$\Delta y = \frac{311 - 256}{64} \approx 0.87 \tag{3.2.2}$$

$$\Delta w = \frac{142}{448} \approx 0.31 \tag{3.2.3}$$

$$\Delta h = \frac{250}{448} \approx 0.56 \tag{3.2.4}$$

$$(0.113, 0.87, 0.31, 0.56)$$

The proportions for the bounding box's width and height are determined by dividing the actual width and height by the total image size (448). Moreover, the relative values for the x and y coordinates of the bounding box center are calculated by subtracting the

coordinates of the top-left corner of the grid cell containing the object's center from the actual coordinates of the center. Subsequently, these differences are divided by the grid size (S), which, in this instance, is 64 (448/7).

3.2.1 Output

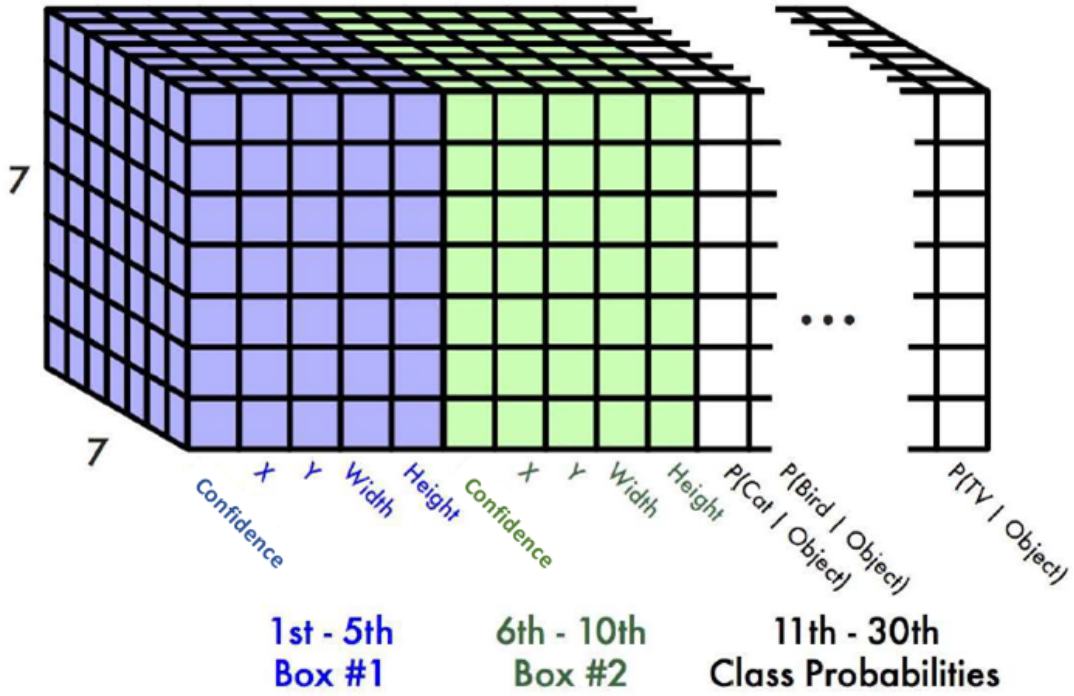


Figure 3.2: Output Layer

This methodology empowers each grid cell to independently forecast a distinct set of target values, enhancing the efficiency and localization of the prediction process. The cell that encompasses the object's center is dedicated to detecting and forecasting the characteristics of that specific object. Through the utilization of relative values, the YOLO framework facilitates effective generalization across various object sizes and positions within the image grid.

In the YOLO architecture, tailored for the [3]PASCAL VOC dataset, each grid cell is configured to predict a set of 5 values: x, y, w, h, and confidence. The grid size, denoted as S, is fixed at 7, resulting in a 7x7 grid of cells. Within each grid cell, the model predicts B bounding boxes, where B is set to 2 in this instance. For every bounding box, the model predicts a confidence score. Additionally, the model predicts C class probabilities, where

C represents the number of labeled classes; for PASCAL VOC, C is 20. Consequently, each grid cell's predictions encompass 2 bounding boxes (B=2), each defined by x, y, w, h, and confidence values. In addition to these, there are 20 class probabilities, summing up to a total of 30 values (2 boxes * 5 values per box + 20 class probabilities) for each grid cell. With a total of 49 grid cells (7x7 grid), the final prediction is encapsulated in a 7x7x30 tensor. This tensor, a multi-dimensional array, contains predictions for each grid cell, encompassing bounding box coordinates, confidence scores, and class probabilities for the specified number of classes. The tensor's structure facilitates efficient object detection across the entire image by localizing and categorizing objects in a grid-based manner.

3.2.2 Architecture

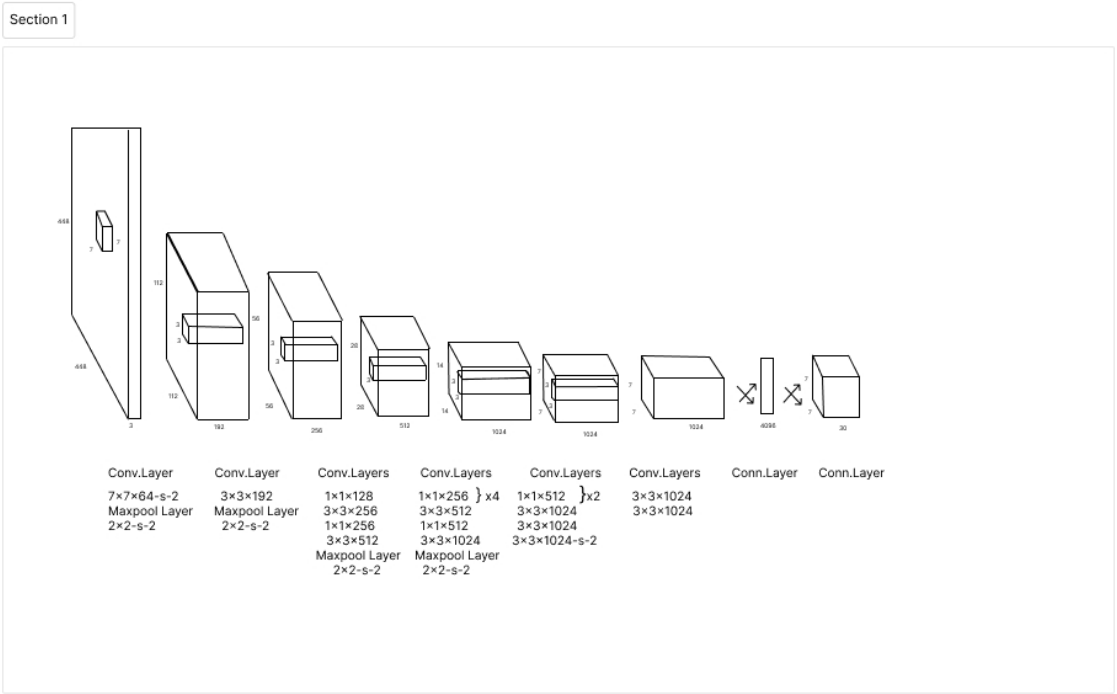


Figure 3.3: Architecture

The structure is fairly straightforward, comprising 24 convolutional layers succeeded by two fully connected layers. The convolutional layers are responsible for extracting features from the input image, while the fully connected layers are tasked with predicting both bounding boxes and class probabilities for the identified objects.

Convolutional Layer:The Convolutional Layer is a fundamental building block in convolutional neural networks (CNNs), widely employed in image and pattern recognition tasks. It consists of learnable filters that convolve across the input data, extracting local patterns and features. These filters, also known as kernels, slide over the input volume, performing element-wise multiplications and aggregating the results to produce feature maps. The convolutional operation enables the network to automatically learn hierarchical representations of the input, capturing spatial hierarchies and reducing the sensitivity to variations in orientation and position. The use of shared weights in the filters promotes parameter sharing, enhancing the model's ability to generalize across the input space. Convolutional layers are pivotal for preserving spatial relationships within the data, allowing the network to recognize complex patterns efficiently while reducing the number of parameters compared to fully connected layers.

Flatten Layer:The Flatten layer is a crucial component in neural network architectures, particularly in transitioning from convolutional layers to fully connected layers. It serves the purpose of converting the multi-dimensional output of the preceding convolutional or pooling layers into a one-dimensional vector. This transformation is essential because fully connected layers require a flat input structure. The Flatten layer essentially unravels the spatial structure of the data, preserving the information contained in each feature map. For instance, if the output from a convolutional layer is a 3D tensor representing spatial features, the Flatten layer reshapes it into a 1D vector, where each element corresponds to a specific feature. This flattened representation allows for the seamless integration of spatial information into subsequent fully connected layers, enabling the network to learn complex hierarchical representations and patterns in the data.

Max Pooling Layer:The Max Pooling layer is a crucial element in convolutional neural networks (CNNs) for downsampling input volumes. Operating with a sliding window mechanism, typically 2x2 or 3x3 in size, the layer selects the maximum value from each window, discarding less relevant information and retaining the most activated features. This downsampling not only reduces the spatial dimensions of the input, making subsequent computations more efficient, but also introduces translation invariance and promotes feature robustness. While max pooling aids in simplifying the network architecture and improving computational efficiency, it's essential to recognize that it discards some

information, and the choice of pooling strategy depends on the specific requirements of the task at hand.

3.2.3 Annotation for Yolo

Labeling is a critical step in preparing a dataset for training the YOLO (You Only Look Once) object detection algorithm. This process involves annotating images with bounding boxes that encapsulate the objects of interest and associating them with corresponding class labels. Each annotated object is represented by a set of normalized coordinates (x, y, width, height), where (x, y) denotes the top-left corner of the bounding box, and (width, height) represent its dimensions. These coordinates are typically scaled to the dimensions of the image, allowing for consistent representation across varied image sizes. The YOLO algorithm relies on a specific label format for each annotated object, which includes the class index and the normalized bounding box coordinates.

Maintaining a consistent class indexing scheme is crucial throughout the dataset. Each class label is assigned a unique index starting from 0, ensuring a standardized representation across the entire dataset. This indexing scheme allows YOLO to distinguish between different object classes during training and inference. The labeled information is then stored in separate text files, usually with a ".txt" extension, associated with each image. Each line in these label files corresponds to one object in the image, containing the class index and normalized coordinates.

The organization of the dataset is equally important. Images and their corresponding label files should be arranged in a structured manner, facilitating easy pairing for training. Tools such as labelImg, RectLabel, or VGG Image Annotator (VIA) streamline the annotation process by providing user-friendly interfaces for drawing bounding boxes and assigning class labels. Once the dataset is labeled, it undergoes preprocessing to ensure that the labels align with the required input format for YOLO. This meticulous labeling process sets the foundation for YOLO's ability to accurately detect and classify objects in diverse scenarios, making it a powerful and widely used object detection algorithm in the field of computer vision.

3.3 Loss Function

The loss function is designed to measure the difference between the predicted bounding boxes and the ground truth bounding boxes. YOLO is an object detection algorithm that divides an input image into a grid and predicts bounding boxes and class probabilities for each grid cell.
The loss function in YOLO v1 is composed of three main components:

Localization Loss (Bounding Box Coordinates): This measures how accurately the model predicts the coordinates of the bounding box for each object. The loss is calculated using the mean squared error between the predicted bounding box coordinates (x, y, width, height) and the ground truth bounding box coordinates.

Confidence Loss: YOLO predicts a confidence score for each bounding box, indicating the probability that the box contains an object. The confidence loss penalizes the model for both false positives (predicting an object where there isn't one) and false negatives (failing to predict an object that exists). It is calculated using the binary cross-entropy loss.

Class Loss: YOLO predicts the probability distribution of the object class for each bounding box. The class loss measures the difference between the predicted class probabilities and the actual class of the object. It is calculated using the categorical cross-entropy loss.

localization loss (L_{coord}), confidence loss (L_{conf}), and class loss (L_{class}). The total loss (L_{total}) is the sum of these three components.

Localization Loss (L_{coord})

The localization loss is calculated using the mean squared error (MSE) between the predicted and ground truth bounding box coordinates. It is given by:

$$L_{\text{coord}} = \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

where S is the number of grid cells, B is the number of bounding boxes, λ_{coord} is a weight

parameter, and $\mathbb{1}_{ij}^{\text{obj}}$ is an indicator function.

Confidence Loss (L_{conf})

The confidence loss penalizes the error in predicting the confidence score and is given by:

$$L_{\text{conf}} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} [(C_i - \hat{C}_i)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2]$$

where C_i is the predicted confidence score, $\hat{C}_i$ is the ground truth confidence score, and λ_{noobj} is a weight parameter.

Class Loss (L_{class})

The class loss penalizes the error in predicting class probabilities and is given by:

$$L_{\text{class}} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \sum_{c=0}^{C} (p_i(c) - \hat{p}_i(c))^2$$

where $p_i(c)$ is the predicted probability of class c and $\hat{p}_i(c)$ is the ground truth probability.

Total Loss (L_{total})

The total loss is the sum of the three components:

$$L_{\text{total}} = L_{\text{coord}} + L_{\text{conf}} + L_{\text{class}}$$

3.4 Evolution of yolo

YOLOv1 (2015): The first version of YOLO, introduced in 2015, was a groundbreaking algorithm that achieved real-time object detection at 45 frames per second (FPS) on a GPU. YOLOv1 divided an image into a 7x7 grid and predicted bounding boxes and class probabilities for each grid cell. YOLOv1 was relatively fast but had lower accuracy than other object detection algorithms at the time.

YOLOv2 (2016): YOLOv2 improved on YOLOv1 in several ways, including: Introducing anchor boxes to improve bounding box prediction accuracy. Adding a batch normalization layer to improve training stability. Using a higher input resolution to improve feature extraction. YOLOv2 achieved 52 FPS with improved accuracy over YOLOv1.

YOLOv3 (2018): YOLOv3 introduced several new features, including: A multi-scale

feature extraction architecture to improve object detection across different scales. A new backbone network, Darknet-53, to improve feature extraction. A new prediction head that used logistic regression to classify objects. YOLOv3 achieved 106 FPS with improved accuracy over YOLOv2.

YOLOv4 (2020): YOLOv4 built on the improvements of YOLOv3 and introduced several new features, including: A new backbone network, CSPDarknet53, that improved feature extraction and reduced computational cost. A new Path Aggregation Network (PANet) to improve feature fusion. A new small object detection head that improved the detection of small objects. YOLOv4 achieved 60 FPS with improved accuracy over YOLOv3.

YOLOv5 (2020): YOLOv5 was developed by Ultralytics and is based on YOLOv4 but with several improvements, including: A new backbone network, PyTorch, that improved ease of use. A new Spatial Pyramid Pooling (SPP) layer that reduced computational cost. A new data augmentation pipeline that improved training stability. YOLOv5 achieved 140 FPS with improved accuracy over YOLOv4.

YOLOv7 (2022): YOLOv7 introduces several new features, including: A new backbone network, Cross Stage Partial Connections (CSP), that improved feature extraction and reduced computational cost. A new neck network, Path Aggregation Network (PANet), that improved feature fusion. A new head network, Anchor-free Head, that improved object detection accuracy.

YOLOv8(2023):YOLOv8 is a new state-of-the-art computer vision model built by Ultralytics, the creators of YOLOv5. The YOLOv8 model contains out-of-the-box support for object detection, classification, and segmentation tasks.Anchor-free architecture: YOLOv8 uses an anchor-free architecture, which makes it easier to train the model on different datasets.It uses a self-attention mechanism in the head of the network, which helps to learn long-range dependencies between features better.

Chapter 4

Research Opportunities And Challenges

4.1 Introduction

Having meticulously dissected the intricacies of the YOLO (You Only Look Once) algorithm and gained a profound understanding of its operational excellence, our focus now shifts towards a practical illustration [2] that vividly underscores its transformative capabilities. In the era of the fourth industrial revolution [11][12][13][15], the pervasive influence of artificial intelligence and deep learning algorithms has significantly elevated efficiency in the domain of product creation. In today's digital landscape, video stands out as the most generated and consumed format, yet paradoxically, it remains one of the least processed.

A noteworthy 2015 report from IHS unveiled that the global daily memory requirement for video surveillance reached a staggering 566 petabytes, with projections anticipating a surge to 2500 petabytes by the end of 2019. The integration of video analytics into the mainstream internet holds the potential to revolutionize applications in surveillance cameras, robotics, content-based video search, and human-computer interfaces.

In recent years, Convolutional Neural Networks (CNNs) have garnered widespread adoption for image classification. As their success in image classification became evident, researchers extended their application to video classification.

The challenge of classifying real-life videos, such as those found in the LIRIS dataset, into arbitrary free-form activities is exacerbated by factors like illumination conditions, occlusion, and intra-class variation.

Current research in this domain typically falls into two categories: human action recognition using two-stream CNNs and human action recognition based on skeleton tracking. While these approaches have demonstrated noteworthy results, they often involve substantial computation time and training overhead. In contrast, our proposed model seeks to recognize and localize actions using fewer video frames, sometimes even a single frame, without relying on optical flow data.

The approach adopted involves assigning a unique action label and confidence score to each frame, determining the global action label for the video sequence by identifying the most frequent action label among periodic frames. This innovative methodology significantly reduces computational time by processing only selected frames, with the flexibility to add more frames in the case of rapidly diminishing confidence scores.

Our project embarks on an exploration of how training You Only Look Once (YOLO) with the LIRIS dataset can provide action labels, confidence scores, and bounding boxes for localized actions. This research endeavors to bridge the gap between theoretical understanding and practical application, offering insights into the efficiency and effectiveness of YOLO in the intricate realm of video analytics, thereby contributing to the broader landscape of artificial intelligence and deep learning applications.

4.2 Case Study

4.2.1 Problems Statement

In the landscape of contemporary research, human action recognition in video analytics has garnered significant attention. Despite the strides made in this domain, prevalent methods often adhere to the practice of assigning a singular action label to an entire video. This assignment is typically accomplished either through a comprehensive analysis of the complete video or by employing a classifier for each frame. This conventional approach diverges from the human vision strategy, which frequently requires only a singular instance of visual data for scene recognition. A notable departure from these methodolo-

gies is evident in the realization that precise action recognition can be achieved with a modest group of frames or even a solitary frame extracted from the video.

[2] This paper presents a novel approach aimed at the nearly real-time detection, localization, and recognition of actions of interest within frames obtained through a continuous stream of video data, such as those captured by surveillance cameras. The proposed model processes input frames at predefined intervals and has the capability to assign an action label based on the information gleaned from a single frame. The innovative aspect lies in the aggregation of results over a specific time duration, ultimately determining the predicted action label for the entire video stream. This methodology not only aligns with the efficiency demanded by real-time applications but also capitalizes on the fact that a temporal sequence of frames contributes significantly to robust action recognition.

To validate the efficacy of our approach, we conducted experiments leveraging the YOLO (You Only Look Once) model, showcasing its effectiveness and speed in recognition and localization on the Liris Human Activities dataset. The results not only highlight the practical feasibility of our proposed method but also underscore the versatility and efficiency of YOLO in handling complex action recognition tasks. This study contributes to the ongoing discourse on human action recognition, offering an innovative approach that combines real-time processing with the nuanced utilization of temporal information for enhanced accuracy[2].

4.2.2 Proposed Methodology

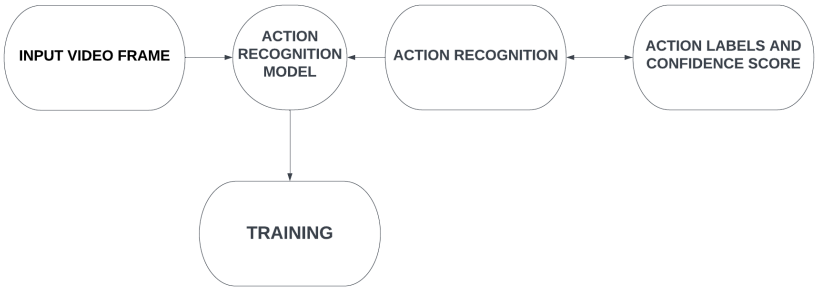


Figure 4.1: Workflow

In recent years, the realm of video analytics has witnessed a surge in comprehensive research, with a specific emphasis on advancing human action recognition. This research is

underpinned by a methodology anchored in the You Only Look Once (YOLO) framework, a pivotal element that will be expounded upon in the subsequent sections. Our model, integral to this exploration, underwent rigorous training utilizing the LIRIS human activities dataset. The training process focused on extracting pertinent frames depicting relevant actions from the dataset's expansive training set, ensuring a targeted and contextually rich learning environment.

To gauge the efficacy of our model, a meticulously designed testing protocol was employed. Specifically, we selected 30 frames within each video for action recognition and localization. The intervals at which these frames were chosen were strategically based on the total number of frames in the video, ensuring a representative and comprehensive evaluation. The paths to these selected frames were meticulously recorded in a text file, serving as the input data for our model. The output, encapsulating detailed information about action recognition and localization for each frame, was methodically stored in a CSV file, facilitating subsequent analyses.

During the intricate process of action detection on individual frames, a definitive action label in the video was considered valid only if it was consistently detected in more than 5 frames, each surpassing a confidence threshold of 0.5, out of the 30 selected frames. This stringent criterion ensured the robustness and reliability of the detected actions. Additionally, the overall confidence score for the action label was computed by averaging all the obtained confidence scores, providing a comprehensive metric to assess the model's certainty in its predictions. This detailed and systematic approach not only establishes a rigorous foundation for evaluating the model's performance but also contributes to the broader landscape of advancements in human action recognition within video analytics.

4.2.3 Dataset

Among all the video dataset present today, most of them belong to:

1) Simple periodic actions like walking, running

2) A more realistic dataset for human-human or human-object interactions

3) Actions in YouTube, Dailymotion videos

4) RGB-D dataset containing depth information in the video

Sr. No	Action Class	Abbreviation	Type of Interaction
1	Discussion between two or more people	DP	Human-Human
2	Give an object to another person	GO	Human-Human-Object
3	Put/take an object into/from a box/desk	TO	Human-Object
4	Enter/leave a room (pass through a door)	ER	-
5	Try to enter a room (unsuccessfully)	EU	-
6	Unlock and enter (or leave) a room	UR	-
7	Luggage left unattended	LB	Human-Object
8	Handshaking	HS	Human-Human
9	Keyboard typing	TK	Human-Object
10	Mobile/Telephone conversation	TC	Human-Object

Table 4.1: Action Classes and Types of Interaction

Figure 4.2: sample data

(i) Try to enter a room (unsuccessfully), (ii) Discussion between two or more people, (iii) Handshaking, (iv)Leave baggage unattended, (v) Give an object to another person, (vi) Unlock and enter (orleave) a room , (vii) Put/take an object into/from a box/desk, (viii) Enter/leave a room (pass through a door), (ix) Telephone conversation.The dataset contains 367 actions from 167 videos. These 167 videos are further divided into 109 training and 58 testing videos. Per class ratio of training to test data is kept approximately 2:1. Figure 4 shows some of the frames in LIRIS dataset. But not all of these datasets give localization of action in the video. As localization of action is important, we are using LIRIS human activities dataset . The dataset contains the videos in form of RGB frames. The annotations provided are in form of an XML file. The dataset is ideal for processing video based on surveillance camera. There are 10 visual annotated actions which include human-human, human-object, and human-human-object interactions.The dataset contains 367 actions from 167 videos. These 167 videos are further divided into 109 training and 58 testing videos. Per class ratio of training to test data is kept approximately 2:1.

4.2.4 Training

The flowchart for training of YOLO based action recognition model is shown in Figure 4.3. First, we converted the LIRIS dataset labels to usable label files for YOLO. YOLO requires a .txt file for each frame with a line for each action in the frame. Further YOLO requires some files to start training which are:

- Total number of action classes.
- Text file with the path to all frames which we want to train.
- Text file with names of all action classes.
- The path to save trained weight files.
- A configuration file with all layers of YOLO architecture.
- Pre-trained convolutional weights.

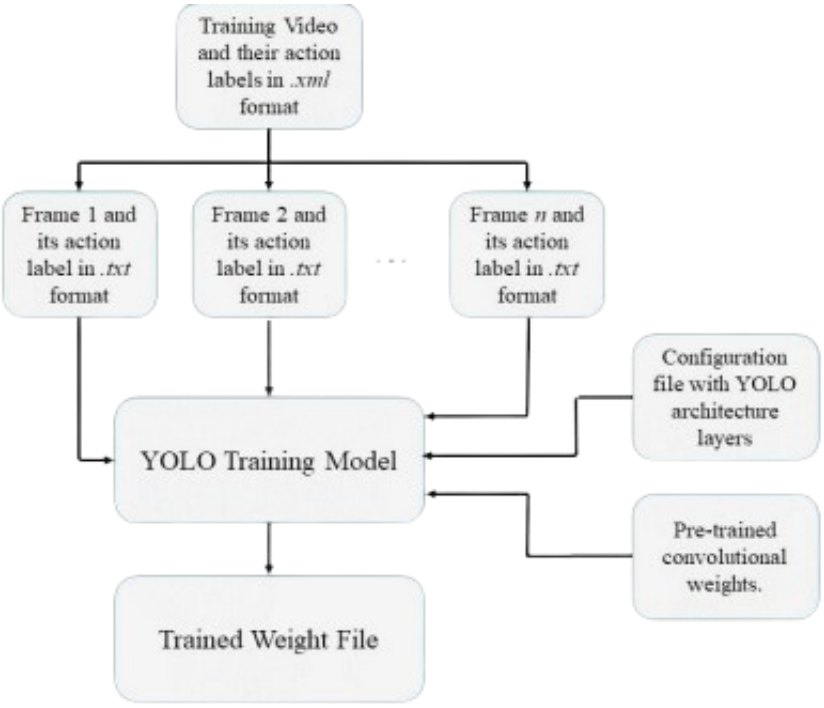


Figure 4.3: training yolo

4.2.5 Advantages over other model

The landscape of object detection in computer vision has seen a proliferation of Convolutional Neural Network (CNN)-based systems, many of which reuse classifiers or localizers to detect objects within images. These models typically apply a two-step method involving classification and localization at multiple locations and scales across an image. Regions with high confidence scores are then considered as potential detections. In contrast, You Only Look Once (YOLO) presents a paradigm shift in object detection, leveraging a single CNN for both classification and localization. This streamlined approach is not only conceptually elegant but also offers practical advantages[2].

One of the primary advantages of YOLO is its efficiency in processing images. Capable of achieving speeds between 40-90 frames per second (FPS), YOLO stands out as remarkably fast. This high processing speed translates to the ability to process streaming video in real-time with negligible latency, often in just a few milliseconds. This speed is particularly noteworthy when compared to traditional methods, as YOLO's architecture enables it to be 1000 times faster than R-CNN and 100 times faster than fast R-CNN.

Moving on to the experimental evaluation of the proposed real-time packaging defect detection system based on the YOLO algorithm, we conducted extensive testing using the LIRIS dataset, which encompasses images capturing different actions in real-life scenarios. During the training process, each iteration demanded approximately 5.25 seconds of computational effort. The model underwent training for a substantial 40,000 iterations, resulting in an encouraging average loss of 0.32. This was achieved with a batch size of 32 and 8 subdivisions, underscoring the model's ability to efficiently learn and generalize from the provided dataset.

The results obtained from our experimentation emphasize the efficacy of the YOLO algorithm in real-world scenarios. Its speed and accuracy in detecting packaging defects demonstrate its practical applicability in industrial settings where swift and accurate defect detection is paramount. The YOLO algorithm's ability to process images in real-time with minimal latency positions it as a powerful tool in applications ranging from quality control in manufacturing to surveillance and security.

In conclusion, the YOLO algorithm's innovative architecture, combining classification and localization in a single pass, represents a breakthrough in object detection. Its superior processing speed, demonstrated by its real-time capabilities, makes it a formidable contender in various domains requiring rapid and accurate visual analysis. The experimental results affirm the effectiveness of YOLO in real-time packaging defect detection, paving the way for its broader application in diverse industries.

4.3 Result And Discussion

To augment accuracy and further diminish the average loss, we recommend an extended training duration involving an even greater number of iterations. Notably, during the testing phase, our model exhibited an impressive average recognition time of about 61 milliseconds for actions within a frame, translating to a commendable 15-16 frames per second (FPS). This underscores the practicality of real-time action recognition on standard desktop PCs, leveraging a mid-level Graphics Processing Unit (GPU).The crux of our evaluation lies in the accuracy obtained on the test dataset, serving as the primary metric for classification in our paper. Leveraging a confusion matrix, we meticulously

label	precision	recall	f-score
DP	77.778	100	87.5
GO	71.429	71.429	71.429
TO	83.333	90.909	86.956
ER	85.714	100	92.307
EU	100	77.778	87.5
UR	91.667	100	95.652
LB	100	85.714	92.307
HS	88.889	80	84.211
TK	100	100	100
TC	100	75	85.714
mean	89.881	88.083	88.358

Table 4.2: Results

calculated Precision, Recall, F-score, and overall accuracy. The proposed methodology demonstrated remarkable efficacy, achieving an impressive accuracy rate of 88.372.
Diving deeper into the performance metrics of our trained YOLOv5s model, which underwent a rigorous 150-epoch training regimen, we observed promising outcomes. The model notched up a precision score of 89.8, an accuracy rate of 88.5, a recall rate of 88.03, and an F-score of 88.358. These metrics collectively signal a praiseworthy performance in the model's ability to accurately identify and classify diverse action classes. It's crucial to note that our proposed approach hinges on the assumption that the actions occur aproximally to the camera and are relatively unobscured.

4.3.1 Confusion Matrix

The accuracy obtained on the test dataset is the primary evaluation metric for classification in our paper. The confusion matrix is used to calculate Precision, Recall, F-score and overall Accuracy of the model. Confusion matrix obtained on test dataset is shown in Table 5.2, where rows represent actual actions while column represents the predicted

	DP	GO	TO	ER	EU	UR	LB	HS	TK	TC
DP	7	0	0	0	0	0	0	0	0	0
GO	1	5	0	0	0	0	0	1	0	0
TO	0	1	10	0	0	0	0	0	0	0
ER	0	0	0	12	0	0	0	0	0	0
EU	0	0	0	1	7	1	0	0	0	0
UR	0	0	0	0	0	11	0	0	0	0
LB	1	0	0	0	0	0	6	0	0	0
HS	0	1	0	1	0	0	0	8	0	0
TK	0	0	0	0	0	0	0	0	4	0
TC	0	0	2	0	0	0	0	0	0	6

Table 4.3: Confusion Matrix

actions.The provided confusion matrix offers a detailed overview of the performance of a classification model across different action classes. Each row represents the predicted classes, while each column represents the actual (ground truth) classes. Taking "Discussion between two or more people" (DP) as an example, the matrix reveals that 7 instances were correctly classified as DP, 1 instance was incorrectly classified as "Give an object to another person" (GO), and 1 instance was misclassified as "Luggage left unattended" (LB). Similar interpretations can be made for other action classes. Notably, the diagonal elements represent correct classifications, while off-diagonal elements indicate misclassifications. For instance, "Enter/leave a room" (ER) has all 12 instances correctly classified.

4.4 Challenges

The project encounters multifaceted challenges that necessitate careful consideration for its successful implementation. Firstly, the inherent temporal dynamics of human actions pose a significant hurdle. Recognizing actions from a single frame or a limited frame set may overlook the temporal context crucial for certain actions. Striking a balance between real-time efficiency and capturing the temporal evolution of actions is a complex challenge that demands innovative solutions.

Secondly, the model's assumption that actions occur near the camera and involve minimal occlusion introduces challenges in handling real-world scenarios with varied perspectives and occlusion levels. Adapting the model to accurately localize and recognize actions across diverse spatial configurations is paramount. Overcoming the limitations of assuming proximity and minimal occlusion is crucial for ensuring the model's robustness and reliability in practical applications.

Thirdly, achieving generalization across environments remains a formidable challenge. The project's performance is demonstrated on a specific dataset, and its ability to adapt to diverse real-world environments, characterized by variations in lighting, backgrounds, and settings, is critical. Ensuring that the model remains effective and reliable across different scenarios is essential for its practical utility and broader applicability. Tackling these challenges demands a comprehensive approach that encompasses model refinement, rigorous testing in diverse contexts, and considerations for ethical implications to ensure responsible deployment in real-world scenarios.

Chapter 5

Conclusion

In conclusion, our project marks a significant departure from the constraints inherent in traditional human action recognition methods within video analytics. While prevailing approaches often revolve around assigning a single action label to an entire video or relying on frame-by-frame classifiers, our methodology endeavors to emulate the efficiency ingrained in human visual recognition. Recognizing the human capacity to discern scenes with just a glimpse of visual data, our approach prioritizes the precision achievable through a small group of frames or even a singular frame extracted from a continuous video stream.

Central to our methodology is the real-time detection, localization, and recognition of actions of interest captured by surveillance cameras. Diverging from conventional methods, our model processes input frames at specified intervals, furnishing action labels based on individual frames. By aggregating results over distinct time intervals, we showcase the capability to predict action labels effectively for the entire video stream.

Our extensive experimentation with the YOLO model serves as a compelling testament to its effectiveness and comparative speed in action recognition and localization, prominently demonstrated on the rigorous Liris Human Activities dataset. These findings underscore the substantial potential of our approach for real-time applications, charting a promising trajectory for amplifying the efficiency and accuracy of human action recognition within the realm of video analytics.

Furthermore, our project not only addresses current limitations but also contributes meaningfully to the ongoing advancements in the field. It offers a nuanced and efficient approach

to video-based human action recognition with far-reaching implications for the broader landscape of intelligent systems. By challenging the conventional norms and pushing the boundaries of real-time recognition, our work opens up new avenues for research and application, providing valuable insights that can influence the trajectory of future developments in the domain of computer vision and artificial intelligence.

In essence, our project stands at the intersection of innovation and practicality, offering a transformative perspective on human action recognition. As technological landscapes evolve, our approach provides a glimpse into the potential of streamlined, real-time recognition systems, contributing to the ongoing narrative of progress in the field.tion recognition with far-reaching implications for the broader landscape of intelligent systems.

5.1 Future Scope

The forward-looking potential of this project extends far beyond its current scope, presenting a roadmap for the evolution of human action recognition within the dynamic realm of video analytics. The realization that accurate recognition can be achieved with minimal frames or even a single frame opens the door to nuanced refinements and expansive applications. Future research could delve into intricate model parameter fine-tuning, optimization of processing pipelines for heightened efficiency, and exploration of cutting-edge sensor technologies, such as depth or multispectral cameras, to bolster recognition capabilities across diverse environmental conditions.

The project's emphasis on real-time applications propels the potential for deployment on edge devices, introducing a transformative dimension to scenarios where immediate action is paramount, particularly in surveillance and security applications. Furthermore, extending the proposed methodology to encompass diverse datasets, integrating it into intricate human-computer interaction scenarios, and elevating model interpretability represent promising trajectories for advanced exploration and innovation.

Beyond the visual domain, the project opens avenues for exploration in multi-modal recognition, ushering in a comprehensive approach that combines visual information with other modalities like audio or sensor data. This holistic strategy holds the promise of significantly enhancing the model's robustness and accuracy, especially when navigating

complex and challenging scenarios. In essence, this project serves as a cornerstone for ongoing research endeavors, providing a sophisticated and efficient framework for video-based human action recognition with profound implications for the evolution of intelligent systems in real-world applications.
The far-reaching impact of this project is poised to resonate across diverse domains, contributing not only to the refinement of human action recognition technologies but also influencing the broader landscape of artificial intelligence and its integration into our daily lives. As technological advancements continue, this project stands as a beacon guiding the way toward a future where intelligent systems seamlessly understand and respond to human actions in real time.

5.2 Application

Surveillance Systems: Real-time human action recognition can enhance surveillance systems by automatically identifying and alerting security personnel to suspicious activities. Monitoring crowded areas, public spaces, or critical infrastructure for abnormal behavior.

Autonomous Vehicles: Integrating human action recognition in the perception systems of autonomous vehicles to enhance safety. Identifying pedestrian actions, such as crossing the road, to facilitate safe navigation.

Retail Analytics: Analyzing customer behavior in retail environments to understand shopping patterns. Monitoring and improving store layouts based on customer movement and interactions.

Human-Computer Interaction: Enabling natural and intuitive interactions between humans and computers. Gestural control for devices or systems based on recognized human actions.

Healthcare Monitoring: Monitoring patient activities in healthcare settings to ensure adherence to prescribed exercises or movements. Assisting in elderly care by detecting

falls or unusual movements.

Sports Analytics: Analyzing and tracking athlete movements during training or competitions. Providing insights into player strategies and performance in sports analytics.

Gesture Recognition: Implementing gesture-based interfaces in augmented reality (AR) or virtual reality (VR) applications. Recognizing specific gestures for controlling devices or interacting with virtual environments.

Quality Control in Manufacturing: Monitoring and ensuring the correctness of human actions in manufacturing processes. Identifying deviations from standard operating procedures for quality control.

Education and Training: Enhancing virtual training scenarios by providing real-time feedback on user actions. Creating interactive educational content with adaptive feedback based on recognized actions.

Smart Homes: Integrating human action recognition for security and automation in smart home systems. Recognizing and responding to specific actions or gestures for home automation.

Entertainment and Gaming: Creating interactive and immersive gaming experiences with real-time recognition of player actions. Incorporating human action recognition for character animations and responses in virtual environments.

Crowd Management: Monitoring and managing crowd movements in public events or transportation hubs. Ensuring public safety through the early detection of unusual or potentially dangerous behaviors.

5.3 Limitations

While the presented project has made significant strides in real-time human action recognition, it confronts several inherent limitations that merit thoughtful consideration. Chief among these concerns is the reliance on the Liris Human Activities dataset, which introduces a potential challenge regarding the model's generalizability across diverse scenarios. The limited diversity within the dataset raises questions about the model's adaptability to a broader range of real-world environments, possibly impacting its reliability when confronted with unforeseen contexts.

Another critical limitation surfaces in the project's assumption that actions predominantly occur near the camera and involve minimal occlusion. Real-world scenarios often feature actions unfolding at varying distances and may be subject to partial or complete occlusion, posing challenges to the model's robustness in handling such complexities. This limitation underscores the necessity for a more nuanced approach that accommodates diverse spatial configurations and considers the impact of occlusion on action recognition.

Additionally, the project's emphasis on recognizing actions primarily from a single frame or a small set of frames introduces potential challenges in capturing temporal context. Certain actions inherently demand a temporal understanding that extends beyond individual frames. Neglecting to account for this temporal nuance could result in misinterpretations, particularly in more complex and dynamic scenarios. A prospective enhancement to address this limitation involves incorporating temporal considerations into the model architecture, allowing for a more comprehensive understanding of actions over time.

To mitigate these limitations and bolster the project's utility and reliability in real-world applications, it is imperative to explore and incorporate diverse datasets that better capture the variability of human actions in different settings. Additionally, refining the spatial considerations to accommodate actions at various distances and improving the model's ability to handle occlusion scenarios will enhance its robustness. Furthermore, incorporating temporal elements into the recognition process, perhaps through the exploration

of video-based datasets, will enable the model to better understand and interpret actions in dynamic contexts.

In essence, acknowledging and addressing these limitations is pivotal for the continued advancement of the project. By embracing a more comprehensive and adaptive approach that considers diverse scenarios, spatial intricacies, and temporal dynamics, the project can elevate its efficacy and applicability, making significant strides towards achieving reliable real-time human action recognition across a spectrum of real-world scenarios.

References

[1] You Only Look Once: Unified, Real-Time Object Detection Joseph Redmon , Santosh Divvala, Ross Girshick , Ali Farhadi,9 May 2016

[2] YOLO based Human Action Recognition and Localization, S. Shindea, A. Kotharia,V. Guptab,International Conference on Robotics and Smart Manufacturing (RoSMa),2019

[3] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, 111(1):98–136, Jan. 2015

[4] Fast YOLO: Real-Time Object Detection on CPU by Alexey Bochkovych and Andrew Mikolov (2017)

[5] YOLOv3: An Improved Version of You Only Look Once Object Detection by Joseph Redmon and Ali Farhadi (2018)

[6] YOLOv4: Optimal Speed and Accuracy of Object Detection by Alexey Bochkovych, Chien-Yao Wang, and Hong-Yang Lee (2020)

[7] YOLO Object Detection Explained-https://www.datacamp.com/blog/yolo-object-detection-explained

[8] Computer Vision and Image Processing: A Paper Review February 2018 International Journal of Artificial Intelligence Research 2(1):22 Victor - Wiley Thomas - Lucas

[9] An Analysis Of Convolutional Neural Networks For Image Classification,panelNeha Sharma, Vibhor Jain, Anju Mishra

[10] F. Psarommatis, G. May, P. A. Dreyfus, and D. Kiritsis, "Zero defect manufacturing: state-of-the-art review, shortcomings and future directions in research," Taylor Fr. Online, vol. 58, no. 1, pp. 1–17, Jan. 2019, doi: 10.1080/00207543.2019.1605228

[11] J. F. Arinez, Q. Chang, R. X. Gao, C. Xu, and J. Zhang, "Artificial Intelligence in Advanced Manufacturing: Current Status and Future Outlook," J. Manuf. Sci. Eng. Trans. ASME, vol. 142, no. 11, Nov. 2020, doi: 10.1115/1.4047855/1085487.

[12] R. S. Peres, X. Jia, J. Lee, K. Sun, A. W. Colombo, and J. Barata, "Industrial Artificial Intelligence in Industry 4.0 -Systematic Review,Challenges and Outlook," IEEE Access, 2020, doi: 10.1109/ACCESS.2020.3042874.

[13] M. Ghobakhloo, "The future of manufacturing industry: a strategic roadmap toward Industry 4.0," J. Manuf. Technol. Manag., vol. 29, no. 6, pp. 910–936, Oct. 2018, doi: 10.1108/JMTM-02-2018-0057.

[14] R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015

[15] K. Zhou, T. Liu, and L. Zhou, "Industry 4.0: Towards future industrial opportunities and challenges," in 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2015, Jan. 2016, pp. 2147–2152, doi: 10.1109/FSKD.2015.7382284

[16] FFmpeg, Open-source multimedia framework." https://ffmpeg.org/ (accessed Apr. 15, 2022)

[17] D. Mishkin. Models accuracy on imagenet 2012 val. https://github.com/BVLC/caffe/wiki/Models-accuracy-on-ImageNet-2012-val. Accessed: 2015-10-2.

[18] J. Redmon. Darknet: Open source neural networks in c. http://pjreddie.com/darknet/, 2013–2016. 3

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV), 2015.

[20] S. Ren, K. He, R. B. Girshick, X. Zhang, and J. Sun. Object detection networks on convolutional feature maps. CoRR, abs/1504.06066, 2015.

[21] Object Classification Using Spectral Images and Deep,Carlos López; Román Jácome; Hans Garcia; Henry Arguello

[22] An Introduction to Convolutional Neural Networks Keiron O'Shea, Ryan Nash,26 November 2015

[23] Object classification with deep convolutional neural network using spatial information:Ryusei Shima; He Yunan; Osamu Fukuda; Hiroshi Okumura; Kohei Arai; Nan Bu

[24] A Review of Yolo Algorithm Developments Author links open overlay panelPeiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, Bo Ma

[25] Yolo V1 -https://youtu.be/zgbPj4lSc58?si=2JDmj8LqWApYksG7

[26] "Data Generated by new survellillance cameras to increase exponentially in the coming years." [Online,Accessed on Mar 12, 2018]. http://www.securityinfowatch.com/news/12160483/data-generated-by-new-surveillance-cameras-to-increase-exponentially-in-the-comingyears

[27] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 779-788. doi: 10.1109/CVPR.2016.91

[28] C Wolf, J. Mille, E. Lombardi, O. Celiktutan, M. Jiu, E. Dogan, G. Eren, M. Baccouche, E. Dellandrea, C.-E. Bichot, C. Garcia, B. Sankur, Evaluation of video activity localizations integrating quality and quantity measurements, In Computer Vision and Image Understanding (127):14-30, 2014.

[29] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In CVPR, pages 4305– 4314, 2015.

[30] Simonyan, Karen Zisserman, Andrew. (2014). Two-Stream Convolutional Networks for Action Recognition in Videos. Advances in Neural Information Processing Systems. 1.

[31] Papadopoulos G.T., Axenopoulos A., Daras P. (2014) Real-Time Skeleton-Tracking-Based Human Action Recognition Using Kinect Data. In: Gurrin C., Hopfgartner F., Hurst W., Johansen H., Lee H., O'Connor N. (eds) MultiMedia Modeling. MMM 2014. Lecture Notes in Computer Science, vol 8325. Springer, Cham.

[32] "YOLO-You only look once, real time object detection explained." [Online, Accessed on Mar 12, 2018]. https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained- 492dc9230006