ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ

Αναγνώριση καταστροφών σε κτίρια με χρήση εναέριων μη επανδρωμένων οχημάτων

Ντάλλας Γεώργιος

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

ΥΠΕΥΘΥΝΟΣ

Κολομβάτσος Κωνσταντίνος

Επίκουρος Καθηγητής

Λαμία …… 2024

SCHOOL OF SCIENCE

DEPARTMENT OF COMPUTER SCIENCE & TELECOMMUNICATIONS

Damage assessment in buildings using aerial drones

George Ntallas

FINAL THESIS

ADVISOR

Dr Konstantinos Kolomvatsos, BSc MSc PhD
Assistant professor in Intelligent Systems for Pervasive Computing

Lamia …… 2024

«Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις [(1)], που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

*1. Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.*

*2. Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.*

3. Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια

4. Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.

Ημερομηνία: ……/…../20……

Ο – Η Δηλ.

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση
του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων
σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.»

ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία διερευνά την αποτελεσματικότητα της χρήσης ενός μοντέλου YOLOv7 για την ανίχνευση ρωγμών σε πραγματικό χρόνο σε τοίχους με τη χρήση εναέριων μη επανδρωμένων οχημάτων (UAV). Ξεκινάμε με τη διερεύνηση των σχετικών μοντέλων βαθιάς μάθησης και της αρχιτεκτονικής YOLOv7, εστιάζοντας στην καταλληλότητά της για εργασίες ανίχνευσης αντικειμένων. Στη συνέχεια προτείνουμε μια νέα προσέγγιση που περιλαμβάνει τη λεπτομερή ρύθμιση του μοντέλου YOLOv7 σε ένα επιμελημένο σύνολο δεδομένων με διάφορους τύπους ρωγμών τοίχων που έχουν καταγραφεί από UAV. Παρέχονται λεπτομέρειες σχετικά με τη διαδικασία τελειοποίησης, την ενσωμάτωση με πλατφόρμες UAV και την υλοποίηση της ανίχνευσης σε πραγματικό χρόνο. Πραγματοποιείται αξιολόγηση επιδόσεων με τη χρήση των μετρικών precision, recall και Mean Average Precision (mAP) σε αθέατα δεδομένα βίντεο UAV. Τα αποτελέσματα καταδεικνύουν την αποτελεσματικότητα του λεπτομερώς ρυθμισμένου μοντέλου YOLOv7 στην επίτευξη ακριβούς ανίχνευσης ρωγμών σε πραγματικό χρόνο. Συζητάμε τους περιορισμούς και τις προκλήσεις που αντιμετωπίστηκαν, περιγράφοντας μελλοντικές ερευνητικές κατευθύνσεις για την περαιτέρω ενίσχυση της ανίχνευσης ρωγμών σε πραγματικό χρόνο με τη χρήση του YOLOv7 και των UAV.

ABSTRACT

This work examines the effectiveness of employing UAVs to detect cracks in walls in real time using a YOLOv7 model. We first investigate the YOLOv7 architecture and related deep learning models, emphasizing how well-suited it is for object detection tasks. The new method we suggest then involves fine-tuning the YOLOv7 model using a carefully selected dataset of various wall fracture kinds that were photographed by UAVs. The development of real-time detection, integration with UAV platforms, and fine-tuning procedures are described in detail. On unseen UAV video data, performance evaluation is carried out utilizing precision, recall, and Mean Average Precision (mAP) metrics. The outcomes show how well the adjusted YOLOv7 model performs in terms of producing precise real-time crack detection. We talk about the drawbacks and difficulties we had, and we outline future research directions to improve real-time crack detection using YOLOv7 and UAVs.

Table of Content

ΠΕΡΙΛΗΨΗ 2
ABSTRACT 3

TABLE OF CONTENT 4

1) INTRODUCTION 6

2) INTODUCTION TO DEEP LEARNING 8

2.1) ARTIFICIAL INTELLIGENCE 8
2.2) MACHINE LEARNING 9
2.2.A) TYPE OF MODELS 9
2.2.B) APPLICATIONS 10
2.2.C) ETHICAL CONSIDERATION 11
2.3) DEAP LEARNING 12
2.3.A) DEFINITIONS 12
2.3.B) DEEP LEARNING CLASSES 13
2.3.C) NEURAL NETWORKS 14
2.4) OBJECT DETECTION MODELS 19
2.5) APPLICATION IN DAMAGE RECOGNITION 22

3) PROPOSED MODEL 24

3.1) YOLO SERIES 24
3.2) OVERVIEW OF YOLOv7 MODEL 26
3.3) DESIGN ARCHITECTURE 29
3.3.A) LAYER AGGREGATION NETWORKS 29
3.3.B) MODEL SCALING 29
3.4) TRAINING PROCESS 32
3.5) INTEGRATION WITH UAVs 33
2.5.A) OBJECT DETECTION WITH UAVs 33
2.5.B) INTEGRATION OF YOLOv7 WITH UAV 35
3.6) RELATED WORKS 39

4) EXPERIMENTAL EVALUATION 41

4.1) DATASET DESCRIPTION 41
4.2) IMAGE AUGMENTATION 43
4.3) RASPBERRY PI 43
4.4) EVALUATION METRICS 45
4.5) RESULTS AND ANALYSIS 47

5) CONCLUSIONS AND FUTURE EXTENSIONS 54

BIBLIOGRAPHY..... 55

1) INTRODUCTION

Unmanned aerial vehicles, or UAVs, have become indispensable instruments in a number of domains, including disaster relief. Rapid aerial reconnaissance of impacted regions provides unmatched benefits for damage assessment and relied operation management. Effective catastrophe identification is still a difficult undertaking, especially when in comes to determining structural damage in structures. Large-scale disaster can make traditional procedures unfeasible and time-consuming since the frequently rely on manual inspection or scant sensor data. In addition, deep learning's quick development has created new opportunities for automating difficult jobs, which makes it a desirable strategy for improving disaster response capacities.

In emergencies, it's key to identify building damage swiftly and accurately. This sharpens our rescue actions and aids resource allocation. Yet, old methods to detect this harm don't always measure up. For this issue, fresh techniques involving high-end tools like solid algorithms and drones are needed. These apparatus can heighten precision and efficacy in aid movements during disasters. Blending high-tech learning systems with drones isn't always simple. The systems have to be robust, adapting to varied structures and surroundings, plus they require real-time functioning. The demand for studies on innovating new methods narrows the gap between technology and their application in managing disasters.

This thesis aims to make a new plan for spotting building damages using UAVs with cutting-edge deep learning models. Here's what the study plans on:

a) Check out YOLOv7, a top deep learning design, to see if it can spot building damage from aerial pictures.

b) Build a one-of-a-kind model to spot damage. This needs to think about the special parts and hurdles of inspecting buildings.

c) Run tests to check how good, fast, and sturdy the new plan is.

d) Share ideas on where could the plan be used and its limitations. It also talks about where future studies and upgrades can be made.

2) INTODUCTION TO DEEP LEARNING

2.1) ARTIFICIAL INTELLIGENCE

Making machines as intelligent as the human brain is known as artificial intelligence, or AI. Artificial intelligence in computer science refers to the study of "intelligent agents," or any device that senses its surroundings and makes decisions that increase the likelihood that it will succeed in reaching its objectives. When a machine is able to carry out tasks that people associate with other human minds, like "learning" and "problem solving", it is referred to as "artificial intelligence".[16] A common definition is: a technology that allows machines to mimic a variety of specific human abilities. It is hardly unexpected that defining AI is so challenging. After all, it's really a simulation or imitation of human intelligence, which we still have difficulty fully understanding. But a fixed definition that AI HLEG provides is “systems that display intelligent behaviour by analyzing their environment and taking actions – with some degree of autonomy – to achieve specific goals” [17]

One area of AI is machine learning, considering that machine’s ability to learn is essential. Significant attempts have been made to develop machine learning during the past couple decades. As a result, people have greater standards for machines. One effort in this approach is the subset of machine learning, the deep learning. Deep is the term which refers to a number of layers in a neural network. The deep network has more than one hidden layer whereas a shallow network has only one. [16]

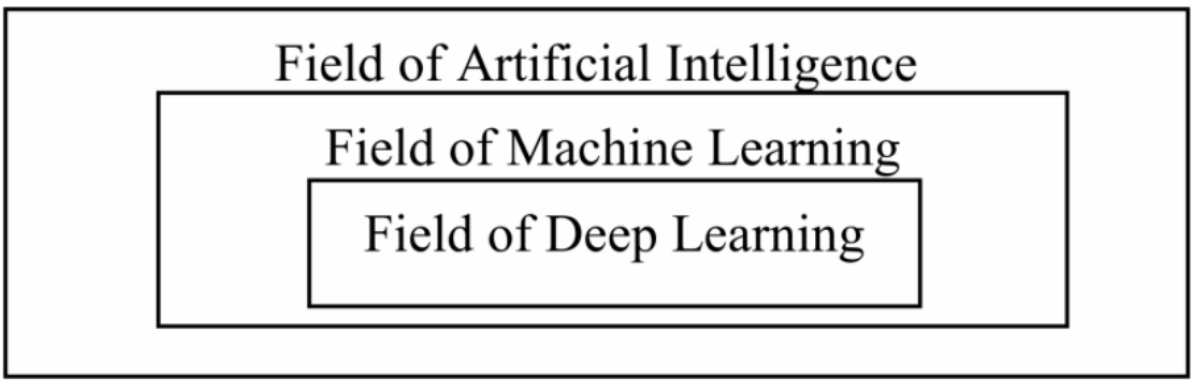

Figure 1. Sub-fields of Artificial Intelligence [16]

2.2) MACHINE LEARNING

2.2.a) TYPE OF MODELS

The field of computer science known as "machine learning" makes programmatic learning possible. Learning can be classified into four categories: supervised, unsupervised, semi-supervised and reinforcement learning. In order to set the parameters of a machine learning model, must first supply the training data and the methodology [23, 24].There are various types of machine learning models such as:

a) Autoregressive Integrated moving average (ARIMA) [18]
b) Linear Regression [19]
c) Logistic Regression (LR) [20]
d) Decision Tree (DT) [21]
e) Random Forest (RF) [21]
f) Support Vector Machine (SVM) [22]
g) k-Nearest Neighbor (kNN) [21]

Using input samples and labels, supervised learning trains the machine learning algorithm. As much as is practical, the model matches the function y = f(x). Supervised learning involves teaching the algorithm with a training dataset that has the appropriate labels [23]. The categories for supervised learning algorithms are:

a) Regression: is determined by the variable that represents the output. If the output variable is continuous, then the job in question is referred to as a regression task.
b) Classification: Categorical variables like color and form are used in classification tasks. Supervised learning is used in the majority of machine learning applications. Random forests, SVMs, logistic regression, and linear regression are examples of supervised learning techniques.

In unsupervised learning, we use only the input examples, not the output labels, to train the machine learning algorithm. The program looks for patterns by attempting to understand the underlying structure of the input instances. Two tasks,clustering and association, can be used to further characterize unsupervised learning methods. [23] The semi-supervised is a mix of both supervised and unsupervised learning. This type of algorithm is involving both labeled and unlabeled data where the training process starts with labeled and then with unlabeled data. On the other hand, the reinforcement learning involve an agent

learns how to communicate with its surroundings by carrying out an action, assessing the reward or penalty it receives, and updating its state. The agent makes use of these rewards to figure out the optimal sequence of steps to complete a task. [24]

2.2.b) APPLICATIONS

Machine learning applications come in a variety of application areas and subdomains. These applications are: computer vision, prediction, semantic analysis, natural language processing and information retrieval. [16]

- Computer Vision: The subdomains of the Computer Vision are: object recognition, object detection, and object processing.
- Prediction: Prediction has also subdomains: Classification, analysis and recommendation.
- Semantic Analysis: Semantic Analysis is the technique of connecting syntactic structures from paragraphs, phrases, and words
- Natural language processing: Computing programs computers to accurately handle natural language data through a process called natural language processing.
- Information Retrieval: Information retrieval is the science of finding information within a document, finding documents, finding metadata that describes the material, and finding databases of sounds and pictures

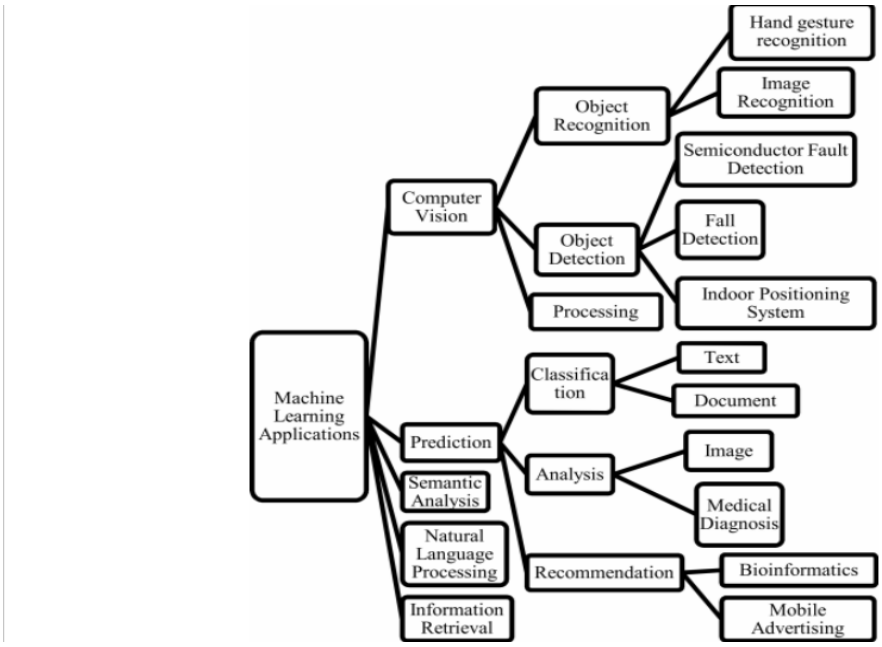

Figure 2. Machine learning applications [16]

2.2.c) ETHICAL CONSIDERATION

These intelligent technologies represent developments in a wide range of industries, including healthcare, banking, transportation, and entertainment, with the promise of unleashing limitless possibilities and efficiencies. Algorithms autonomously explore complex settings requiring decision-making, data drives innovation, and the line between human purpose and machine execution becomes becoming blurred in the fields of artificial intelligence (AI) and machine learning. Here are some cutting-edge and unique foundations to consider in the area of machine learning ethics: [25]

a) Temporal Ethics: The idea of temporal ethics emphasizes the need for moral standards to adapt throughout time and in different contexts.It acknowledges that what is morally right now could not be morally right tomorrow as technology advances and social standards change. This approach encourages continuous assessment and updating of ethical frameworks to maintain their current value and effectiveness.
b) Ethical Imperative: The promotion of AI literacy across society is essential. Ensuring that everyone is aware of the opportunities and limitations of artificial intelligence can empower people to make sensible choices. This literacy can help individuals challenge AI-generated material, avoid deep fakes, and better protect their privacy in an AI-driven society.
c) Ethical Design Patterns: These are preexisting frameworks that are used to design AI systems with ethics in mind. Instead of adding transparency, responsibility, and justice as afterthoughts to their work, developers should utilize ethical design patterns as a tool.
d) Ethical Reward Mechanisms: Tax exemptions or certificates for companies that adhere to strict ethical standards could be used as promotions for advancing ethical AI innovation.
e) AI Impact Assessment: These assessments would examine the potential ethical, social, and environmental consequences of implementing AI before technology is put into use, helping to identify and reduce any negative impacts.

2.3) DEAP LEARNING

2.3.a) DEFINITIONS

Let's start by defining certain terms before diving into the specifics of deep learning. There are several closely related definitions or high-level descriptions for deep learning:

a) A class of machine learning techniques that exploit many layers of non-linear information processing for 200 Introduction supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification.[26]
b) "A sub-field within machine learning that is based on algorithms for learning multiple levels of representation in order to model complex relationships among data. Higher-level features and concepts are thus defined in terms of lower-level ones, and such a hierarchy of features is called a deep architecture. Most of these models are based on unsupervised learning of representations." (Wikipedia on "Deep Learning" around March 2012.)[26]
c) "A sub-field of machine learning that is based on learning several levels of representations, corresponding to a hierarchy of features or factors or concepts, where higher-level concepts are defined from lower-level ones, and the same lower level concepts can help to define many higher-level concepts. Deep learning is part of a broader family of machine learning methods based on learning representations. An observation (e.g., an image) can be represented in many ways (e.g., a vector of pixels), but some representations make it easier to learn tasks of interest (e.g., is this the image of a human face?) from examples, and research in this area attempts to define what makes better representations and how to learn them." (Wikipedia on "Deep Learning" around February 2013.)[26]
d) "Deep learning is a set of algorithms in machine learning that attempt to learn in multiple levels, corresponding to different levels of abstraction. It typically uses artificial neural networks. The levels in these learned statistical models correspond to distinct levels of concepts, where higher-level concepts are defined from lower-level ones, and the same lower level concepts can help to define many higher-level concepts." See Wikipedia http://en.wikipedia.org/wiki/Deep_learning on "Deep Learning" as of this most recent update in October 2013.[26]
e) "Deep Learning is a new area of Machine Learning research, which has been introduced with the objective of moving Machine Learning closer to one of its original goals: Artificial Intelligence. Deep Learning is about learning multiple levels of representation and abstraction that help to make sense of

data such as images, sound, and text." See https://github.com/lisalab/DeepLearningTutorials [26]

Two key characteristics have been highlighted among all of the previous high-level explanations of deep learning. First one is: "models with several nonlinear information processing layers or stages". And the second one: "methods for supervised or unsupervised learning of feature representation at successively higher, more abstract layers" [26]. The fields of neural networks, artificial intelligence, graphical modeling, optimization, pattern recognition, and signal processing all intersect with deep learning. The significantly improved chip processing power, the much larger training datasets, and the latest developments in machine learning and signal/information processing research are three key factors contributing to the current popularity of deep learning. Thanks to these developments, deep learning techniques can now efficiently utilize both labeled and unlabeled data, learn distributed and hierarchical feature representations, and use complicated, compositional nonlinear functions. [26]

2.3.b) DEEP LEARNING CLASSES

As mentioned before, deep learning includes a wide variety of machine learning methodologies and frameworks, characterized by the utilization of multiple hierarchical layers of non-linear information processing.There are three major classes: Deep networks for unsupervised or generative learning, Deep networks for supervised learning and Hybrid deep networks. [26]

a) Deep networks for unsupervised or generative learning: Are meant to capture what is seen or observable data's high-order association for use in pattern analysis or synthesis when target class labels are unknown.This class of deep networks is known in the scientific community as unsupervised feature or representation learning. When available and considered an aspect of the visible data, it may also be used in the generative mode to characterize joint statistical distributions of the visible data and their associated classes. [26]
b) Deep networks for supervised learning: Are generally designed to characterize the following distributions of classes conditioned on the observable data, with the aim of immediately providing discriminative power for pattern classification purposes [26]. Finding weights that produce episodes with little total error, the sum of all such episodes, is a common objective of supervised neural network training. It is hoped that in subsequent episodes, the NN would generalize successfully and introduce only minor errors on sequences of input events that have never been observed before [28].

Also, for this kind of supervised learning, target label data are always accessible in direct or indirect forms.

c) Hybrid deep networks: The objective is discrimination, which is frequently greatly assisted by the results of generative or unsupervised deep networks. Improved optimization and/or normalization of the deep networks in the category can help achieve this. [26]

2.3.c) NEURAL NETWORKS

Deep Learning is a multilayered neural network which is developed to understand how the human brain works and based on this, to dive deeper into machine learning.[23] A typical neural network (NN) is made up of several neurons, which are small, linked processors that generate a series of real-valued activations. Sensors that detect their surroundings activate input neurons, and weighted connections from previously active neurons stimulate additional neurons.Such behavior may involve extensive causal chains of computational steps, where each stage modifies the network's aggregate activation, depending on the issue and how the neurons are connected [28]. The most common types are:

a) Artificial Neural Network (ANN) [23, 27]

Artificial neural network is a type of machine learning which has a node layer, input layer, hidden layer, and output layer. Artificial neurons are connected to one another and have weights and thresholds. If a node's output over the threshold, it sends data to the subsequent layer. Nothing moves on to the next network layer.

Advantages

i. remarkable ability to handle complex nonlinear patterns.
ii. extremely accurate modeling of group data. It is possible to modify the model for both linear and non-linear dynamics.
iii. Having the ability to include noisy and missing input without leading the model to fail.

Disadvantages

i. Overfitting
ii. simply provide projected target values for certain unknown variables without including variance information to evaluate the prediction's accuracy

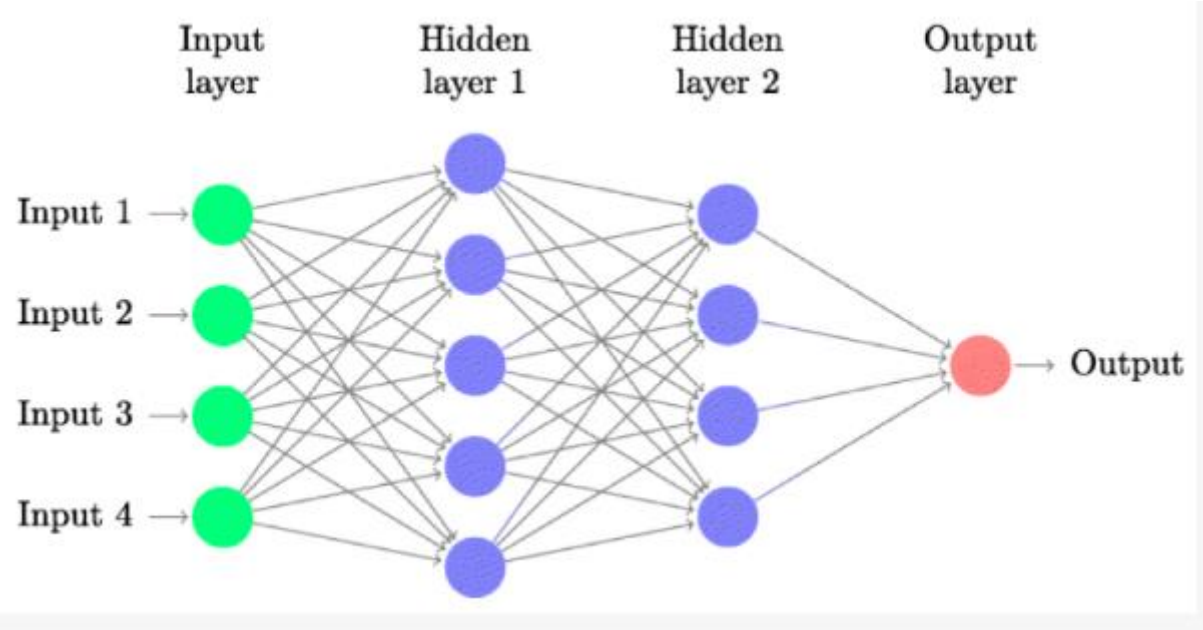

Figure 3. The structure of Artificial Neural Network [23]

b) Recurrent Neural Network (RNN) [23, 27]

There are connections between passes and time in this neural network. This type of artificial neural network allows information to flow back into previous layers and remain there because its nodes form a directed graph along a path.

Advantages

i. This tool serves a useful purpose in showing the temporal connections between the neural network's inputs and outputs.

Disadvantages

i. Difficult to effectively teach and train.

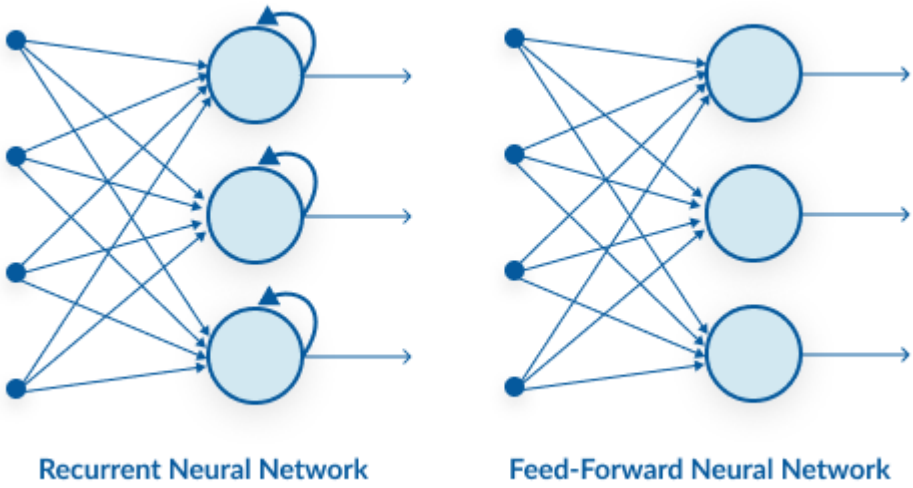

Figure 4. The structure of Recurrent Neural Network [29]

c) Long Short-Term Memory (LSTM) [23, 27]

This is a variation of Recurrent neural network (RNN) and does not suffer from the issue of vanishing gradient compared with RNN. The LSTM is used to learn patterns that happen one after an other and it is made up of several memory modules that are repeated and have three gates in each.

Advantages

i. Able to independently learn patterns and interactions from data
ii. makes precise predictions by examining hidden patterns and data interactions.
iii. capable of holding onto data for a long time.

Disadvantages

i. As a result of the connection between recurrent weight matrix dimensions and memory cell count, indexing the memory during write or read operations is challenging.

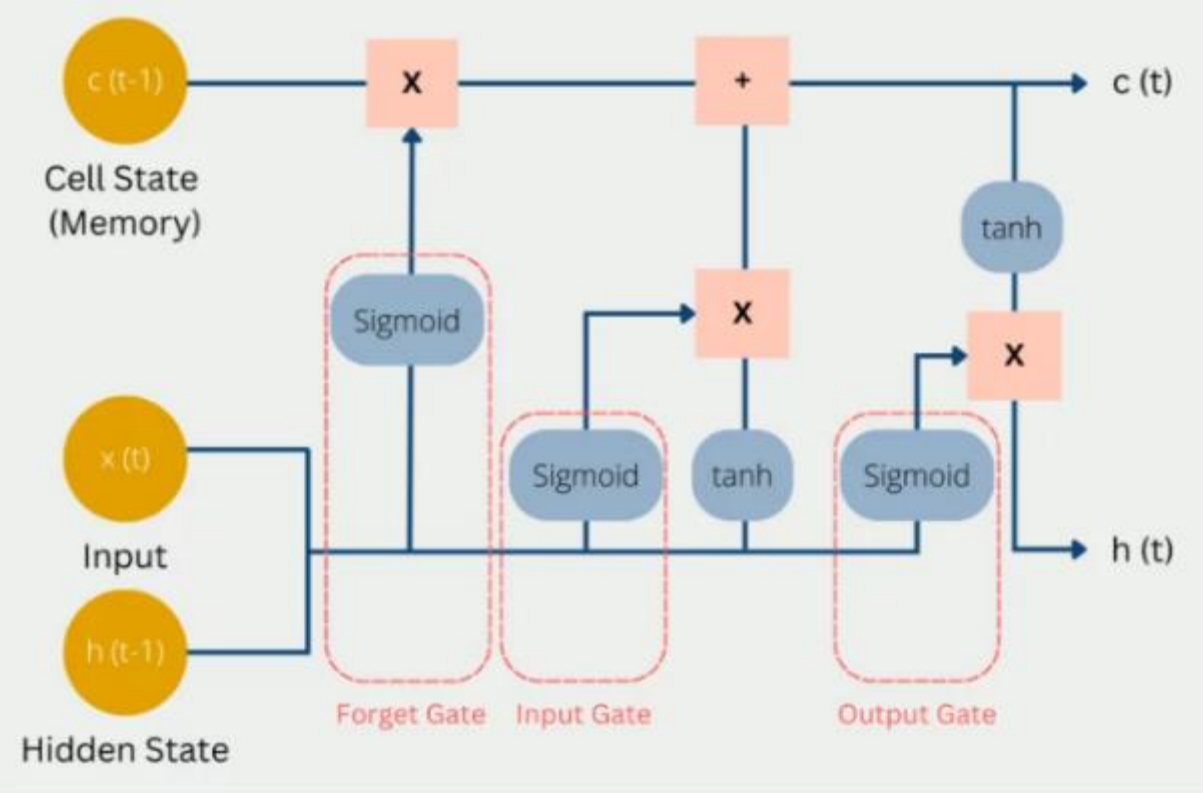

Figure 5. The structure of Long Short-Term Memory [30]

d) Convolutional Neural Network (CNN) [23]

CNNs are inspired from ANNs and are feed-forward networks that only transfer data from sources to destinations. Also, CNN designs include modules of convolutional and pooling layers and is has Input Layers, Convolution Layers, Pooling Layers, Dense Layers, and Output Layers.

Advantages

i. They require less pre-processing than traditional classification algorithms and are capable of self-learning new filters and characteristics.
ii. Offers the distribution of weight.

Disadvantages

i. Requires significant amount of training data.
ii. Significantly increased latency because of the maxpool.

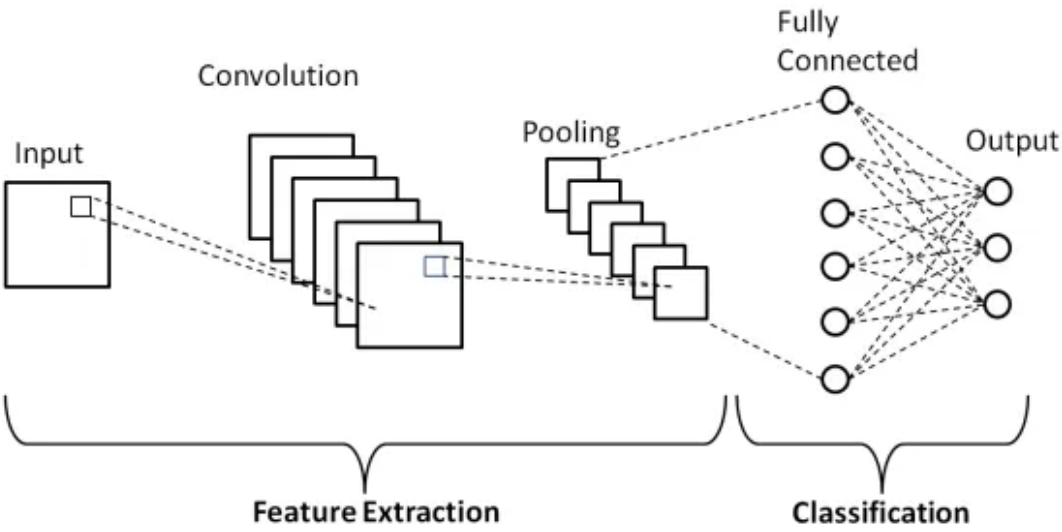

Figure 6. The structure of Convolutional Neural Network [31]

In the new millennium, DL made significant progress with the introduction of low-cost, multi-processor graphics cards, or GPUs. The previous millennium witnessed multiple attempts to develop fast NN-specific hardware and to take advantage of standard hardware. Video games are one of the largest and most competitive markets for GPUs, which has lowered hardware costs. GPUs are particularly good at doing the fast matrix and vector multiplications needed for NN training, which can accelerate learning by a factor of 50 or more, as well as for creating realistic virtual worlds. Recent success in competitions for object detection, image segmentation, and pattern recognition have been mostly due to several GPU-based systems. [28]

2.4) OBJECT DETECTION MODELS

Typically, a present-day detector has three components: a head that predicts object bounding boxes and classes, a backbone that has been pre-trained on ImageNet and a neck that used to collect feature maps from the other stages. Also there are detector that their backbones are running on GPU platfroms and other that are running on CPU platforms: [32]

a) For those that running on GPU: VGG, ResNet, ResNeXt or DenseNet
b) For those that running on CPU: SqeezeNet, MobileNet or ShuffleNet.

Regarding the head part, it is typically divided into two types, one-stage object detector and two-stage object detector. In addition there are some anchor-free detectors. The most popular detector are:[32]

a) One-stage: YOLO, SSD and RetinaNet
b) One-stage and Anchor-Free: CenterNet, CornerNet and FCOS
c) Two-stage: R-CNN, faster R-CNN, R-FCN and Libra R-CNN.
d) Two-stage and Anchor-Free: RepPoints

On the other hand, a neck is composed of several bottom-up and top-down paths. Networks that have this technique installed are: Feature Pyramid Network(PAN), BiFPN and NAS-FPN [32]

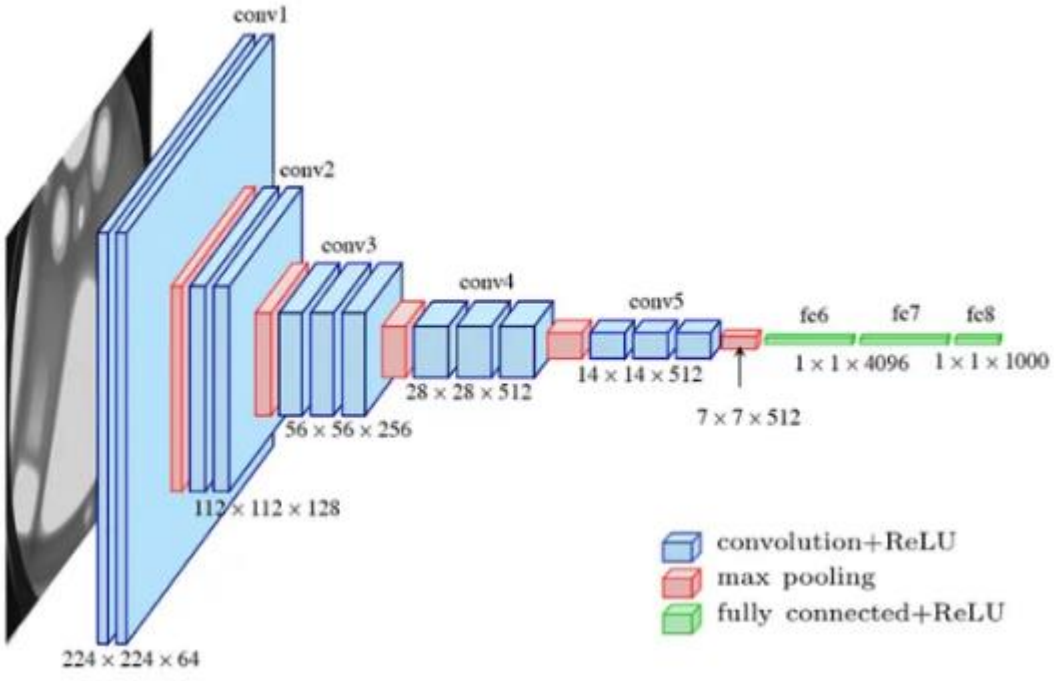

Figure 7. vgg-Net Architecture [33]

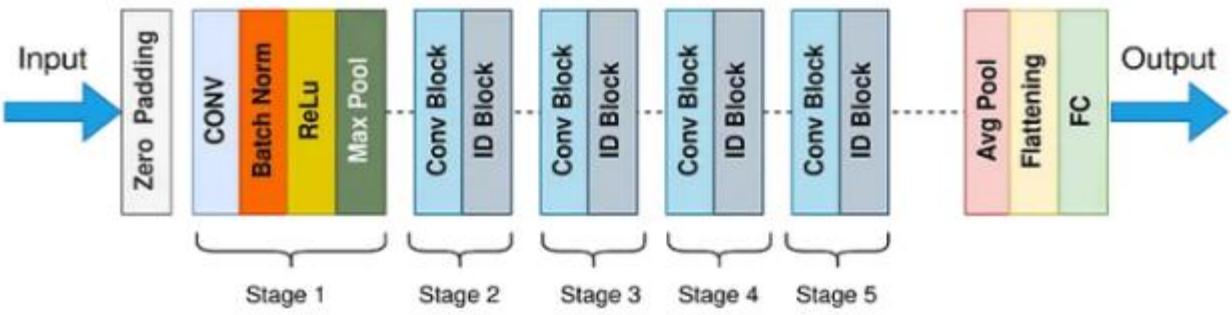

Figure 8. ResNet50 Architecture [34]

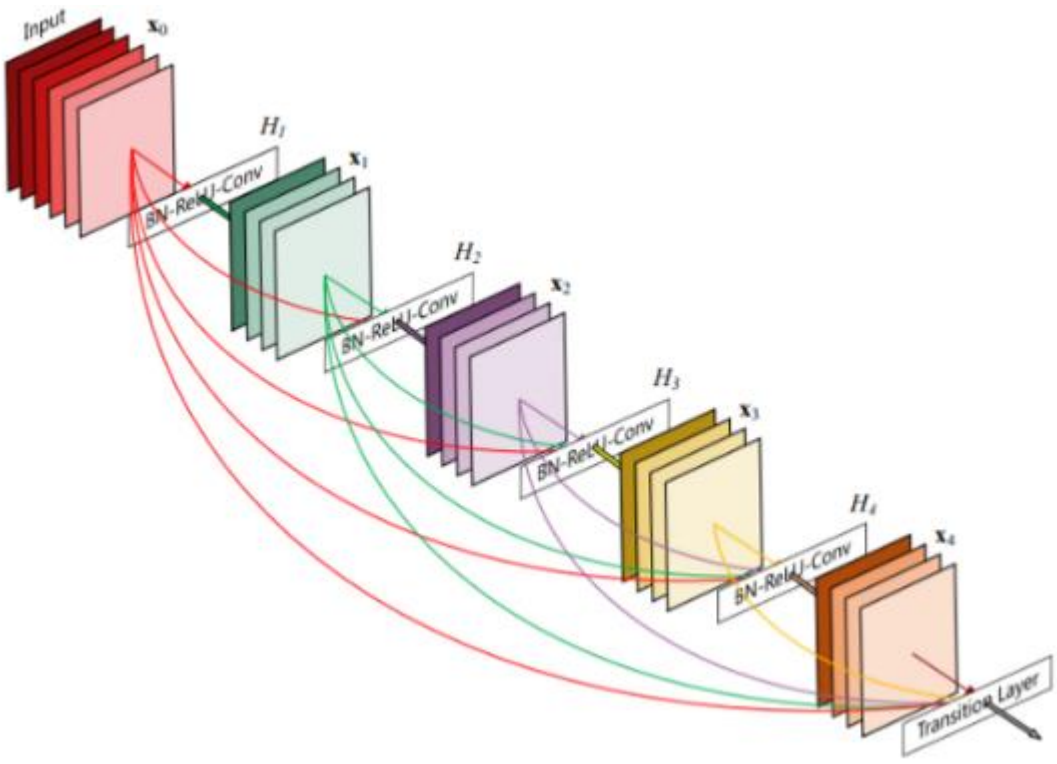

Figure 9. DenseNet Architecture [35]

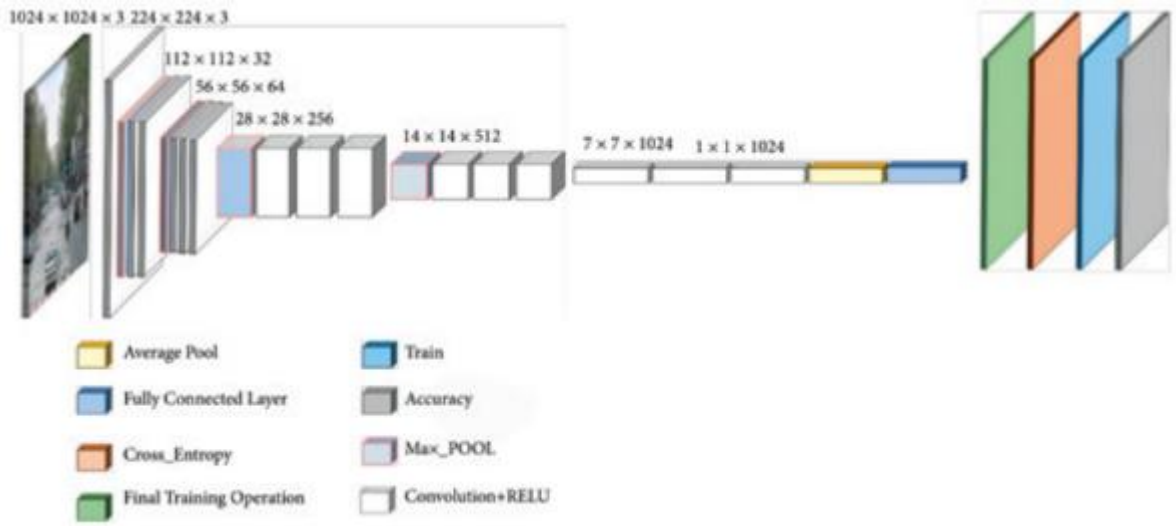

Figure 10. MobileNet V1 Architecture [36]

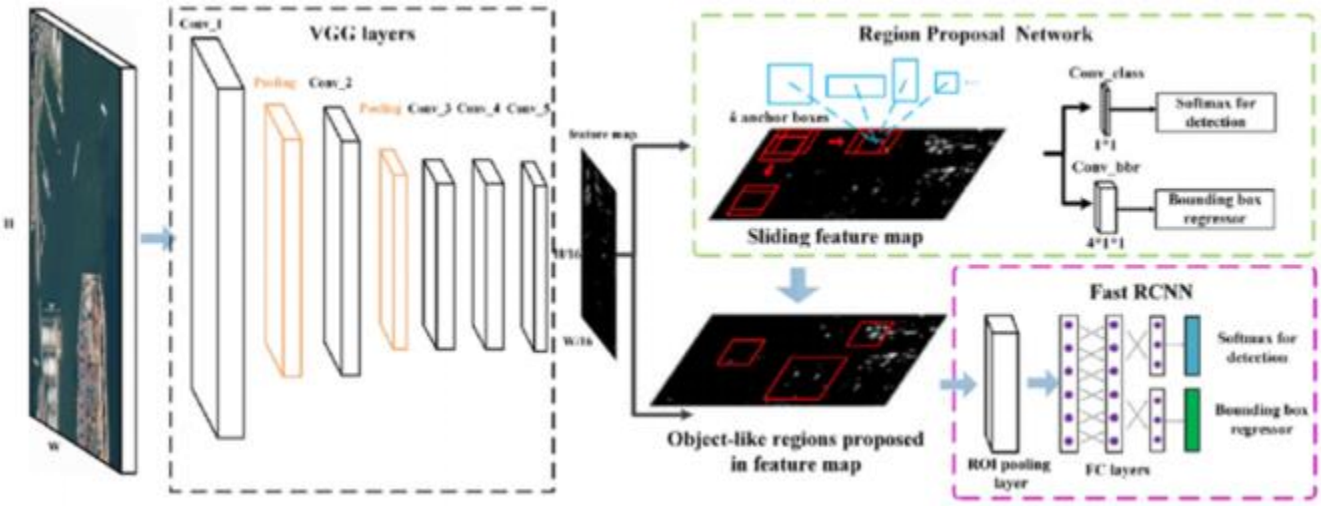

Figure 11. Faster R-CNN Architecture [37]

2.5) APPLICATION IN DAMAGE RECOGNITION

The ability to recognize damage quickly and accurately has become increasingly important across diverse industries, impacting safety, efficiency, and cost-effectiveness. Recent advancements in deep learning, particularly object detection models like YOLOv7, have fueled a surge in innovative applications for damage recognition. However, it's not just YOLOv7 making waves; numerous AI models are finding their niche in this critical field. Let's explore some key areas where these models are shining:

a. Infrastructure Inspection: From towering bridges to sprawling pipelines, traditional inspections often rely on manual visual checks, prone to subjectivity and time constraints. AI models, including Faster R-CNN, SSD, and Mask R-CNN, are stepping in to automate this process. Analyzing aerial or ground-level images, they can detect cracks, corrosion, and other damage types, facilitating timely intervention and preventing potential disasters.[2, 3, 4]

b. Building Maintenance: AI models like EfficientDet and RetinaNet are playing a vital role in building maintenance. Analyzing drone footage or inspection photos, they pinpoint damage accurately and efficiently, enabling targeted repairs and optimized resource allocation.[5]

c. Insurance Claims Processing: Traditionally, human adjusters assess vehicle damage after accidents and property damage following natural disasters. AI models like U-Net and DeepLabCut are changing the game. By automatically analyzing photos or videos submitted by policyholders, these models can identify the extent and type of damage with impressive accuracy, leading to faster claim settlements and improved customer satisfaction.[6, 7]

d. Disaster Response and Recovery: The aftermath of natural disasters necessitates rapid damage assessment for prioritizing rescue efforts and allocating resources effectively. Satellite imagery or drone footage processed by AI models like DensePose and PointNet++ can quickly identify damaged buildings, roads, and infrastructure, guiding critical initial response and recovery efforts.[8, 9]

This glimpse into the world of AI-powered damage recognition showcases just a few exciting possibilities. As these models continue to evolve, their potential to revolutionize damage detection across various fields will only grow, fostering increased safety, cost optimization, and streamlined processes across industries.

3) PROPOSED MODEL

3.1) YOLO SERIES

You Only Look Once (YOLO) is one of the most popular object detection methods, according to the literature's findings. Numerous variations of this popular object detection technique have been released. There has been significant evolution in terms of detection time when we examine the evolution of all the YOLO series. The YOLO was designed to run on low-processor devices when it was initially released.[42]

The YOLOv3 and YOLOv4 versions can be seen in the YOLO algorithm timeline. In conclusion, both YOLOv3 and YOLOv4 are deep learning-based object detection algorithms; however, YOLOv4 outperforms YOLOv3. To increase its accuracy, YOLOv4 has been trained on a big dataset of photos and videos and optimized for real-time object detection. To improve performance, YOLOv4 includes new methods like DropBlock and Mosaic data augmentation.[42]

The algorithm's fifth iteration, known as YOLOv5, was then made public. Though it still needs to get closer to the fifth major update, this method turned out to be an excellent model, offering more alternatives as we can point out the image segmentation. It is predicated on a novel SPADE architecture that enhances object detection accuracy by utilizing both semantic and geographical information. To improve the model's generalization, YOLOv5 additionally employs a novel training approach known as Mosaic Data Augmentation.[42]

The most recent iteration in the cycle of YOLO models, the seventh version of the algorithm, was made available. It is based on a novel architecture known as Efficient-YOLO, with EfficientNet serving as the main network. YOLOv7 is optimized for real-time object recognition after being trained on an extensive dataset. It is faster and more accurate than the earlier iterations of YOLO.[42]

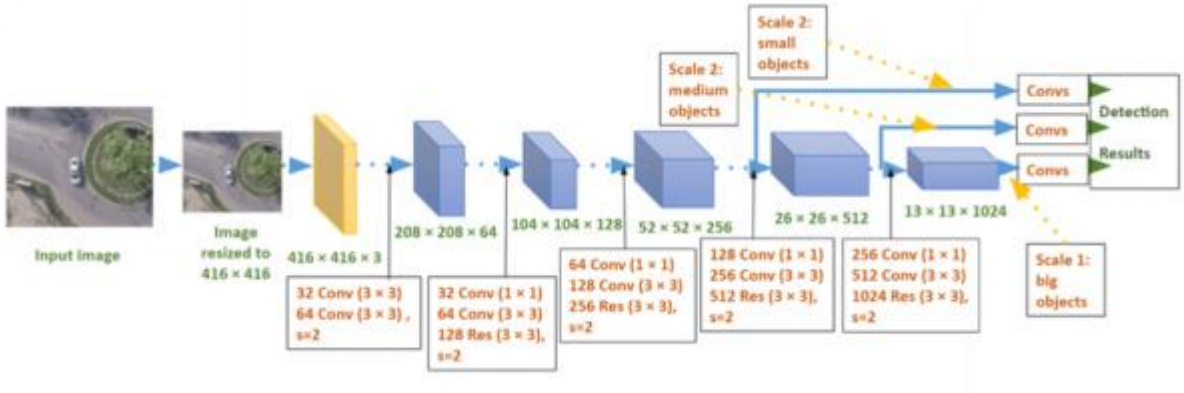

Figure 12. YOLOv3 Architecture [46]

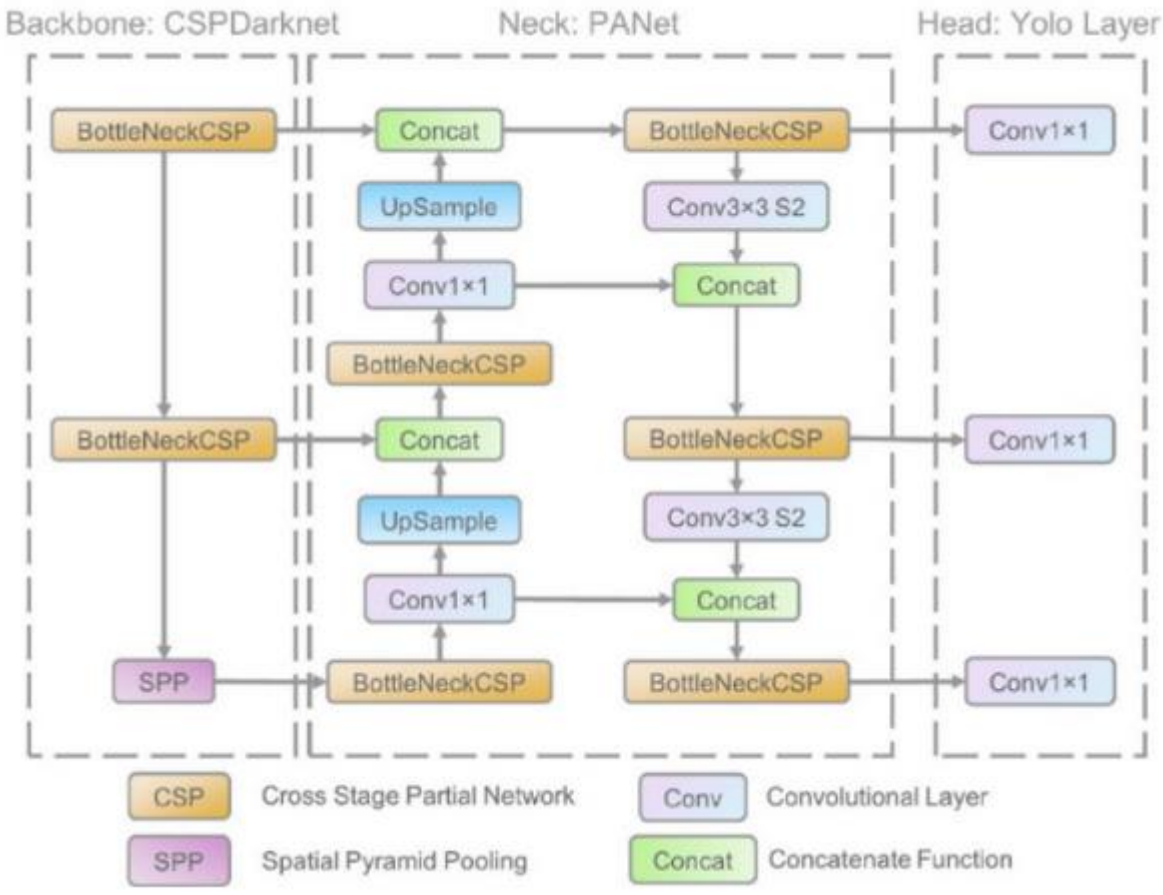

Figure 13. YOLOv5 Network Architecture [47]

3.2) OVERVIEW OF YOLOv7 MODEL

YOLOv7 stands as a groundbreaking real-time object detector that has caused a significant stir within the computer vision (CV) industry owing to its remarkable features. Remarkably, YOLOv7 was meticulously trained solely on the MS COCO dataset from scratch, eschewing the utilization of any additional datasets or pre-trained weights, as highlighted by Wang et al. (2022). According to their findings, YOLOv7 outshines all existing object detectors in terms of both speed and accuracy, operating seamlessly within the impressive range of 5 FPS to 160 FPS. Notably, it achieves the highest accuracy, boasting a remarkable 56.8% Average Precision (AP) among all known real-time object detectors with a framerate of 30 FPS or higher on GPU V100. YOLOv7 achieves this feat while maintaining a steady inference cost, as it reduces parameters by approximately 40% and computational requirements by 50% compared to state-of-the-art real-time object detectors. This results in faster inference speeds and heightened detection accuracy, solidifying YOLOv7's position as a game-changer in the field of real-time object detection.[1]

YOLOv7 introduces the innovative Efficient Layer Aggregation Networks (E-ELAN), which leverage techniques such as expand, shuffle, and merge cardinality to continuously enhance the network's learning capacity while preserving the original gradient path, as elucidated by Wang et al. (2022). Notably, E-ELAN modifies only the computational block architecture while keeping the transition layer architecture intact. Moreover, E-ELAN facilitates diverse feature learning among different groups of computational blocks while adhering to the original E-LAN design architecture. Additionally, YOLOv7 incorporates model scaling for concatenation-based models, aimed at adjusting specific attributes to generate models of varying scales to accommodate diverse inference speed requirements. The proposed compound scaling method ensures that the model retains its initial design properties and optimal structure. Figure 3 illustrates the model scaling for concatenation-based models in YOLOv7.[1]
While YOLOv6 showcases notable improvements in detection capabilities, it falls short in terms of scalability and training ease when juxtaposed with YOLOv5 and YOLOv7. Additionally, YOLOv6 exhibits superior accuracy in single-image inference scenarios compared to the multiple-image inference accuracy offered by YOLOv5 and YOLOv7, as highlighted by Banerjee (2022). Consequently, an experiment was conducted focusing on YOLOv5 and YOLOv7 due to their suitability for multiple object detection tasks, offering streamlined customization options for both training and inference processes.[1]

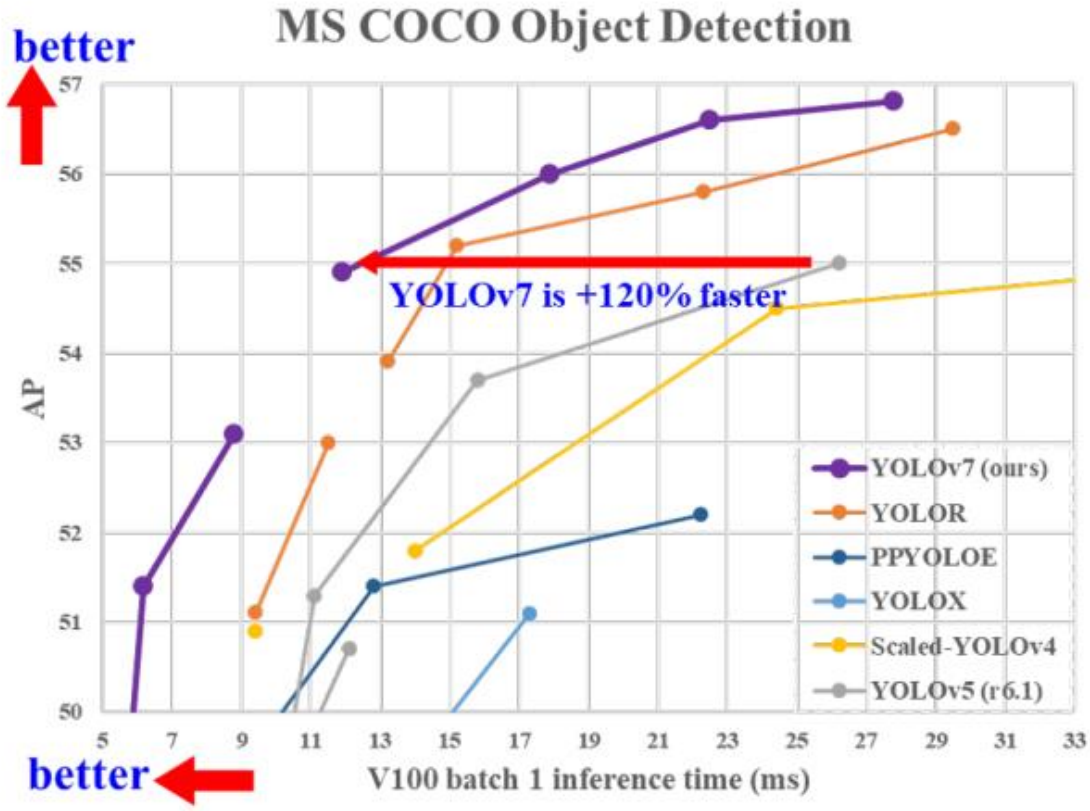

Figure 14. Comparison of the best real-time object detectors [11]

Model	#Param.	FLOPs	Size	FPS	AP^{test} / AP^{val}
YOLOX-S [21]	9.0M	26.8G	640	102	40.5% / 40.5%
YOLOX-M [21]	25.3M	73.8G	640	81	47.2% / 46.9%
YOLOX-L [21]	54.2M	155.6G	640	69	50.1% / 49.7%
YOLOX-X [21]	99.1M	281.9G	640	58	51.5% / 51.1%
PPYOLOE-S [85]	7.9M	17.4G	640	208	43.1% / 42.7%
PPYOLOE-M [85]	23.4M	49.9G	640	123	48.9% / 48.6%
PPYOLOE-L [85]	52.2M	110.1G	640	78	51.4% / 50.9%
PPYOLOE-X [85]	98.4M	206.6G	640	45	52.2% / 51.9%
YOLOv5-N (r6.1) [23]	1.9M	4.5G	640	159	- / 28.0%
YOLOv5-S (r6.1) [23]	7.2M	16.5G	640	156	- / 37.4%
YOLOv5-M (r6.1) [23]	21.2M	49.0G	640	122	- / 45.4%
YOLOv5-L (r6.1) [23]	46.5M	109.1G	640	99	- / 49.0%
YOLOv5-X (r6.1) [23]	86.7M	205.7G	640	83	- / 50.7%
YOLOR-CSP [81]	52.9M	120.4G	640	106	51.1% / 50.8%
YOLOR-CSP-X [81]	96.9M	226.8G	640	87	53.0% / 52.7%
YOLOv7-tiny-SiLU	6.2M	13.8G	640	286	38.7% / 38.7%
YOLOv7	36.9M	104.7G	640	161	51.4% / 51.2%
YOLOv7-X	71.3M	189.9G	640	114	53.1% / 52.9%
YOLOv5-N6 (r6.1) [23]	3.2M	18.4G	1280	123	- / 36.0%
YOLOv5-S6 (r6.1) [23]	12.6M	67.2G	1280	122	- / 44.8%
YOLOv5-M6 (r6.1) [23]	35.7M	200.0G	1280	90	- / 51.3%
YOLOv5-L6 (r6.1) [23]	76.8M	445.6G	1280	63	- / 53.7%
YOLOv5-X6 (r6.1) [23]	140.7M	839.2G	1280	38	- / 55.0%
YOLOR-P6 [81]	37.2M	325.6G	1280	76	53.9% / 53.5%
YOLOR-W6 [81]	79.8G	453.2G	1280	66	55.2% / 54.8%
YOLOR-E6 [81]	115.8M	683.2G	1280	45	55.8% / 55.7%
YOLOR-D6 [81]	151.7M	935.6G	1280	34	56.5% / 56.1%
YOLOv7-W6	70.4M	360.0G	1280	84	54.9% / 54.6%
YOLOv7-E6	97.2M	515.2G	1280	56	56.0% / 55.9%
YOLOv7-D6	154.7M	806.8G	1280	44	56.6% / 56.3%
YOLOv7-E6E	151.7M	843.2G	1280	36	56.8% / 56.8%

Figure 15. Comparison of the best real-time object detectors [9]

3.3) DESIGN ARCHITECTURE

3.3.a) LAYER AGGREGATION NETWORKS

YOLOv7 takes object detection to new heights with its innovative design. Built on efficient building blocks, it learns diverse features for accurate object recognition. Unlike older versions, it cleverly combines information from different scales, resulting in sharper predictions for objects of all sizes. This clever architecture pushes the boundaries of both speed and accuracy, making YOLOv7 a powerful tool for real-world applications.[10]

a) Backbone: YOLOv7 uses an Extended Efficient Layer Aggregation Network (E-ELAN) as its backbone.This builds upon the ELAN architecture by adding residual connections and bottleneck layers for improved efficiency. E-ELAN allows the model to learn diverse features at different scales while remaining computationally efficient.[10]

b) Neck: YOLOv7 employs a PAN (Path Aggregation Network) neck, which fuses features from different levels of the backbone to create rich feature maps for object detection. This helps in recognizing objects of various sizes and improves accuracy.[10]

c) Head: The head uses YOLOv5's head structure with three prediction branches for different anchor box sizes. This predicts bounding boxes, objectiveness scores, and class probabilities. Introduces CIOU Loss for better bounding box predictions and Cross-Stage Partial Connections (CSPC) for improved information flow between feature maps, further enhancing accuracy.[10]

3.3.b) MODEL SCALING

YOLOv7 uses "compound scaling" to offer different versions with varied inference speeds and accuracy levels. This caters to diverse needs – from real-time applications on mobile devices to high-precision tasks on powerful computers.[10]

a) Backbone: YOLOv7 uses two scaling factors: "depth" and "width." Increasing depth expands the number of convolutional layers for deeper feature extraction, impacting accuracy more than speed. Conversely, increasing width expands the number of channels in each layer, impacting both accuracy and speed.[10]

b) Head: Here, scaling focuses on "channels" only, expanding the feature dimensions for prediction, affecting both accuracy and speed.[10]

This approach creates several YOLOv7 models (YOLOv7-Tiny, YOLOv7-Nano, etc.) with different trade-offs in speed and accuracy.

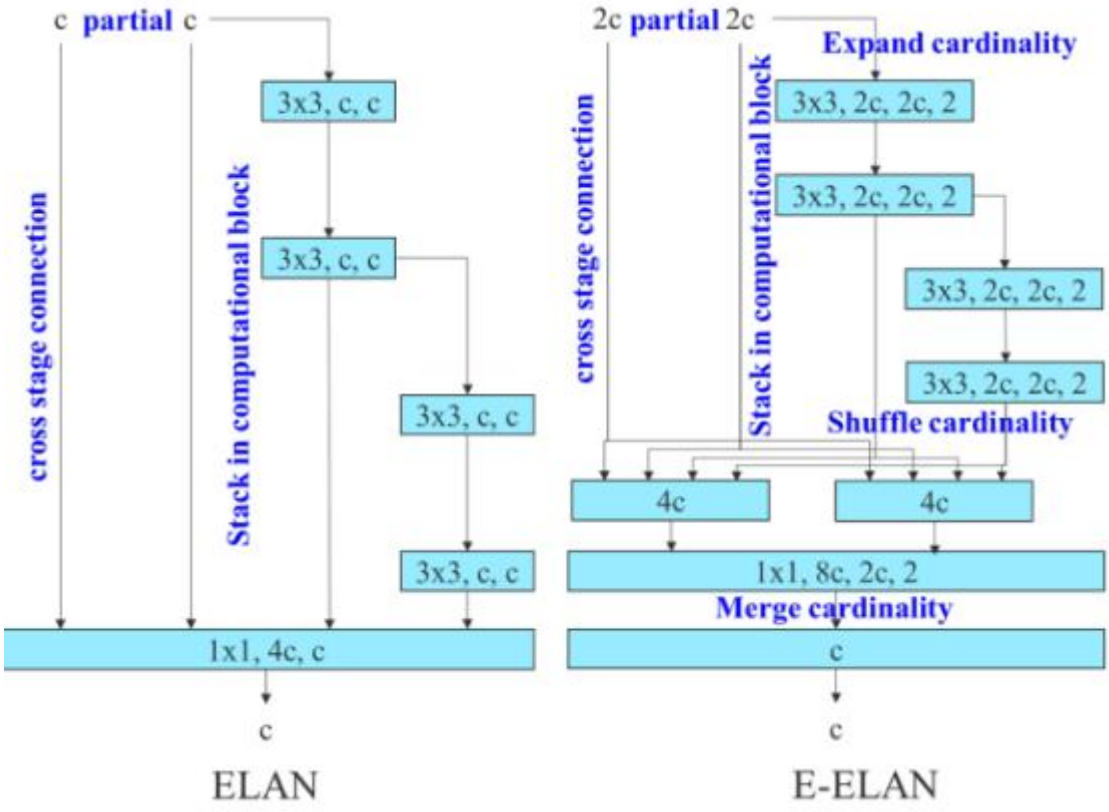

Figure 16. Difference between ELAN and E-ELAN [10]

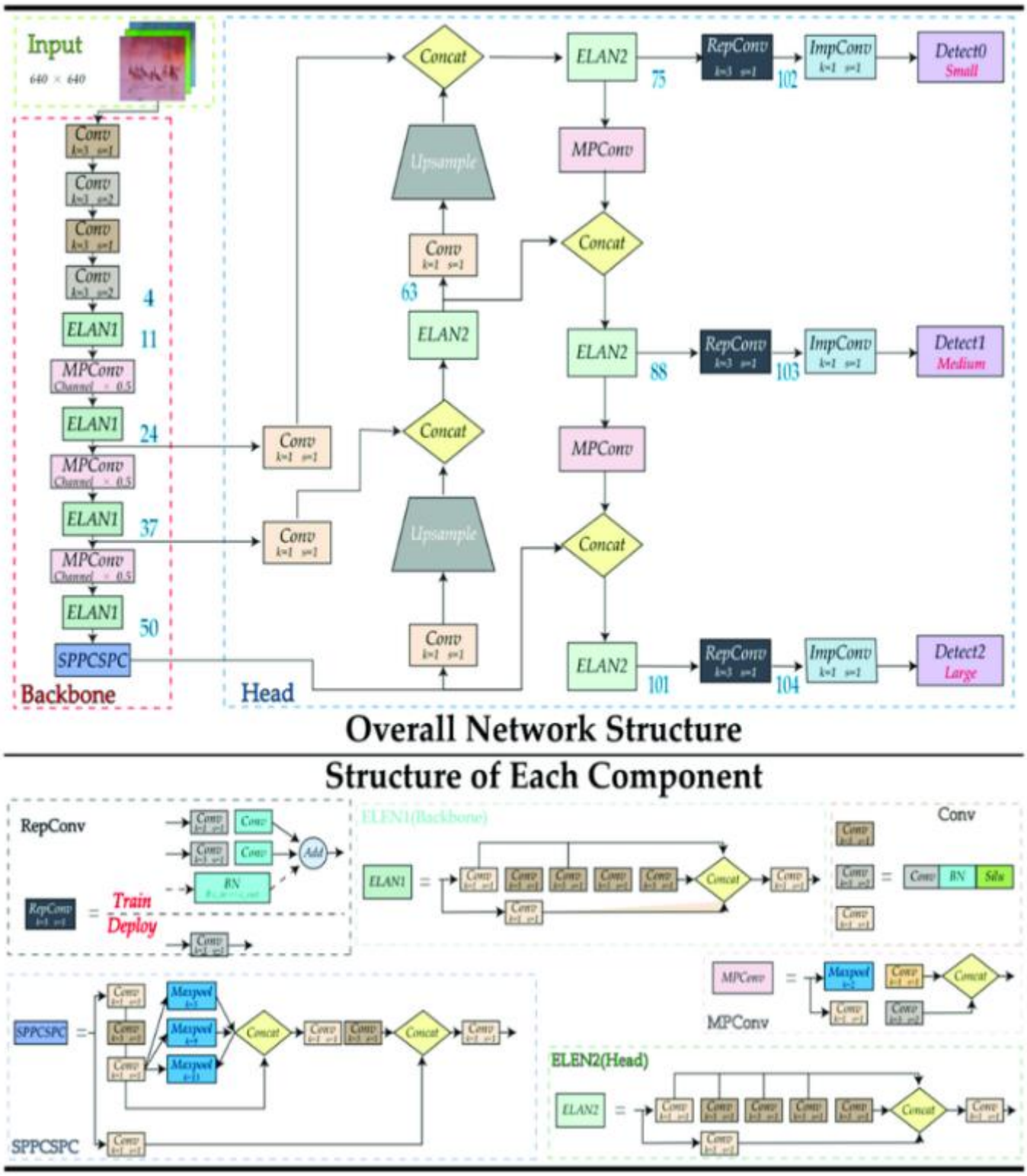

Figure 17. Architecture of YOLOv7 [11]

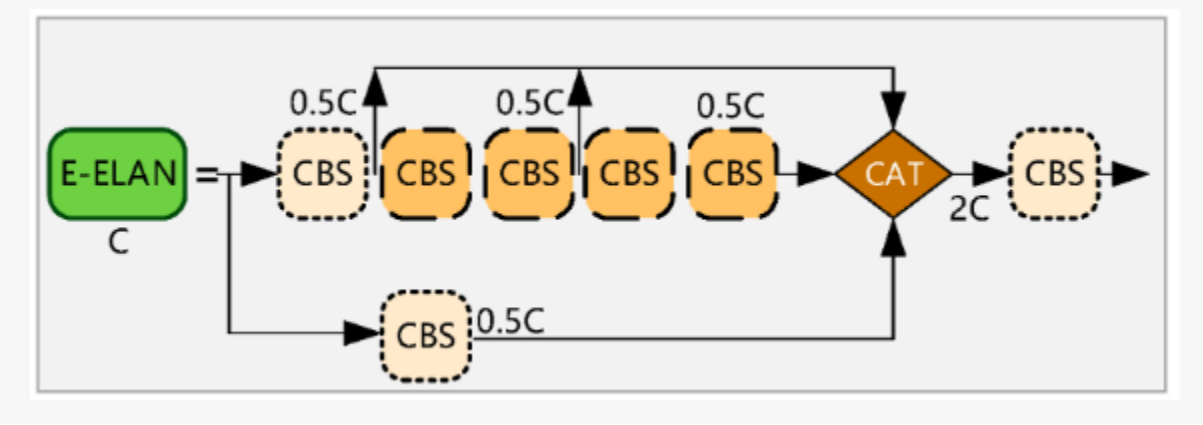

Figure 18. Architecture of E-ELAN [41]

3.4) TRAINING PROCESS

YOLO, an object detection algorithm, dissects images into a uniform grid. Each grid cell predicts object presence and type. The algorithm then generates potential bounding boxes around objects in each cell, but not all are accurate. To find the best ones, YOLO employs Intersection over Union (IoU), a metric measuring the overlap between predicted and actual object boxes. Using these scores, YOLO selects the most accurate boxes, ultimately identifying and classifying objects within the image. Using the following formula the IoU will be calculated for every grid cell:[12]

$$IoU = \frac{area(A_p \cap A_{gt})}{area(A \cup A_{gt})}$$

A_p: Region of the detected object
A_{gt}: Region of ground truth

While Intersection over Union (IoU) is a common technique in object detection, it's not perfect. An object can be identified by multiple bounding boxes, all exceeding the IoU threshold, leading to inaccurate results. To overcome this, Non-Maximum Suppression (NMS) steps in as a final step. This clever technique keeps only the bounding boxes with the highest detection probabilities, ensuring a more accurate picture.[12]

3.5) INTEGRATION WITH UAVs

2.5.a) OBJECT DETECTION WITH UAVs

The development of high-definition cameras and embedded gadgets has opened up a wide range of uses for unmanned aerial vehicles (UAVs), including aerial photography, exploration, and search and rescue. One of the primary components of many advanced UAV applications is a real-time object detection model that works well with UAVs.

The scales of items in aerial photos vary considerably more than in images captured in daily life. In addition, most items in aerial image detection tasks, such humans and vehicles, have lower sizes. As a result, the majority of them are small objects (32*32). Another problem with object recognition on UAVs is that small objects have fewer visual cues and so are more likely to be incorrectly recognized or overlooked. Therefore, it is more challenging to detect effectively in photographs taken by drones than it is in other object detection jobs facing typical scenarios.[48]

However, the computational power of UAV-equipped edge computing modules is not as strong as that of GPU-equipped deep learning platforms. The majority of widely used deep neural networks have many trainable parameters and require a significant amount of floating point computations. Using ResNet as an example, the 152-layer network has over 60 million parameters and calculates at a rate of over 20 billion floating-point operations per second (FLOPS).Known for its excellent balance between speed and accuracy, Yolov3 is an object identification model with over 62 million parameters. The performance of using full-size object detection models directly on AI edge modules is not optimal, hence lighter detectors for UAVs must be designed. [48]

For UAV detectors, accuracy and speed must be traded off. Features can be transported across networks more quickly by refining the feature extraction strategy in the backbone, which may increase the accuracy of detection when small objects are present. However, by requiring channel pruning, it is possible to compress models with minimal loss to accuracy due to the unpromising detection performance of deep networks. [48]

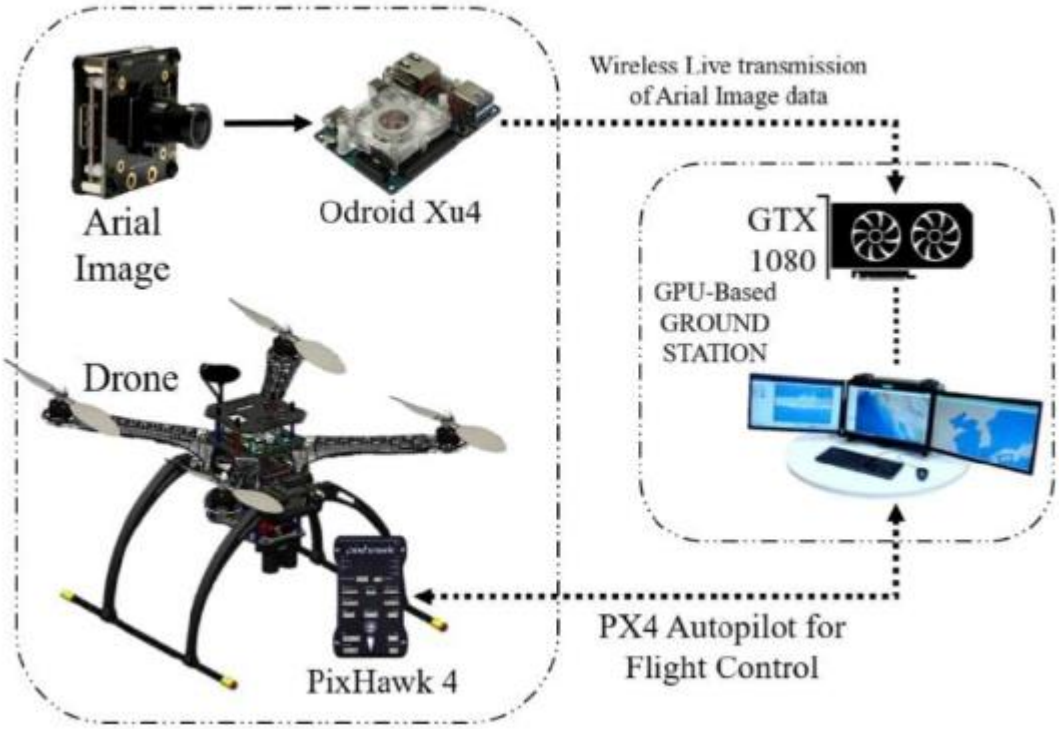

Figure 19. GPU-Based ground station and UAV [49]

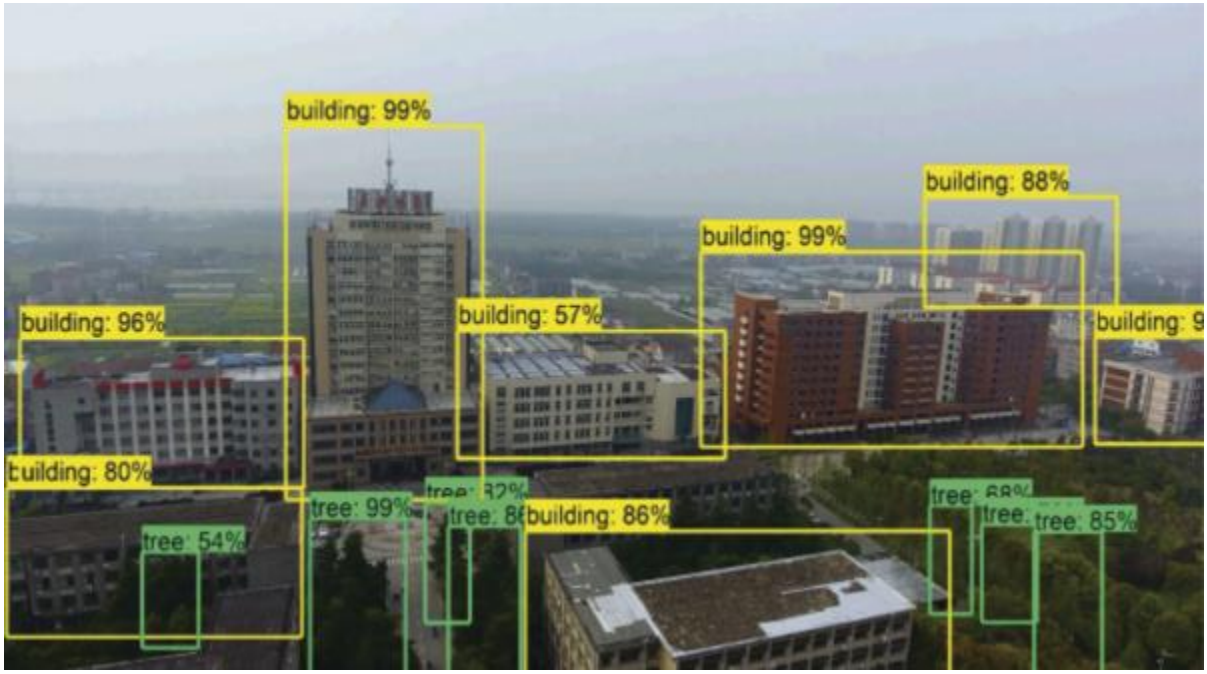

Figure 20. Object Detection using UAV [50]

2.5.b) INTEGRATION OF YOLOv7 WITH UAV

To enhance damage detection in UAV aerial images, this method proposes an improved YOLOv7-based approach. Building upon the original YOLOv7 structure, it incorporates lightweight modules to streamline the model and utilize the CBAM attention mechanism [13] for focused damage recognition. Furthermore, diverse data augmentation techniques (mosaic, mixup) and label smoothing address dataset limitations and model generalization, respectively. Lastly, k-means clustering optimizes prior bounding boxes. This refined method achieves superior detection accuracy, real-time capability, and enhanced robustness and generalization for ground damage assessments.[14]

Figure 21. Improved yolov7 network structure diagram[14]

Overhead UAV images often suffer from abundant irrelevant details that hinder object detection. The lightweight CBAM mechanism effectively tackles this issue by sequentially analyzing feature maps across "channel" and "spatial" dimensions. It generates attention maps highlighting crucial features while suppressing distracting background information. This pinpointed focus empowers the YOLO algorithm to learn and identify even small ground targets amidst visual clutter, significantly improving its detection accuracy.[14]

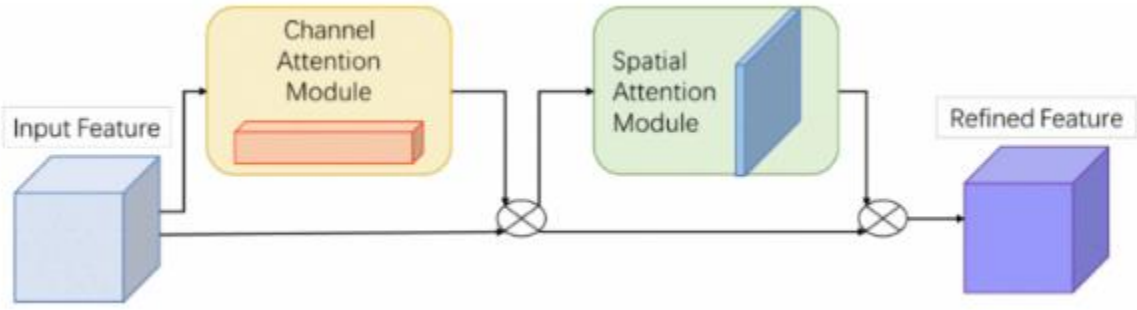

Figure 22. CBAM structure diagram[14]

Imagine CBAM as a multi-stage information processor for images. First, it feeds the input features into a channel attention module that acts like a filter, prioritizing relevant information based on individual channels (think of color intensities). This filtered output (CAM) is then multiplied back onto the original features, highlighting important channels. Next, the spatial attention module takes center stage. It analyzes the CAM output across different image locations, identifying and amplifying areas containing key features. Essentially, it zooms in on the "where" while the channel module focused on the "what." Finally, the combined result emerges, emphasizing both important channels and relevant image regions, guiding the YOLO algorithm towards accurate target detection.In simpler terms: CBAM works like a two-step lens. One lens sifts through different color information, focusing on essential channels. The other lens scans various image locations, pinpointing areas with key features. Combining these views creates a sharpened image where the YOLO algorithm can easily spot its targets.[14]

$$F' = M_c(F) \otimes F$$
$$F'' = M_s(F') \otimes F'$$

Where

$$M_c \in R^{c*1*1}$$
$$F \in R^{c*H*W}$$
$$M_s \in R^{1*H*W}$$

Figure 23. YOLOv7 applied for computer vision [51]

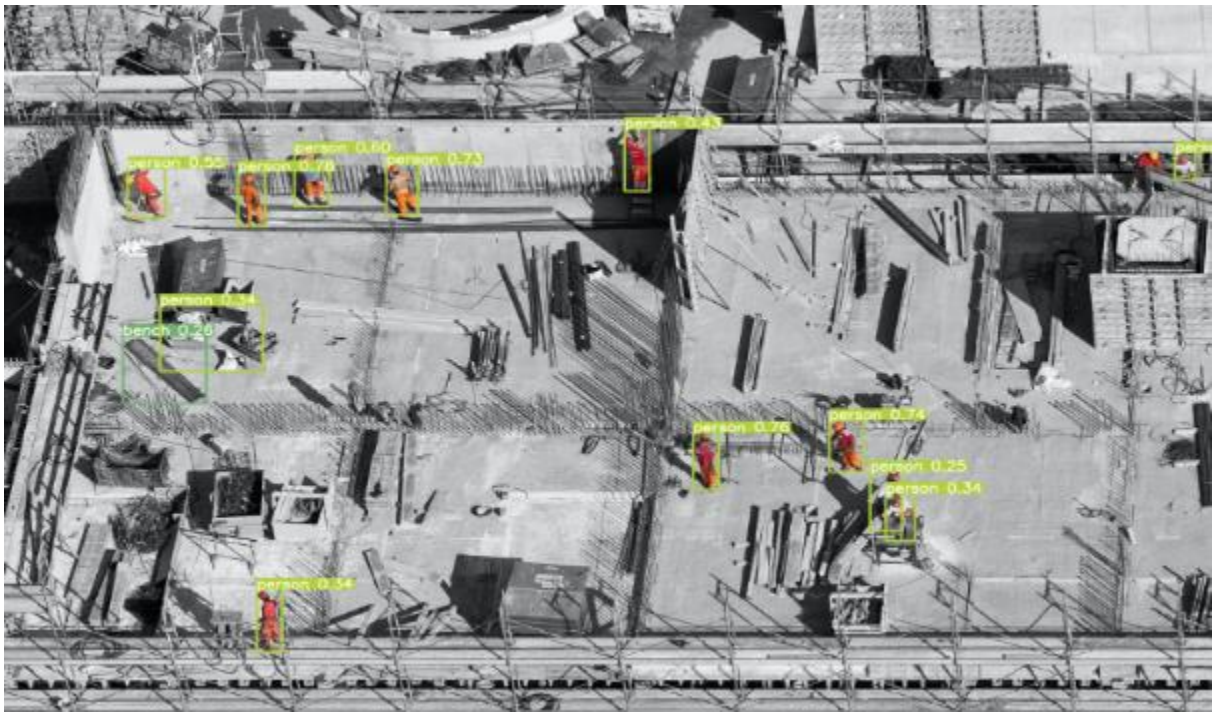

Figure 24. YOLOv7 applied for computer vision in construction [51]

3.6) RELATED WORKS

When making an initial assessment of the state of a wall or building, imagery collection is essential. An efficient and economical method of taking detailed and high-quality photos of the wall surface from different angles is to use a UAV, more especially a drone. Drones need to be equipped with cutting-edge technology like GPS, obstacle avoidance sensors, and a high-resolution camera in order to take clear photos of road surfaces with little distortion. Furthermore, it is safe and quick to use a UAV to cover a larger area of the road surface, particularly in locations that are difficult to access. [42]

Related articles concentrate on enhancing current deep learning algorithms and unmanned aerial vehicle (UAV) algorithms. Autonomous UAVs have been employed, for instance, in real-time damage mapping and structural health monitoring utilizing acoustic beacons with GPS tracking and deep learning techniques. In a number of fields, including animal recognition, wind generator inspection, electric component detection, vehicle traffic monitoring, and huge population monitoring, deep learning techniques, like CNNs, have demonstrated positive results. These methods are useful for automatic road damage detection since they may also be used to evaluate pictures or videos taken by cameras installed on cars in order to find potholes in the road.[42]

The authors of a different recent study [43] suggested an automated road damage identification technique utilizing UAV photos that relies on deep learning-based object detection. The object detector they employed was the Faster R-CNN algorithm. The suggested strategy outperforms previous approaches for detecting road damage, according to the results.

In order to achieve high performance and quick processing speed, Kang [44] presents a unique semantic transformer representation network (STRNet) for crack segmentation in complicated scenarios. When the network was assessed and contrasted with other cutting-edge networks, it proved to have better processing speed and performance. With a mean intersection over the union of 0.900, a positive predictive value of 0.952, an F1-score of 0.941, and a sensitivity of 0.942, the attention-based IDSNet performs better than cutting-edge networks.

Another viewpoint that is particularly useful for identifying concrete or road damage is Single-Shot Detection (SSD). The SDDNet, a deep learning model for real-time segmentation of concrete cracks in photographs, is shown in work [45]. On a manually constructed dataset, the model achieves good accuracy. When compared to more recent models, the model performs better and processes images at a rate of 36 frames per second, which is much faster than earlier efforts.

.

4) EXPERIMENTAL EVALUATION

4.1) DATASET DESCRIPTION

The dataset used for this experiment was Roboflow Public Dataset (Roboflow,n.d), content 4029 images annotated as cracks. The data labeling for segmentation will be a polygon box, while data labeling for object detection will be a bounding box. The dataset was split into training, testing and validation on ratio 92:05:03 of the number of images annotated. 3717 images which makes up 92% of the dataset was used for training while 112 images which makes up 3% of the total images was used for testing and 200 images which amounts to 5% of the total images remaining was used for validation [15]

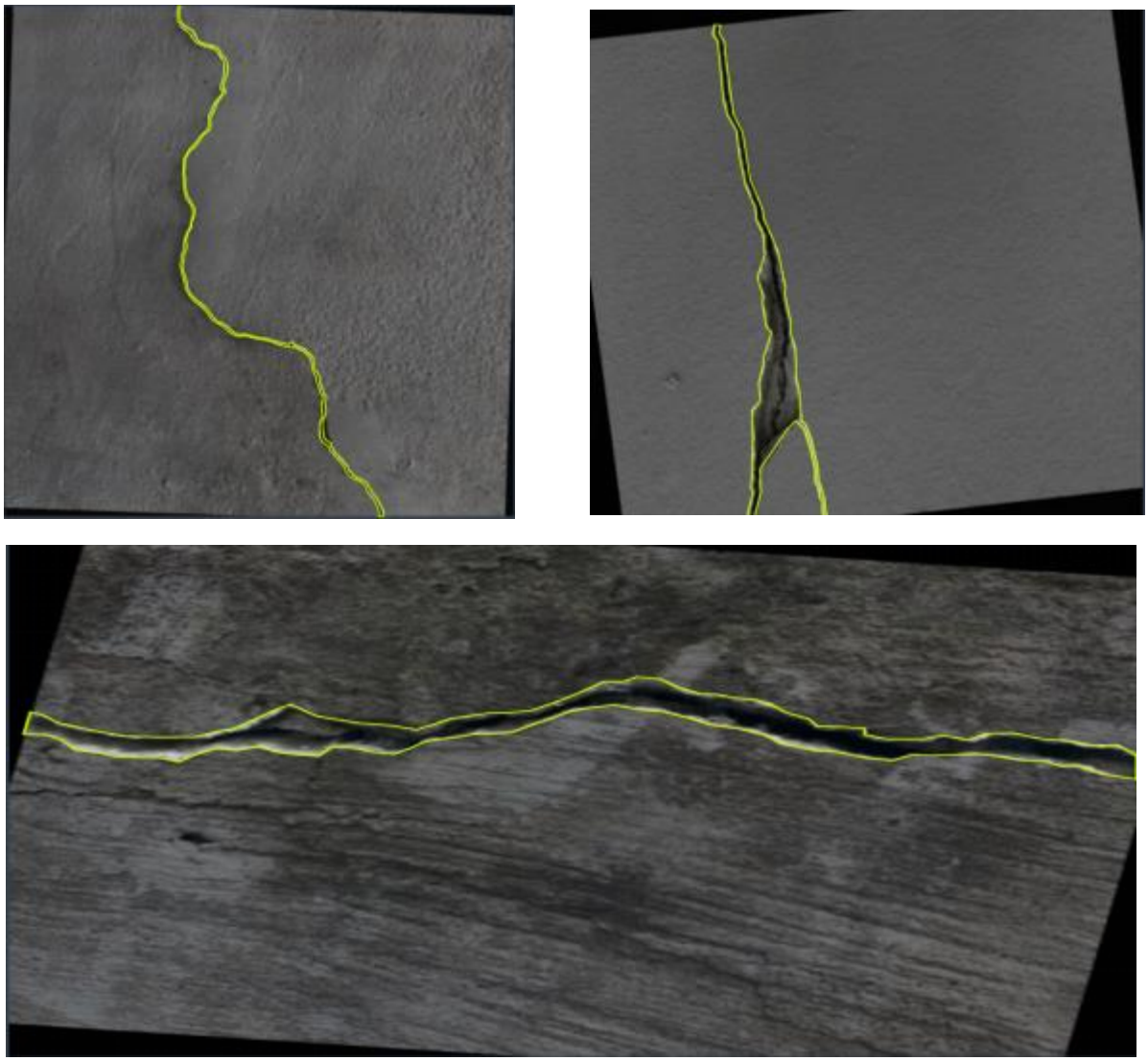

Figure 25. Annotated crack images [15]

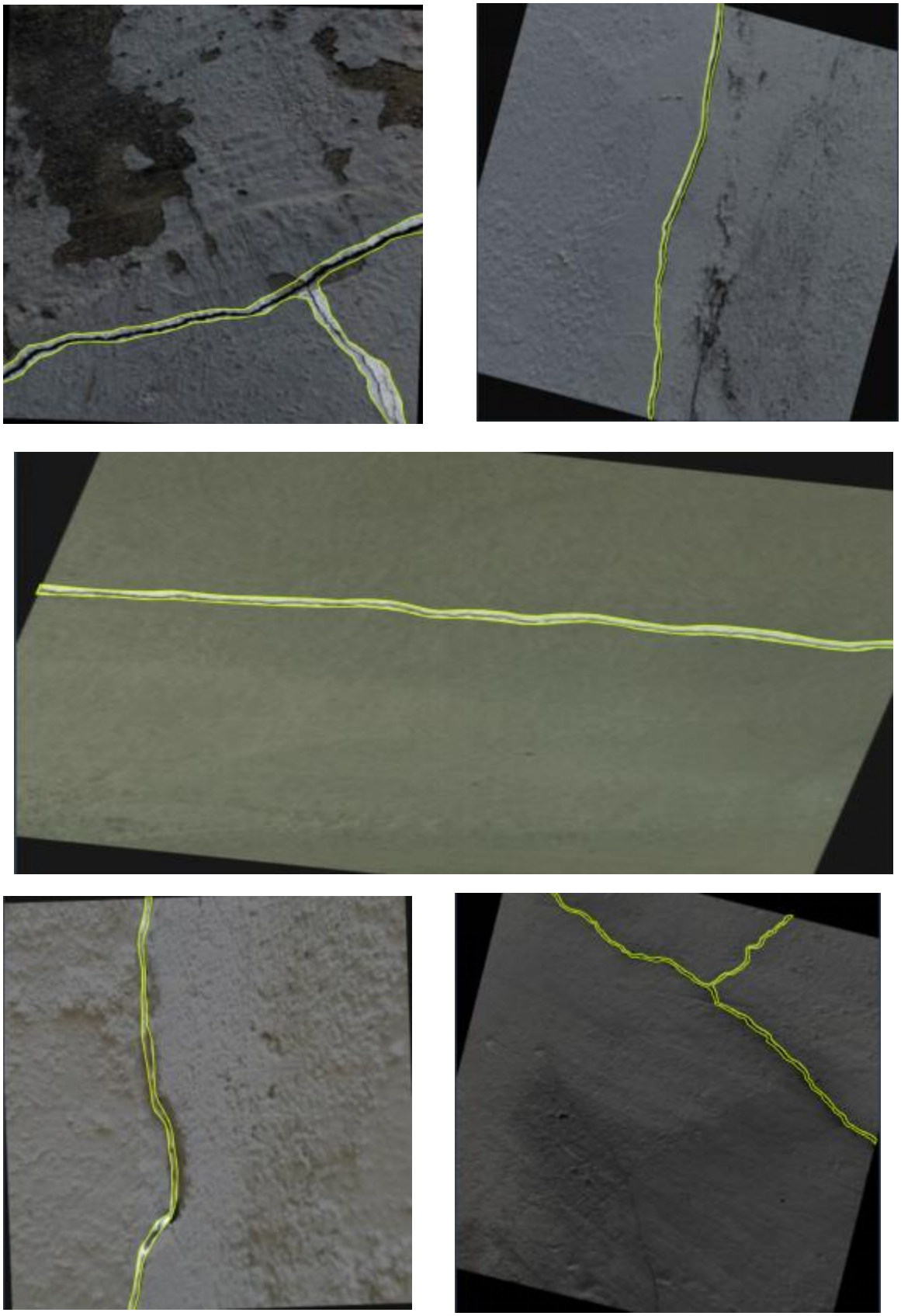

Figure 26. Annotated crack images [15]

4.2) IMAGE AUGMENTATION

The process of adding more photos to the training dataset by altering the current ones is known as image augmentation. The goal of image augmentation is to add diversity and variability to the training dataset, which enhances the generalization capacity of the model. YOLOv7 has access to a number of image augmentation methods. [42]

a) Random horizontal flipping: By randomly flipping the image horizontally, this strategy provides the model with additional instances of the same object in various orientations.
b) Random cropping: By cropping a section of the image at random, this strategy provides the model with more examples of the object in various scales and positions.
c) Random rotation: This method randomly rotates the image, creating more examples of object with different orientations
d) Random brightness and contrast: By randomly adjusting the image's brightness and contrast, this strategy gives the model additional examples of the object under various lighting scenarios.
e) Random color jitters: This method randomly changes the color and gives more examples of the object with different color variations

4.3) RASPBERRY PI

For the research purpose we use Raspberry Pi 4 as UAV. The Raspberry Pi foundation was founded in the UK and produces the cheap, credit card-sized single-board computer known as the Raspberry Pi. Because the graphics processing unit (GPU), random access memory (RAM), central processing unit (CPU), and other peripherals are all incorporated into a single circuit board, the computer is referred to as a single-board computer. It runs Raspbian, a 32-bit operating system based on Linux and distributed by the Debian project, in addition to having a 64-bit ARM CPU. Throughout the years, the Raspberry Pi has been introduced in a number of variants, including the Raspberry Pi 1, 2, 3, Zero, Raspberry Pi 3 model B, and Raspberry Pi 4.[53]

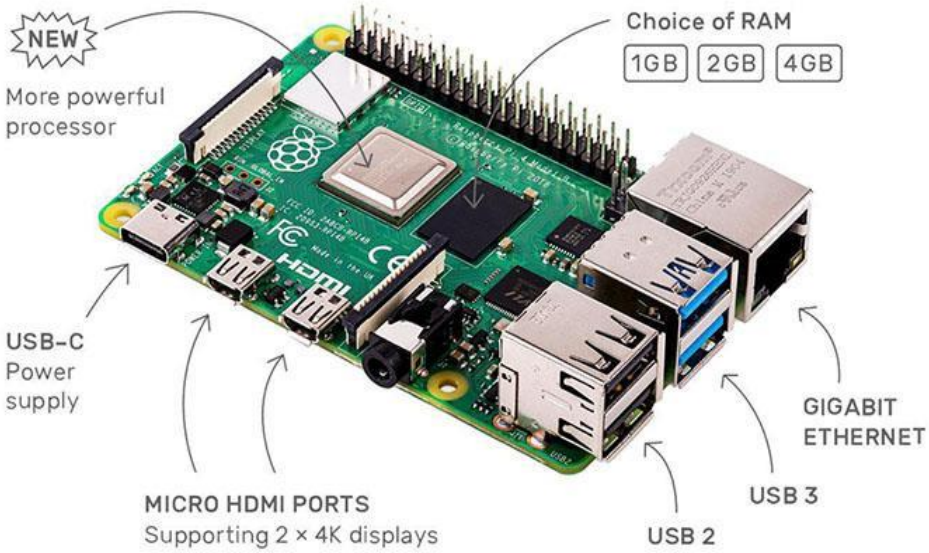

Figure 27. Raspberry Pi 4 [52]

PyTorch is used to train our model on Python. After that, we use ONNX to convert the model to Tensorflow Lite (TFLite), a Deep Learning framework that enables us to translate the model from Python to C++ so that it can be deployed into CPU-only ARM devices like our Raspberry Pi 4. TFLite comes along with several tools for quantization such as static quantization and dynamic-range quantization. In case of static quantization, INT8 is applied to the data based on a predetermined, set range while in dynamic-range quantization, each inference cycle, the range is recalculated. Because there is less chance of distortions during the data conversion to INT8, this could lead to a slower rate of inference but perhaps more accurate models. [54]

4.4) EVALUATION METRICS

The output values of the training results of YOLOv7 model are shown in Table1 and Table 2.

Epoch	GPU_mem	box_loss	seg_loss	obj_los	cls_loss	Instances	Size
0/29	4.2G	0.08575	0.04788	0.02064	0	17	640
1/29	5.22G	0.06195	0.02466	0.01764	0	7	640
2/29	5.22G	0.05993	0.0232	0.01499	0	10	640
3/29	5.22G	0.05702	0.02317	0.01369	0	5	640
4/29	5.22	0.0529	0.02371	0.01275	0	7	64
5/29	5.22	0.05087	0.02264	0.01218	0	5	64
6/29	5.22	0.04676	0.02339	0.01225	0	5	64
7/29	5.22	0.04417	0.02261	0.01213	0	6	64
8/29	5.22	0.04228	0.02183	0.01129	0	23	64
9/29	5.22	0.04117	0.02191	0.01126	0	13	64
10/29	5.22	0.04032	0.0215	0.0114	0	13	64
11/29	5.22	0.0386	0.02161	0.01088	0	11	64
12/29	5.22	0.03773	0.02188	0.01091	0	8	64
13/29	5.22	0.03603	0.02099	0.01087	0	11	64
14/29	5.22	0.03559	0.02089	0.01087	0	10	64
15/29	5.22	0.03354	0.02069	0.01054	0	14	64
16/29	5.22	0.03308	0.0211	0.01037	0	13	64
17/29	5.22	0.03137	0.02011	0.01005	0	14	64
18/29	5.22	0.03234	0.02095	0.01071	0	16	64
19/29	5.22	0.03048	0.02056	0.009961	0	6	64
20/29	5.22	0.0298	0.02056	0.009837	0	14	64
21/29	5.22	0.02965	0.02024	0.01036	0	19	64
22/29	5.22	0.02789	0.0204	0.009358	0	17	64
23/29	5.22	0.02754	0.01985	0.009533	0	7	64
24/29	5.22	0.02662	0.01986	0.009416	0	10	64
25/29	5.22	0.02554	0.0189	0.009384	0	10	64
26/29	5.22G	0.02477	0.01935	0.009169	0	12	640
27/29	5.22G	0.0242	0.01965	0.009118	0	12	640
28/29	5.22G	0.02338	0.01925	0.008669	0	6	640
29/29	5.22G	0.02219	0.01903	0.008626	0	7	640

Table 1:Training Results

Epoch	Class	Images	Inst	Box(P	R	mAP50	mAP50-95)	Mask(P	R	mAP50	mAP50-95)
0/29	all	312	398	0.169	0.618	0.347	0.0999	0.121	0.523	0.204	0.0434
1/29	all	312	398	0.479	0.56	0.472	0.16	0.475	0.555	0.419	0.112
2/29	all	312	398	0.581	0.565	0.426	0.175	0.559	0.513	0.364	0.092
3/29	all	312	398	0.492	0.487	0.404	0.134	0.295	0.352	0.213	0.0504
4/29	all	312	398	0.69	0.548	0.592	0.312	0.624	0.521	0.498	0.167
5/29	all	312	398	0.624	0.618	0.584	0.3	0.563	0.553	0.431	0.128
6/29	all	312	398	0.733	0.62	0.647	0.402	0.639	0.558	0.506	0.161
7/29	all	312	398	0.661	0.616	0.618	0.346	0.597	0.533	0.476	0.161
8/29	all	312	398	0.704	0.663	0.651	0.298	0.532	0.503	0.397	0.103
9/29	all	312	398	0.672	0.653	0.618	0.376	0.608	0.608	0.48	0.154
10/29	all	312	398	0.738	0.651	0.663	0.41	0.659	0.568	0.51	0.161
11/29	all	312	398	0.743	0.677	0.696	0.369	0.63	0.568	0.483	0.143
12/29	all	312	398	0.759	0.671	0.691	0.444	0.689	0.613	0.568	0.178
13/29	all	312	398	0.745	0.704	0.709	0.473	0.662	0.626	0.577	0.197
14/29	all	312	398	0.776	0.704	0.713	0.443	0.674	0.598	0.534	0.167
15/29	all	312	398	0.823	0.671	0.714	0.487	0.743	0.616	0.583	0.2
16/29	all	312	398	0.785	0.706	0.727	0.492	0.717	0.638	0.584	0.203
17/29	all	312	398	0.792	0.696	0.726	0.497	0.689	0.651	0.581	0.21
18/29	all	312	398	0.808	0.685	0.733	0.486	0.728	0.611	0.579	0.197
19/29	all	312	398	0.774	0.729	0.737	0.492	0.656	0.626	0.557	0.18
20/29	all	312	398	0.803	0.711	0.751	0.52	0.739	0.621	0.623	0.208
21/29	all	312	398	0.81	0.709	0.741	0.487	0.729	0.633	0.594	0.205
22/29	all	312	398	0.828	0.701	0.746	0.499	0.757	0.634	0.602	0.205
23/29	all	312	398	0.782	0.711	0.752	0.524	0.712	0.633	0.608	0.215
24/29	all	312	398	0.77	0.73	0.744	0.545	0.693	0.648	0.616	0.22
25/29	all	312	398	0.781	0.719	0.746	0.544	0.712	0.641	0.613	0.219
26/29	all	312	398	0.777	0.736	0.748	0.537	0.697	0.678	0.627	0.219
27/29	all	312	398	0.818	0.725	0.759	0.553	0.778	0.641	0.643	0.221
28/29	all	312	398	0.81	0.749	0.765	0.565	0.733	0.676	0.63	0.232
29/29	all	312	398	0.851	0.734	0.784	0.577	0.794	0.631	0.636	0.231

Table 2: Training Results

The output values of the performance results gotten from testing of YOLOv7 model are shown in Table 3.

Class	Images	Inst	Box(P	R	mAP50	mAP50-95)	Mask(P	R	mAP50	mAP50-95)
All	312	398	0.851	0.731	0.784	0.577	0.794	0.631	0.636	0.231

Table 3: Performance Result of YOLOv7

4.5) RESULTS AND ANALYSIS

1) Precision

For precision, comparing the results of Box and Mask from Table 3, it can be seen that Box outperforms Mask. Box's classes had 85.1% compared with Mask's having 79.4%. From the comparison, Box has more true positives to total number of detected cracks compared Mask by 5.7% difference in overall class detection. The model in this case will efficiently identify cracks and create the bounding boxes and the masks.

2) Recall

For the results of recall in Table 3, it can be seen that Box outperforms Mask with results of 73.1% and 63.6% respectively. Meanwhile, Box also has better recall compared to Mask.

3) Accuracy in Terms of mAP@0.5 and mAP@0.5:0.95

For mAP@0.5 and mAP@0.5:0.95, comparing the results in Table 3, it is seen that Box gave a better result in terms of accuracy than Mask with the overall class results in mAP@0.5 and mAP@0.5:0.95 of 78.4% and 57.7% compared with 63.6% and 23.1% of Mask's. With Box having mAP@0.5 of 14.8% difference compared with that of Mask's shows how well the model is able to rightly and accurately detect objects when compared with the ground truth objects

4) Conclusion

It is observed that the Box performs better than Mask for all the performance metrics. It is deduced that Box has better detection accuracy, precision and recall than Mask especially when used during production as deduced from the testing results. Overall the efficiency of the model is good with great accuracy.

Figure 28. Screenshot before segmentation [38]

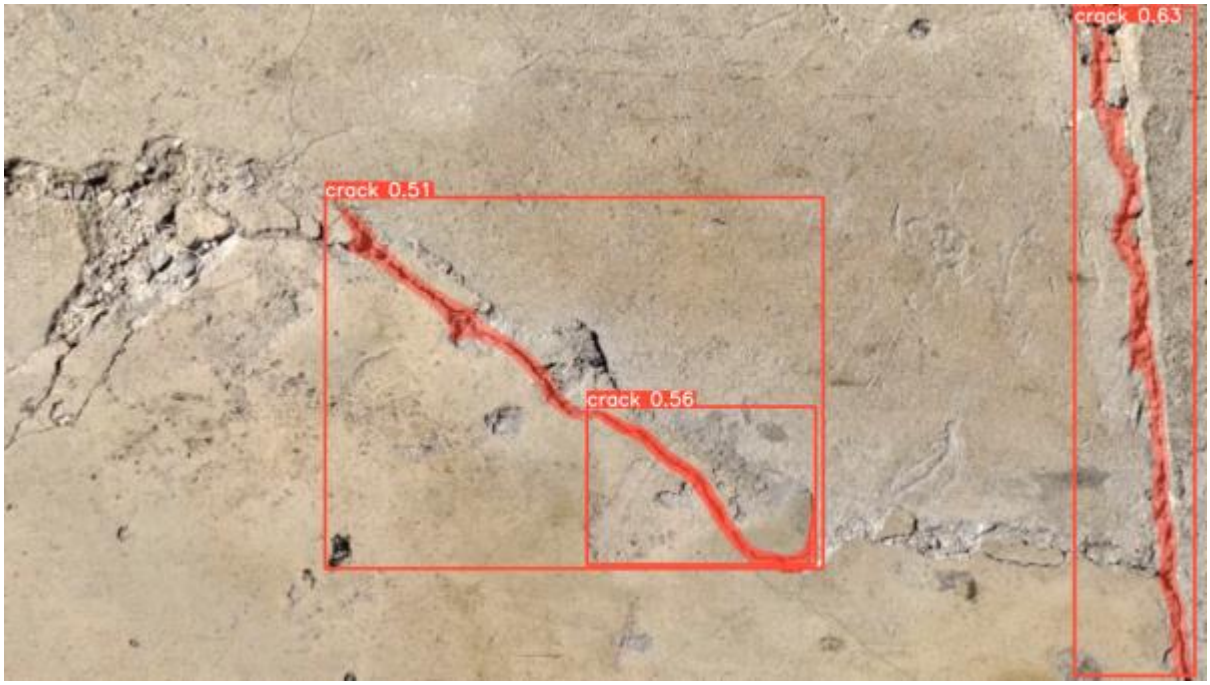

Figure 29. Screenshot after segmentation [38]

Figure 30. Screenshot before segmentation [39]

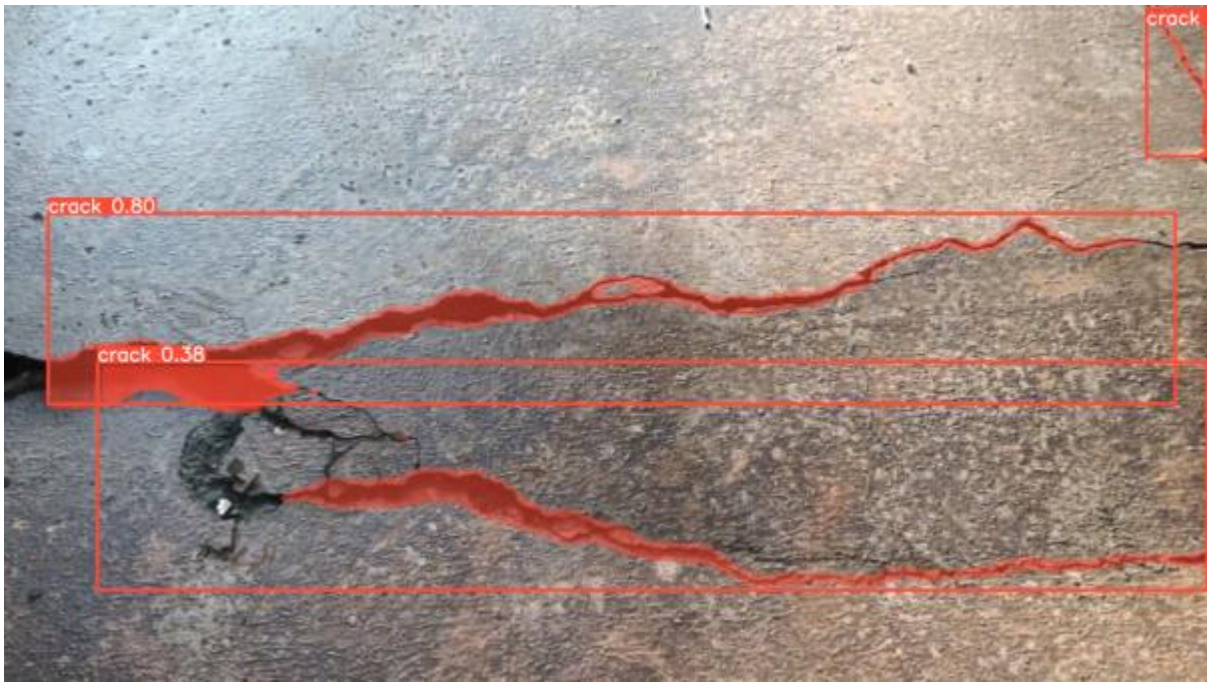

Figure 31. Screenshot after segmentation [39]

Figure 32. Screenshot before segmentation [40]

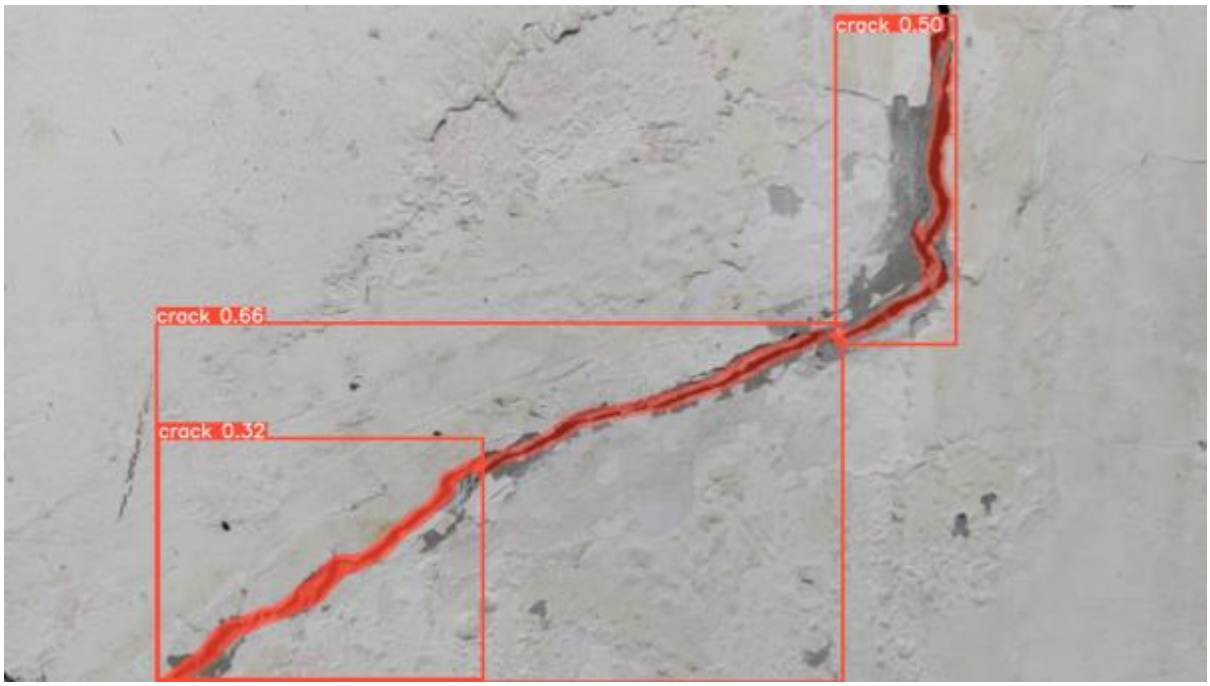

Figure 33. Screenshot aftersegmentation [40]

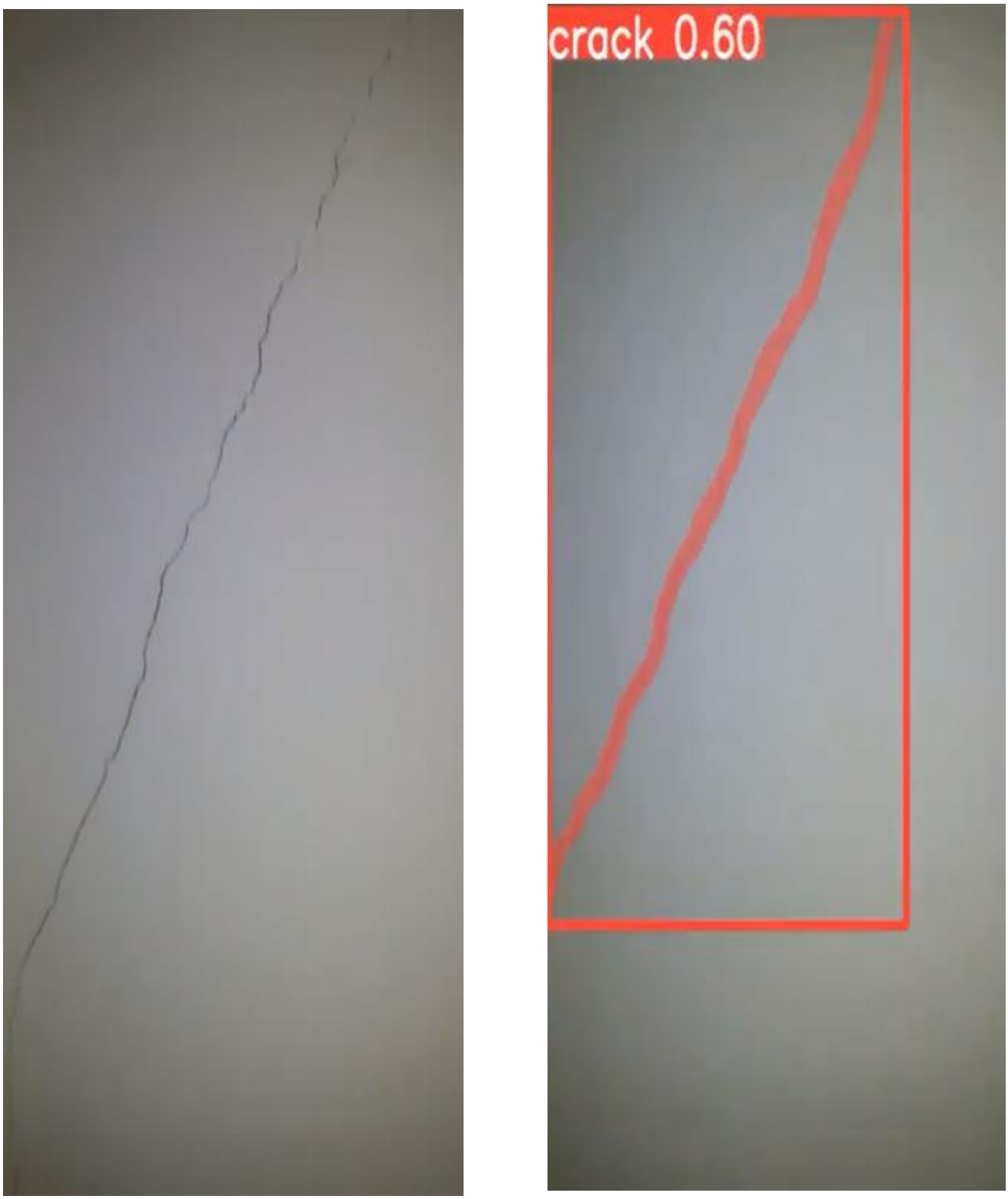

Figure 34. Screenshot before and after segmentation

As shown in Figures 28, 29, 30. 31, 32, 33 and 34, YOLOv7 was able to recognize and classify wall damage structures with accuracy.Even when there are little imperfections or unclear classes in the photos, the model accurately identifies the damages. Also, there are more bounding boxes surrounding the same area of damage, indicating that YOLOv7 tends to divide the damage class into smaller sub-regions.

In Figures 29 and 31 there are areas that model does not detect damage, possibly duo to occlusion or high box loss value.This indicates the need for additional development in the model's capacity to recognize and accurately construct bounding boxes surrounding such damage incidents.

5) CONCLUSIONS AND FUTURE EXTENSIONS

In conclusion, the project analyzes how deep learning can interact with UAVs to detect damage and cracks in building walls. Using the YOLOv7 model, which as we have seen with some changes can become a very good tool for the detection of damage in buildings using it on UAV, the research successfully achieved its goal using an architecture capable of detecting damage in building walls.

A significant contribution of this research was the huge annotated image variations of the dataset. This improved the damage detection of cracks on walls and reduced the difference in class for certain types of damage. The results of this research provide a significant addition to the field and open up new avenues for investigation. As seen in the section on results, our approach achieved high accuracy across all levels, having 85.1% and 79.4% precision, 73.1% and 63.6% recall results in the Box and the Mask respectively. The results are pretty good, but there is still room for development.

To further improve the performance, future studies can investigate various image kinds, such as multispectral images and LIDAR sensors. By combining this data, an embedded computer may be able to produce superior outcomes. Another improvement could be the combination of the YOLOv7 model with other models with the aim of identifying not only cracks but also large damages such as large holes in building walls or the detection of curvature of buildings. Also, an important improvement would be the categorization of the damage, so that the model could, depending on the size of the damage, color the area with a corresponding color or display an appropriate message when making the prediction

BIBLIOGRAPHY

[1] Olorunshola, Oluwaseyi Ezekiel, Martins Ekata Irhebhude, and Abraham Eseoghene Evwiekpaefe. "A Comparative Study of YOLOv5 and YOLOv7 Object Detection Algorithms." *Journal of Computing and Social Informatics* 2, no. 1 (February 8, 2023): 1 – 12. https://doi.org/10.33736/jcsi.5070.2023.

[2] Chen, Lingkun, Wenxin Chen, Lu Wang, Chencheng Zhai, Xiaolun Hu, Linlin Sun, Yuan Tian, Xiaoming Huang, and Lizhong Jiang. "Convolutional Neural Networks (CNNs)-Based Multi-Category Damage Detection and Recognition of High-Speed Rail (HSR) Reinforced Concrete (RC) Bridges Using Test Images." *Engineering Structures* 276 (February 2023): 115306. https://doi.org/10.1016/j.engstruct.2022.115306.

[3] Valipour, Parisa Setayesh, Amir Golroo, Afarin Kheirati, Mohammadsadegh Fahmani, and Mohammad Javad Amani. "Automatic Pavement Distress Severity Detection Using Deep Learning." *Road Materials and Pavement Design*, November 2023, 1 – 17. https://doi.org/10.1080/14680629.2023.2276422.

[4] Nihal, Ragib Amin, Benjamin Yen, Katsutoshi Itoyama, and Kazuhiro Nakadai. "From Blurry to Brilliant Detection: YOLOv5-Based Aerial Object Detection with Super Resolution." arXiv, January 26, 2024. http://arxiv.org/abs/2401.14661.

[5] Suo, Dajiang, and Sanjay E. Sarma. "A Test-Driven Approach for Security Designs of Automated Vehicles." In *2019 IEEE Intelligent Vehicles Symposium (IV)*, 26 – 32. Paris, France: IEEE, 2019. https://doi.org/10.1109/IVS.2019.8814172.

[6] Li, Pei, Bingyu Shen, and Weishan Dong. "An Anti-Fraud System for Car Insurance Claim Based on Visual Evidence." arXiv, April 30, 2018. http://arxiv.org/abs/1804.11207.

[7] Jayawardena, Srimal. (2013). Image Based Automatic Vehicle Damage Detection.

[8] Shin, Donghoon, Sachin Grover, Kenneth Holstein, and Adam Perer. "Characterizing Human Explanation Strategies to Inform the Design of Explainable AI for Building Damage Assessment." arXiv, November 4, 2021. http://arxiv.org/abs/2111.02626.

[9] Zhao, Shanshan, Mingming Gong, Xi Li, and Dacheng Tao. “Adaptive Edge-to-Edge Interaction Learning for Point Cloud Analysis.” arXiv, November 20, 2022. http://arxiv.org/abs/2211.10888.

[10] Wang, Chien-Yao, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. “YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors.” arXiv, July 6, 2022. http://arxiv.org/abs/2207.02696.

[11] Chen, Xian, Hongli Pu, Yihui He, Mengzhen Lai, Daike Zhang, Junyang Chen, and Haibo Pu. “An Efficient Method for Monitoring Birds Based on Object Detection and Multi-Object Tracking Networks.” *Animals* 13, no. 10 (May 22, 2023): 1713. https://doi.org/10.3390/ani13101713.

[12] Ramazhan, Muhammad Remzy Syah, Alhadi Bustamam, and Rinaldi Anwar. “Car Body Damage Detection System Using YOLOv7.” In *2023 3rd International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)*, 498–502. Yogyakarta, Indonesia: IEEE, 2023. https://doi.org/10.1109/ICE3IS59323.2023.10335254.

[13] Woo Sanghyun et al., "Cbam: Convolutional block attention module", Proceedings of the European conference on computer vision (ECCV), 2018.

[14] Zhang, Xin, and Daqing Huang. “Research on UAV Ground Target Detection Based on Improved YOLOv7.” In *2023 3rd International Conference on Computer, Control and Robotics (ICCCR)*, 28–32. Shanghai, China: IEEE, 2023. https://doi.org/10.1109/ICCCR56747.2023.10193961.

[15] Dwyer, B., Nelson, J. (2022), Solawetz, J., et. al. Roboflow (Version 1.0) [Software]. Available from https://roboflow.com computer vision.

[16] Shinde, Pramila P., and Seema Shah. “A Review of Machine Learning and Deep Learning Applications.” In 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), 1–6. Pune, India: IEEE, 2018. https://doi.org/10.1109/ICCUBEA.2018.8697857.

[17] Sheikh, Haroon, Corien Prins, and Erik Schrijvers. “Artificial Intelligence: Definition and Background.” In *Mission AI*, by Haroon Sheikh, Corien Prins, and Erik Schrijvers, 15–41. Research for Policy. Cham: Springer International Publishing, 2023. https://doi.org/10.1007/978-3-031-21448-6_2.

[18] Lv, Pin, Qinjuan Wu, Jia Xu, and Yating Shu. “Stock Index Prediction Based on Time Series Decomposition and Hybrid Model.” *Entropy* 24, no. 2 (January 19, 2022): 146. https://doi.org/10.3390/e24020146.

[19] Raubitzek, Sebastian, and Thomas Neubauer. “An Exploratory Study on the Complexity and Machine Learning Predictability of Stock Market Data.” *Entropy* 24, no. 3 (March 2022): 332. https://doi.org/10.3390/e24030332.

[20] “Time Series Data Analysis of Stock Price Movement Using Machine Learning Techniques | Soft Computing.” Accessed February 21, 2024. https://link.springer.com/article/10.1007/s00500-020-04957-x.

[21] “Entropy | Free Full-Text | A Hybrid Method Based on Extreme Learning Machine and Wavelet Transform Denoising for Stock Prediction.” Accessed February 21, 2024. https://www.mdpi.com/1099-4300/23/4/440.

[22] Ayala, Jordan, Miguel García-Torres, José Luis Vázquez Noguera, Francisco Gómez-Vela, and Federico Divina. “Technical Analysis Strategy Optimization Using a Machine Learning Approach in Stock Market Indices.” *Knowledge-Based Systems* 225 (August 5, 2021): 107119. https://doi.org/10.1016/j.knosys.2021.107119.

[23] Sahu, Santosh Kumar, Anil Mokhade, and Neeraj Dhanraj Bokde. “An Overview of Machine Learning, Deep Learning, and Reinforcement Learning-Based Techniques in Quantitative Finance: Recent Progress and Challenges.” *Applied Sciences* 13, no. 3 (January 2023): 1956. https://doi.org/10.3390/app13031956.

[24] Afzal, A. L. and S. Asharaf. “Deep Learning in Kernel Machines.” (2019).

[25] Rashmi, Ritika Sharma, Priyanshu Sharma, and Shipra Mangal. “Ethical Consideration in AI & Machine Learning.” *Industrial Engineering Journal* 52, no. 05 (2023): 1543–51. https://doi.org/10.36893/IEJ.2023.V52I5.1543-1551.

[26] Deng, Li. “Deep Learning: Methods and Applications.” *Foundations and Trends® in Signal Processing* 7, no. 3–4 (2014): 197–387. https://doi.org/10.1561/2000000039.

[27] Obthong, Mehtabhorn, Nongnuch Tantisantiwong, Watthanasak Jeamwatthanachai, and Gary Wills. “A Survey on Machine Learning for

Stock Price Prediction: Algorithms and Techniques:" In *Proceedings of the 2nd International Conference on Finance, Economics, Management and IT Business*, 63 – 71. Prague, Czech Republic: SCITEPRESS – Science and Technology Publications, 2020. https://doi.org/10.5220/0009340700630071.

[28] Schmidhuber, Jürgen. "Deep Learning in Neural Networks: An Overview." *Neural Networks* 61 (January 2015): 85 – 117. https://doi.org/10.1016/j.neunet.2014.09.003.

[29] "Recurrent Neural Network (RNN)." Accessed February 22, 2024. https://machine-learning.paperspace.com/wiki/recurrent-neural-network-rnn.

[30] "Long Short-Term Memory Networks (LSTM)- Simply Explained! | Data Basecamp," June 4, 2022. https://databasecamp.de/en/ml/lstms.

[31] Balaji, Sai. "Binary Image Classifier CNN Using TensorFlow." *Techiepedia* (blog), August 26, 2023. https://medium.com/techiepedia/binary-image-classifier-cnn-using-tensorflow-a3f5d6746697.

[32] Bochkovskiy, Alexey, Chien-Yao Wang, and H. Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection." *ArXiv*, April 23, 2020. https://www.semanticscholar.org/paper/2a6f7f0d659c5f7dcd665064b71e7b751592c80e.

[33] Bangar, Siddhesh. "VGG-Net Architecture Explained." *Medium* (blog), June 28, 2022. https://medium.com/@siddheshb008/vgg-net-architecture-explained-71179310050f.

[34] Mukherjee, Suvaditya. "The Annotated ResNet-50." Medium, August 18, 2022. https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758.

[35]Datagen. "ResNet: The Basics and 3 ResNet Extensions." Accessed February 22, 2024. https://datagen.tech/guides/computer-vision/resnet/.

[36] ResearchGate. "The Architecture of MobileNet V1 [30]." Accessed February 22, 2024. https://www.researchgate.net/figure/The-architecture-of-MobileNet-V1-30_fig5_357623745.

[37] ResearchGate. "Fig. 2. The Architecture of Faster R-CNN." Accessed February 22, 2024. https://www.researchgate.net/figure/The-architecture-of-Faster-R-CNN_fig2_324903264.

[38] Vecteezy. “Download Road Surfaces from Cracked Cement or Damaged by Earthquakes or Prolonged Use. for Free.” Accessed February 22, 2024. https://www.vecteezy.com/video/17630541-road-surfaces-from-cracked-cement-or-damaged-by-earthquakes-or-prolonged-use.

[39] Vecteezy. “Download Cracked Concrete Ground Broken at Floor Home or Street Road Subside from Earthquake for Free.” Accessed February 22, 2024. https://www.vecteezy.com/video/11463693-cracked-concrete-ground-broken-at-floor-home-or-street-road-subside-from-earthquake.

[40] Vecteezy. “Download Cracked Concrete Ground Broken at Old Wall from Bad Construction or Earthquake for Free.” Accessed February 22, 2024. https://www.vecteezy.com/video/19200846-cracked-concrete-ground-broken-at-old-wall-from-bad-construction-or-earthquake.

[41] Qiu, Zifeng, Huihui Bai, and Taoyi Chen. “Special Vehicle Detection from UAV Perspective via YOLO-GNS Based Deep Learning Network.” *Drones* 7, no. 2 (February 2023): 117. https://doi.org/10.3390/drones7020117.

[42] Silva, Luís Augusto, Valderi Reis Quietinho Leithardt, Vivian Félix López Batista, Gabriel Villarrubia González, and Juan Francisco De Paz Santana. “Automated Road Damage Detection Using UAV Images and Deep Learning Techniques.” *IEEE Access* 11 (2023): 62918–31. https://doi.org/10.1109/ACCESS.2023.3287770.

[43] Böyük, Mustafa, Ramazan Duvar, and Oğuzhan Urhan. “Deep Learning Based Vehicle Detection with Images Taken from Unmanned Air Vehicle.” In *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*, 1–4, 2020. https://doi.org/10.1109/ASYU50717.2020.9259868.

[44] Kang, Dong H, and Young-Jin Cha. “Efficient Attention-Based Deep Encoder and Decoder for Automatic Crack Segmentation.” *Structural Health Monitoring* 21, no. 5 (September 1, 2022): 2190–2205. https://doi.org/10.1177/14759217211053776.

[45] Choi, Wooram, and Young-Jin Cha. “SDDNet: Real-Time Crack Segmentation.” *IEEE Transactions on Industrial Electronics* 67, no. 9 (September 2020): 8016–25. https://doi.org/10.1109/TIE.2019.2945265.

[46] ResearchGate. “Figure 2. YOLOv3 Architecture.” Accessed February 23, 2024. https://www.researchgate.net/figure/YOLOv3-architecture_fig2_350502286.

[47] OpenGenus IQ: Computing Expertise & Legacy. “YOLO v5 Model Architecture [Explained],” October 28, 2022. https://iq.opengenus.org/yolov5/.

[48] Wang, Xueli, Xingtao Zhuang, Wei Zhang, Yunfang Chen, and Yanchao Li. “Lightweight Real-Time Object Detection Model for UAV Platform.” In *2021 International Conference on Computer Communication and Artificial Intelligence (CCAI)*, 20–24, 2021. https://doi.org/10.1109/CCAI50917.2021.9447518.

[49] Nene, Vidi. “Deep Learning-Based Real-Time Multiple-Object Detection and Tracking via Drone,” August 2, 2019. https://dronebelow.com/2019/08/02/deep-learning-based-real-time-multiple-object-detection-and-tracking-via-drone/.

[50] Sun, Chenfan, Wei Zhan, Jinhiu She, and Yangyang Zhang. “Object Detection from the Video Taken by Drone via Convolutional Neural Networks.” *Mathematical Problems in Engineering* 2020 (October 13, 2020): e4013647. https://doi.org/10.1155/2020/4013647.

[51] Boesch, Gaudenz. “YOLOv7: A Powerful Object Detection Algorithm (2024 Guide).” viso.ai, November 21, 2023. https://viso.ai/deep-learning/yolov7-guide/.

[52] The Pi Hut. “A Comprehensive Guide to the Raspberry Pi 4.” Accessed February 26, 2024. https://thepihut.com/blogs/raspberry-pi-roundup/the-comprehensive-guide-to-the-raspberry-pi-4.

[53] Qasim, Hamzah H., Ali M. Jasim, and Khalid A. Hashim. “Real-Time Monitoring System Based on Integration of Internet of Things and Global System of Mobile Using Raspberry Pi.” *Bulletin of Electrical Engineering and Informatics* 12, no. 3 (June 1, 2023): 1418–26. https://doi.org/10.11591/eei.v12i3.4699.

[54] Liberatori, Benedetta, Ciro Antonio Mami, Giovanni Santacatterina, Marco Zullich, and Felice Andrea Pellegrino. “YOLO-Based Face Mask Detection on Low-End Devices Using Pruning and Quantization.” In *2022 45th Jubilee International Convention on Information, Communication and Electronic Technology (MIPRO)*, 900–905, 2022. https://doi.org/10.23919/MIPRO55190.2022.9803406.