REPUBLIC OF TURKEY
OSTİM TECHNICAL UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF SOFTWARE ENGINEERING
SOFTWARE ENGINEERING
MASTER'S PROGRAM

DIGITAL TWIN OF UAV: IMAGE PROCESSING IN DIFFERENT VIRTUAL ENVIRONMENTS

MASTER'S THESIS

PREPARED BY
SAMET AKÇAY

THESIS SUPERVISOR
ASSIST. PROF. DR. İCLAL ÇETİN TAŞ

ANKARA-2023

THESIS ACCEPTANCE AND APPROVAL

This study, titled "DIGITAL TWIN OF UAV: IMAGE PROCESSING IN DIFFERENT VIRTUAL ENVIRONMENTS" and submitted by SAMET AKÇAY on 30/05/2023, was found successful as a result of the thesis defense on 15/06/2023 and accepted as a Master's Thesis by our jury.

Date of Approval:................/......./................

Jury Member:	**Assist. Prof. Dr. İclal ÇETİN TAŞ** Ostim Technical University	____________
Jury Member:	**Assist. Prof. Dr. Didem ÖLÇER** Başkent University	_______________
Jury Member:	**Assist. Prof. Dr. Ahmet Anıl MÜNGEN** Ostim Technical University	_______________
Thesis Supervisor:	**Assist. Prof. Dr. Iclal ÇETİN TAŞ** Ostim Technical University	_______________

APPROVAL

I hereby approve that this study, accepted by the jury, fulfills the requirements for being a Master's Thesis.

…../…../20….

____

Assist. Prof. Dr. Hikmet BAL
Institute Direct

DECLERATION

I hereby declare that I have given Ostim Technical University the permission to archive all or any part of my Master's thesis approved by the Institute in printed or digital format and make it accessible under the conditions specified below. With this permission, all intellectual property rights other than the usage rights granted to the University will remain with me and the usage rights of all or a part of my thesis for future studies (article, book, license, patent, etc.) will belong to me alone.

I declare and undertake that my thesis is entirely my own work, that I do not violate the rights of others and that I am the sole authorized owner of my thesis. I undertake that I use the copyrighted resources with written permission, which must be used with written permission from their owners, and to submit copies of the permissions to the University upon request.

Within the scope of the "Directive on Collecting, Organizing and Opening to Access The Theses and Dissertations in Electronic Environment"" published by the Council of Higher Education, my thesis is made available on the YÖK National Thesis Center and Ostim Technical University Open Access System, except for the following conditions.

☒ With the decision of the Institute / Faculty Administrative Board, the open access of my thesis has been postponed for 2 years from my graduation date.[1]

☐ With the reasoned decision of the Graduate School / Faculty Administrative Board, the opening of my thesis has been postponed for maximum of 6 months from the date of my graduation.[2]

☐ A confidentiality decision has been made regarding my thesis.[34]

Date:

Signature:

---

[1] ARTICLE 6(1) In the event that a patent application is made for a graduate thesis or the patenting process is ongoing, upon the recommendation of the thesis advisor and the approval of the graduate school department, the graduate school or faculty executive board may decide to postpone the access to the thesis for two years.

[2] AR 6(2) For theses that use new techniques, materials and methods, that have not yet turned into articles or are not protected by methods such as patents, and that contain information and findings that may create unfair gain for third parties or institutions if shared on the internet, the thesis may be prevented from being accessed for a period not exceeding six months upon the recommendation of the thesis advisor and the approval of the institute department, with the reasoned decision of the institute or faculty board of directors.

[3] ARTICLE 7(1) The decision of confidentiality regarding graduate theses related to national interests or security, security, intelligence, defense and security, health, etc. is made by the institution where the thesis is conducted. The confidentiality decision regarding graduate theses prepared within the framework of a cooperation protocol with institutions and organizations is made by the university board of directors upon the recommendation of the relevant institution and organization and the approval of the institute or faculty. Theses for which a confidentiality decision is made are notified to the Council of Higher Education.

[4] ARTICLE 7(2) Theses for which a confidentiality decision has been made are kept by the institute or faculty within the framework of confidentiality rules during the confidentiality period, and are uploaded to the Thesis Automation System if the confidentiality decision is lifted.

OSTİM TECHNICAL UNIVERSITY

INSTITUTE OF NATURAL AND APPLIED SCIENCES MASTER'S THESIS ORIGINALITY REPORT

Thesis Title: Digital Twin of UAV: Image Processing In Different Virtual Environments

Student Name Surname: Samet Akçay

Thesis Supervisor Title Name Surname: Assist. Prof. Dr. İclal Çetin Taş

Department: Software Engineering

Program: Software Engineering

Date: 30/05/2023

The part of my Master's thesis, which consists of Introduction, Main Chapters and Conclusion and consists of 62 pages in total, was examined by my thesis advisor and myself on 30/05/2023 by the turnitin plagiarism detection program.

According to the originality report, the similarity score of my thesis is 9%.

Applied filters:

1. Bibliography (excluding)

2. Citations (excluding)

3. Parts of text with less than five (5) words overlapping (excluding)

I declare that my thesis does not contain any plagiarism; that I accept all kinds of legal responsibility that may arise in the event that the contrary is detected and that the information I have given above is correct.

Student Signature:………………….

APPROVAL

Date: …. / …. /…..

Assist. Prof. Dr. İclal Çetin Taş

ETHICAL DECLERATION

I declare that this study is an original study, that I act in accordance with scientific ethics and rules in the preparation, data collection, analysis, presentation of information and all other stages of the study, that I have obtained all the document information in the study within the framework of academic ethics and rules, that I have presented all visual, audio, and written information and results in accordance with the scientific rules of ethics, that I have not made any falsifications in the data I have used, that I have referred to the sources I have used in accordance with scientific norms, that my thesis has been written by me and is original, except for the cases where I cited sources, that it has been produced by me under the supervision of my thesis advisor Assist. Prof. Dr. İclal Çetin Taş and written in accordance with Ostim Technical University thesis writing guide.

Samet Akçay

30.05.2023

To my beloved family

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervisor Assist. Prof. Dr. İclal Çetin Taş for her support, advice, guidance, and encouragement throughout the research. Similarly, I would like to special thanks to Assist. Prof. Dr. Rayan Abri for his kindly support.

I am also grateful to my university to giving me that opportunity to conduct this valuable research. I wish also thank to Assist. Prof. Dr. Yücel Tekin and Assist. Prof. Dr. Hikmet Bal for their support and advice in the university.

Lastly and forever, I would like to thank my wonderful wife and love Filiz Akçay for her support and motivation to me all the time. Also, I would like to thank my sweet daughter Asude İpek Akçay who makes me happy and smile. Especially, I would like to thank my father and mother for their unending supports.

30.05.2023
Samet AKÇAY

ÖZ

Yazar Adı ve Soyadı	: Samet Akçay
Üniversite	: OSTİM Teknik Üniversitesi
Enstitü	: Fen Bilimleri Enstitüsü
Program Adı	: Yazılım Mühendisliği
Tezin Türü	: Yüksek Lisans
Sayfa Sayısı	: 62
Tarihi	: 2023

İHA'NIN DİJİTAL İKİZİ: FARKLI SANAL ORTAMLARDA GÖRÜNTÜ İŞLEME

Dijital ikiz kavramı son yıllarda giderek önemini arttırmakta ve bu alanda yapılan akademik çalışmalar yaygınlaşmaktadır. Dijital ikiz kavramı otomasyon, sanal gerçeklik, ve havacılık alanında özellikle insansız hava araçlarında kullanılmaktadır. Test süreçlerindeki maliyetler ve riskler göz önüne alındığında dijital ikize bağlı simülasyon ortamlarının geliştirilmesi kaçınılmazdır. Bu çalışmada, dijital ikiz yaklaşımı kullanılarak çok rotorlu mikro insansız hava aracı tasarlanmıştır. Tasarlanan bu araç, oyun geliştirme motoru olan Unreal yazılımıyla oluşturulan farklı ortam koşullarındaki sanal ortamda uçurulmuş ve üzerindeki sanal kamera yardımıyla görüntü elde edilmişitir. Bu görüntüler, derin öğrenme teknikleri kullanan YOLO mimarisi ile işlenerek yapay sinir ağları eğitilmiş. Eğitilen model ile birbirinden farklı nesneler üzerinde nesne tespit çalışması yapılmıştır. Bu sayede farklı ortam koşullarının farklı nesnelerin tespitindeki etkileri incelenmiştir.

Anahtar Sözcükler: İHA, Dijital ikiz, Görüntü işleme, Unreal oyun motoru

ABSTRACT

Thesis	: Samet Akçay
University	: OSTİM Technical University
Institute	: Graduate School of Natural and Applied Sciences
Program’s Name	: Software Engineering
Thesis Type:	: Master
Pages	: 62
Year	: 2023

DIGITAL TWIN OF UAV: IMAGE PROCESSING IN DIFFERENT VIRTUAL ENVIRONMENTS

The concept of the digital twin has gained increasing importance in recent years, and academic studies in this field have become widespread. The digital twin concept finds applications in automation, virtual reality, and aviation, particularly in unmanned aerial vehicles (UAVs). To address the costs and risks associated with testing processes, the development of simulation environments connected to the digital twin is inevitable. In this study, a digital twin approach was employed to design a multi-rotor micro-unmanned aerial vehicle. The designed vehicle was tested in a virtual environment created using the Unreal game development engine, simulating various environmental conditions. The vehicle was equipped with a virtual camera, allowing for image acquisition. These images were processed using the YOLO architecture, leveraging deep learning techniques to train artificial neural networks for object detection. Tests conducted with the different objects and the different environment are investigated with its effects.

Keywords: UAV, Digital Twin, Image processing, Unreal engine

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

A/C	Aircraft
Adam	Adaptive Moment Estimation
AI	Artificial Intelligence
API	Application Programming Interface
BLDC	Brushless Direct Current Motor
DCM	Direction Cosine Matrix
DoF	Degree of Freedom
DT	Digital Twin
CAD	Computer-Aided Design
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSP	Cross-stage-partial
FW	Fixed-Wing
GPU	Graphical Processing Unit
GUI	Graphical User Interface
IMU	Inertial Measurement Unit
MAV	Micro Aerial Vehicle
MOSFET	Metal Oxide Semiconductor Field Effect Transistor
MTOW	Maximum Takeoff Weight
NASA	National Aeronautics and Space Administration
3D	Three Dimensional
PAN	Path Aggregation Network
PID	Proportional Integral Derivative
PR	Precision / Recall
PS	Physical Signal
PWM	Pulse with Modulation
R-CNN	Region Based Convolutional Neural Network
ROI	Region of Interest
SLAM	Simultaneous Localization Mapping
SPP	Spatial Pyramid Pooling
SSD	Single Shot Multibox Detector
SVM	Support Vector Machine
UAV	Unmanned Aerial Vehicle
VFX	Visual Effects

VTOL	Vertical Takeoff and Landing
XML	Extensible Markup Language
YOLO	You Only Look Once
hr	Hour
kg	Kilogram
km	Kilometer
m	Mass
cg	Center of Gravity
g	Gravitational acceleration
F_B	Body fixed reference frame
X_B, Y_B, Y_B	Coordinates relative to the F_B
O_B	Origin of the F_B
F_E	Earth fixed reference frame
O_E	Origin of the F_E
X_E, Y_E, Z_E	Coordinates relative to the F_E
ϕ, θ, ψ	Euler angles (roll, pitch, and yaw)
R_X, R_Y, R_Z	Transformation matrix from F_E to F_B
I_B	Inertia matrix in F_B
I_{XX}, I_{YY}, I_{ZZ}	Moment of inertia in x, y, z axis
F_b	Force vector acting on F_B
F_X, F_Y, F_Z	Scalar force components of F_b
Fg	Gravity vector
F_T	Thrust force vector
F_1, F_2, F_3, F_4	Thrust forces belongs to four motor
F_D	Drag force
ρ	Air density
S	Propeller reference area
C_D	Drag coefficient
V	velocity
M_{XYZ}	Moment vector
L, M, N	Moment components in x, y, z axis
p, q, r	Angular velocity rates in roll, pitch, yaw
u, v, w	Scalar velocity components in F_B
T	Thrust
Q	Torque
k_f, k_t	Thrust and torque coefficients

Ω	Revolution per minute (rpm)
σ_x	Angle measured from IMU in x axis
Ac_x, Ac_y, Ac_z	Acceleration measured from IMU in x, y, z axis
Mag_x	Magnetometer angle in x axis
Mag_y	Magnetometer angle in y axis
Mag_z	Magnetometer angle in z axis
C_T	Propeller thrust coefficient
n	Revolution per second
D	Propeller diameter
Kp, Ki, Kd	Proportional, integral, derivative gain constants
u(t)	Output value
e(t)	Error value
m_ι	Moving average gradient
$\beta1$, $\beta2$	Decay rates
∇	Gradient
E	Effective magnitude
θ_ι	Update parameters
v_ι	Moving average square gradient
α	Learning rate
ϵ	Epsilon parameter

1. INTRODUCTION

In the last decade, a significant number of projects were carried out related to Unmanned Aerial Vehicles (UAVs). With the technological advancements of electronic and sensor devices, the usage of UAVs has further increased in the military, commercial, and research areas [1]. They are divided into three essential categories according to their flight types and abilities. These are multi-rotor, fixed-wing (FW), and vertical take-off and landing drones (VTOL). These types gain importance according to the task. For example, fixed-wing drones are effective for wide-area scanning and long flight times, while multi-rotor drones are more effective for missions that require hovering and agile movement [2].

On the other hand, UAVs are frequently preferred by the academic community in image processing, mapping, smart agriculture, and other applications, and many academic publications are made. Autonomous applications are developed specially by applying image processing, and artificial intelligence techniques to images obtained from drones. UAVs are also used for digital twin (DT) applications widely. Many engineering domains can be represented with DT to be able to perform more realistic analysis, data accusation, failure detection, and so on [3]. With the help of next-generation game engines, high-fidelity physical modeling tools t can be more cost-effective and easier to simulate real-world applications in a digital twin world.

The first of the basic components of this thesis is to create a digital twin model of the quadcopter (a special type of multi-rotor UAV with 4 rotors). The second phase is the process of creating simulations of real environments in which the drone can fly. In the last component, the images obtained from these environments are analyzed using deep learning architectures for object detection. Thus, object detection installations in different environments can be modeled and implemented virtually. With the approaches proposed in this thesis, it is predicted that more efficient and high-performance steps can be taken for many critical processes such as defense industry, agriculture, concentration and fire extinguishing.

In the following sections, digital twin, quadcopter mathematical modeling stages, virtual environment design, image processing processes related to deep learning and object recognition, and holistic results of the study will be explained in detail.

2. LITERATURE REVIEW

In recent years, UAVs are frequently used for very different applications in terms of increasing usage areas and obtaining efficient and qualified data. Considering their technical capabilities, these devices should be examined as two sub-fields as software and hardware. In addition, the slightest mistake to be made with these devices will lead to bad results. For this reason, simulation used for many products in the field of aviation is used in the design and quality control processes of these devices. The design and use of digital twins is also important in this respect. Simulating the environmental conditions in the real environment is also very critical in order to be able to analyze the performance and capabilities of digital twins in depth. In this way, possible undesirable situations can be prevented to a certain extent. The main components of this study are the creation of digital twins, the creation of different environmental conditions, and the object recognition process from the images taken using digital twins in these environmental conditions, as close to the real thing. The studies in the literature have been discussed many times separately in these sub-fields. However, studies synthesizing these topics are limited. Since it is a newly developing field, it is seen that the studies continue to mature. Most of the work in which the auctions are analyzed is done in the virtual environment. Cost of development in virtual environment helps researchers to study and try many cases without risk and economy. For example, to setup digital environment for UAVs which tools can be used and how that environment can be created is mentioned at see [4]. Another great study is mentioned that introduce the digital twin industry in aerospace [5]. Meyer et al. [6] investigate how to model a way of digital twin usage effectively in aviation. Their aims are draw relevant data and characteristics to improve the digital twin correct usage. Also, digital twin can be adjusted with the cloud system like 5G network to improve UAV based virtual reality technologies [7]. When studying on aerospace, sensors are a common faced area to be understand clearly. Alvarez et al. [8] shows how air data sensor model is simulated with digital twin one. Therefore, any failure accrued in the real sensor data the twins of that system can be aid or correct the system directly. In the test area, evaluation of UAV state and be aware before the failure occur are really important and can be easily performed with the support of digital twin [9]. For the virtualization environment, their effect and importance are given at [10]. In that study a tool AirSim which based on the unreal engine helps the developer to build UAV flying environment and test their algorithm. It can be both tested UAV dynamics and image processing works.

To mention a little bit for the image processing and object detection, how researchers develop algorithms and what type of frameworks are using on the field is important. For example, Biyik et al. [11] use the convolutional neural network (CNN) for aerial photographs taken from UAVs. They find the equivalent of the objects in the global coordinate system with an accuracy of 16.8 cm using the YOLO-v4 CSP model. Sharma et al. [12] use UAV images for 3D localization and real-time drone detection using stereo depth estimation and tiny YOLO-v4 object detection. They reached an accuracy of 77% for multiple drone detection in real time. Another study related to target localization estimation from UAV images shows that implementing the right CNN can get up to 79.5% accuracy [13]. Chen et al. [14] also mentioned that YOLO-based UAV applications called YBUT has increased usage area and aim to expand decision-making systems more reliable for real-time applications. Kao et al. [15] researched their study that Crack edge actual magnitude detection in risky bridges has indicated high precision almost 92% with the help of artificial intelligence and UAV images. Using UAV's search and rescue domain is highly crucial. Caputo et al. [16] show how to rescue humans in a dangerous area as much as faster and more reliably with an average accuracy of 84%. Another different study was done by Zhu et al. [17]. In that work, they implement a Darknet59-based detection algorithm under specific light conditions and gain up to 87.71% accuracy to detect the tiny object in high altitudes.

In this study, it is aimed to combine all these platform and conduct test. To prepare UAV dynamics the Matlab Simscape module is used. For obtain video image and fly in some different environment, the power of the Unreal engine is used. Finally, images are collected and trained with the YOLOv4 neural network framework.

3. MATERIALS AND METHODS

In that part all the development and design steps are showed. Dynamical modelling of the quadcopter with the Simscape, virtual environment design in the Unreal engine, and neural network training steps for object detection based on YOLOv4 framework are done. Additionally, digital twin concept is defined here.

3.1. Digital Twin

Digital twins especially come to the fore in the fields of aviation and automation that came with Industry 4.0. Digital twin means making a copy of a system with exactly the same functional features. In this way, both data related to that system are collected and functional tests are made in the real environment to be tested on the twin of that system before. Therefore, development costs and risks are eliminated. According to the National Aeronautics and Space Administration (NASA) [18], DT is defined as "A Digital Twin is an integrated multi-physics, multi-scale, probabilistic simulation of an as-built vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin.'' For the overall view, three main areas are considered from the perspective of DT as shown in Figure 1.

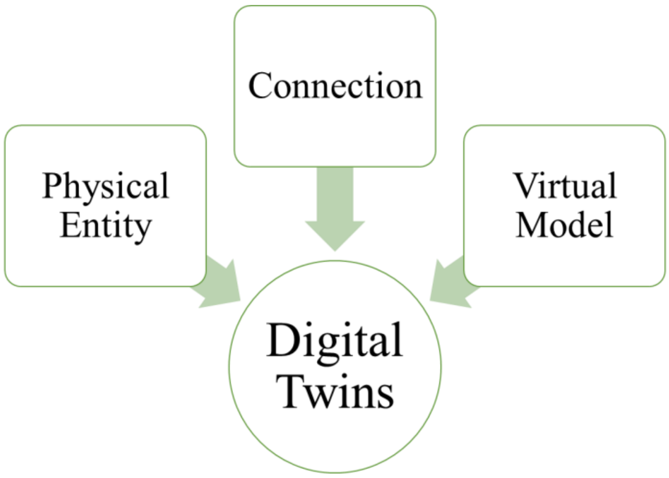

Figure 1. Digital twin main parts

To be able to act and perform with high accuracy as a real physical entity, data from an entity needs to be connected with a virtual model. Data acquisition systems can be used for this purpose [19]. On the other hand, virtual models can be designed with simulation tools such as game engines or model-based design tools. Digital Twin’s hot topics were performed in a variety following of fields [6].

- Fault detection
- Maintenance
- Manufacturing
- Model-based systems engineering
- Data collection.

In the factory, it can be a really quick and simple way to visualize the production line and estimate failure before that happens. In the aviation industry, many aircraft (A/C) can be failed due to sensor reading or estimation faults. But, thanks to the DT, engineers can create A/C dynamic model or any sensor estimation model virtually and make fault detection or correction on the model and fly safely [8].

3.1.1. Quadcopter Reference Model

UAVs can be defined by their classification and types. As shown in Table 1, there is a wide range of UAVs from micro-scale to tactical ones related to characteristic features [20]. Categories of micro UAVs are also called micro aerial vehicles (MAVs). These types of vehicles are highly fast-moving machines and have low maximum takeoff weight (MTOW). Therefore, they are used in restricted places such as caves, indoor and crowded cities.

Table 1. Unmanned aerial vehicle classes

Category	NASA Class	Weight (kg)	Altitude (m)	Range (km)	Endurance (hr)
Micro		<2	<140	5	<1
Mini	Class 1	2-20	<1000	25	2-8
Small		20-150	<2000	50	4-12
Tactical	Class 2	>150	<3300	>200	8-20

In addition to their classification, installation types of UAVs have become more crucial. Multi-copter are special forms of UAVs. Their symmetrical rotating arms count and positions are an indication of their type. From the perspective of the arm count shown in Figure 2, a tri-copter, a quadcopter, and a hexacopter have respectively three, four, and six arms.

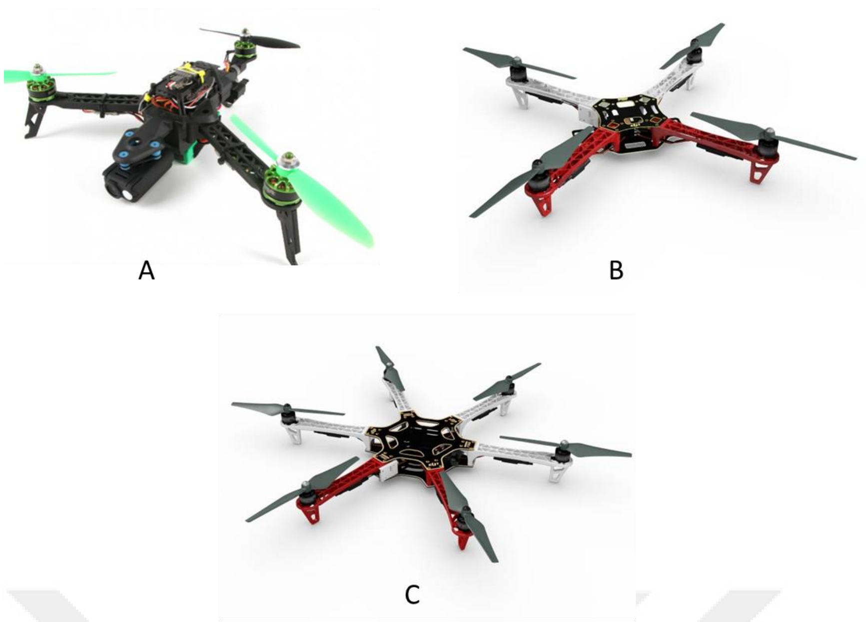

Figure 2. Multi-copter types; A) Tri-copter [21] B) Quadcopter [22] C) Hexacopter [22]

Another identical design criterion for UAVs is flight direction. There is a convention that describes the quad style as a Quad + type or Quad X type as shown in Figure 3. All motor control algorithms development and motor numbering are related to that convention.

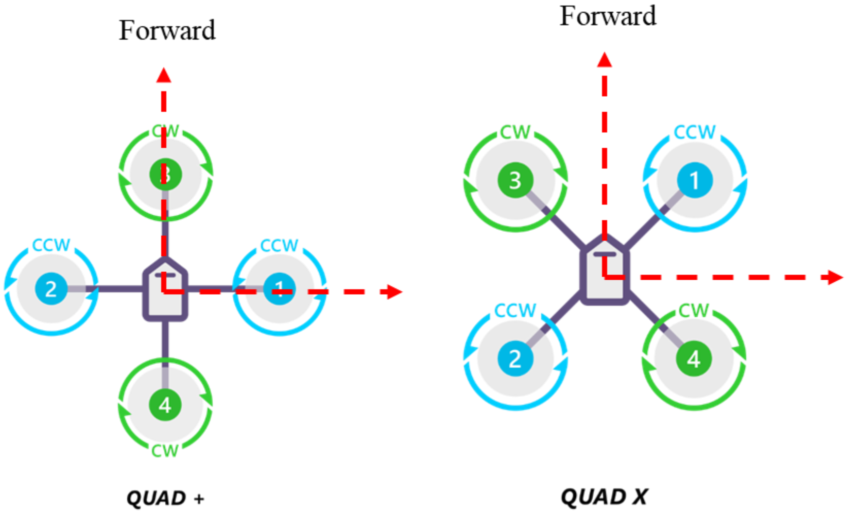

Figure 3. Quadcopter flight direction layout [23]

The reason that all these definitions are related to multi-rotor is the importance of frame. Because the frame is one of the most important parts of a multi-copter. An entire mathematical model is based on it. Arms, motors, and other electronics are attached to it. The frame should be carefully selected and rigid. It is the main part of carrying all the loads.

3.1.1.1. Quadcopter reference frame

In this part, basic concepts of quadcopter flight dynamics such as 6 DoF (Degree of freedom) equations, reference frames, Euler transformations, and other conventions are described. A quadcopter has four rotary motors that produce forces and torques. It moves under these effects forward-backward, right-left, up-down, and rotations. The outline of that movements comes from user inputs or autonomously. Quadcopters can fly by following a trajectory or manually. Based on the velocity, position, and acceleration inputs, it can adjust its control action internally with the help of autopilot board calculations. To accomplish that dynamic model, it needs references such as body, inertial, and earth frame. Quad X type is selected for the dynamic model and reference system preparations.

3.1.1.2. Body reference frame F_B (Body Axes, O_B, X_B, Y_B, Z_B)

Center of the gravity (cg) is represented as the O_B origin of the quadcopter. The forward direction of the quadcopter X_B can be defined towards the nose of a quadcopter. The axis lies in the symmetry plane of a quadcopter to the downside called Z_B and the Y_B axis is placed in the coordinate system by the rule of the right-hand side. The configuration of a quadcopter body reference frame is illustrated in Figure 4.

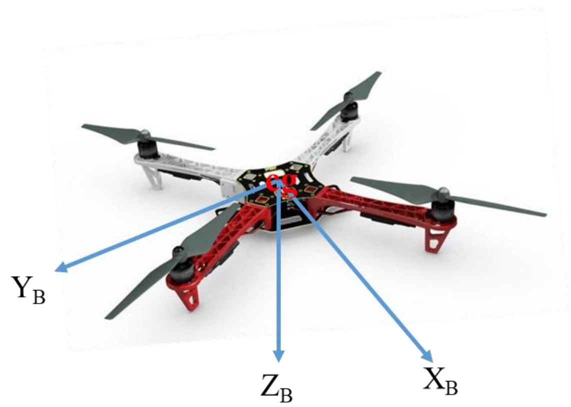

Figure 4. Demonstration of quadcopter body reference frame

3.1.1.3. Earth reference frame FE (Earth Axes, O_E, X_E, Y_E, Z_E)

Under low-speed applications such as UAV dynamics; inertial frames, and earth frames can be performed similarly in general. The rotation of the Earth relative to the inertial frame is negligible. As shown in Figure 5, anywhere on the surface of the Earth act as the origin of Earth due to the low speed. Therefore, O_E is the origin of that reference system. The axis moves toward the Earth's center, which is Z_E. The axis along with the North direction is X_E and lying on the East direction axis is called Y_E

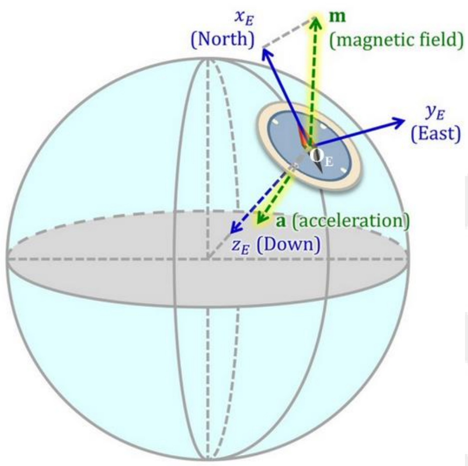

Figure 5. Earth reference frame [24]

3.1.1.4. Euler angles (ϕ, θ, ψ)

An object in three-dimensional (3D) space can be represented in any different orientation with Euler angles. To be able to get quadcopter orientation in angle form, Euler rotation angles, ϕ, θ, ψ, known roll, pitch, and yaw are used respectively. That way it can be easy to gather the equation of motion of a quadcopter with the relationship of the body and Earth fixed reference frame. Euler rotation angles layout with the reference axis is shown in Figure 6.

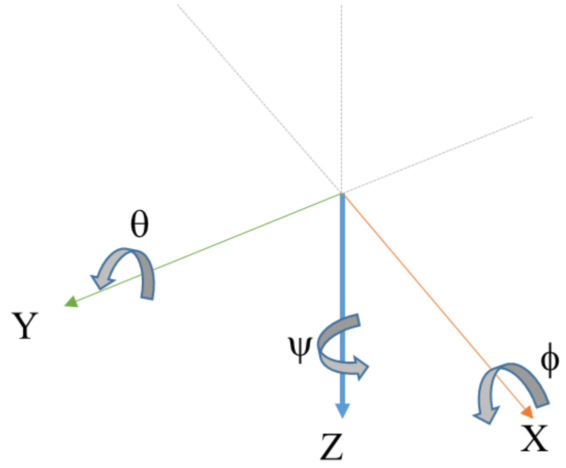

Figure 6. Euler angles

3.1.2. Quadcopter Kinematic Model

In order to understand and implement the dynamic equation of quadcopter, how trust, velocities and force conversion, and also reference frames relation are understood clearly. In Figure 7 inertial frame (as supposed Earth frame) and quadcopter body frame relation and other forces and moments are indicated briefly.

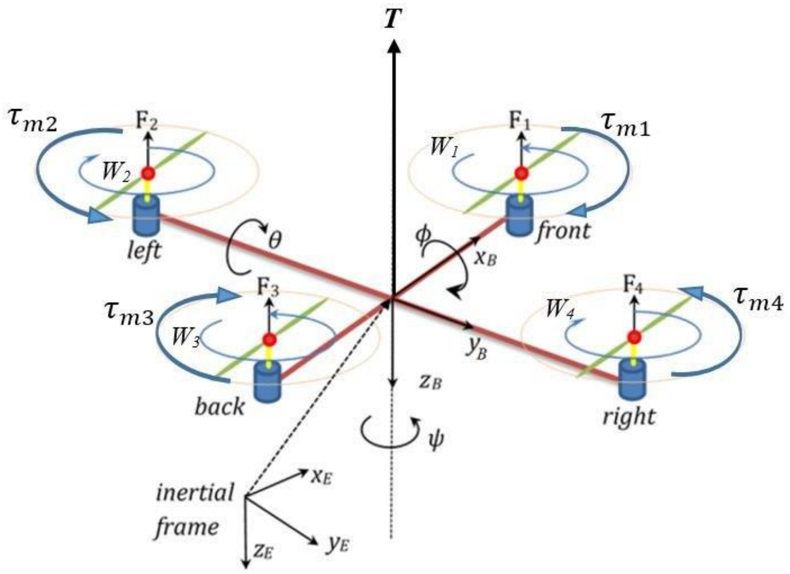

Figure 7. Inertial and body reference frames, forces, and moment acting on a quadcopter [25]

Gravity acting on a quadcopter always points downside, but has a vector component. To find the right equation and overcome vector component effects, it can be defined rotation relation. Therefore, rotation matrix must be defined and calculated as indicated in equation (3.1).

$$R_X(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix}, \qquad R_Z(\theta) = \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix},$$

$$R_Z(\psi) = \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

(3.1)

Full conversion between the body frame and the inertial frame, R_X, R_Y, and R_Z matrixes multiplied together. R_X is the rotation matrix in roll, R_Y is in the pitch, and R_Z is in the yaw axis. The sin (x) and cos (x) are the sinus and cosine respectively. The practical way is to convert the final rotation along with the motor thrust direction for earth-to-body transformation [26]. In equation (3.2), the rotation matrix which is also the same as the direction cosine matrix (DCM) is given.

$$R_{ZYX}(\psi, \theta, \phi) = \begin{bmatrix} c\theta c\psi & c\psi s\theta s\phi - s\psi c\phi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - c\psi s\phi \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}$$

(3.2)

Where s = sinus and c = cosine. R_{ZYX} is the direction cosine matrix from the body frame to the inertial (earth) frame. As an example of roll direction cosine matrix calculation can be done in equation (3.1) and the graphical representation is given in Figure 8.

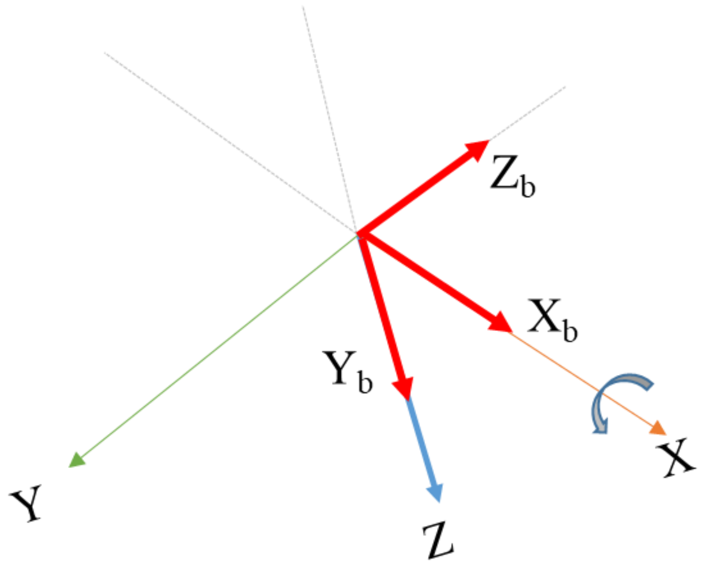

Figure 8. +90 degrees roll angle around X axis

With +90 degrees roll rotation Y_B and Z_B body axis orientations are changed. Thus, direction cosine matrix can be written according to the total rotation matrix.

$$R_X(\phi = +90) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

(3.1)

A selected frame of the quadcopter is assumed symmetrical [27], therefore inertia matrix can be indicated as given in equation (3.2).

$$I_b = \begin{bmatrix} I_{XX} & 0 & 0 \\ 0 & I_{YY} & 0 \\ 0 & 0 & I_{ZZ} \end{bmatrix}$$

(3.2)

Where I_{XX}, I_{YY}, and I_{ZZ} moments of inertia are based on the X, Y, and Z axis respectively. I_{XX} and I_{YY} moment of inertia are equal. In addition, if the angular rate around the Z-axis is small enough, the angular momentum effect on the inertia moments can be negligible [28]. To understand forces, and torques on quadcopters Newton's and Euler's laws should be defined carefully [29]. These forces are generated by rotating motors. In equation (3.3) linear forces are defined.

$$F_b = \begin{bmatrix} F_X \\ F_Y \\ F_Z \end{bmatrix} = m \, . \begin{bmatrix} \ddot{X}_b \\ \ddot{Y}_b \\ \ddot{Z}_b \end{bmatrix}$$

(3.3)

Where F_b is the forces acting on the quadcopter body-fixed frame and indicated as force matrix F_X, F_Y, and F_Z; m is the total mass, and $\ddot{X}_b$, $\ddot{Y}_b$, and $\ddot{Z}_b$ are the acceleration on those axes respectively. Another force acting naturally on the quadcopter and pulling down to it in the Z direction is the gravity vector called Fg. Gravity vector is not directly on the body axis, thus it should be transformed to the body-fixed frame by multiplying with the vector of DCM as shown in the equation (3.4).

$$F_g = m \, . R . \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}$$

(3.4)

Where Fg is the gravitational vector, R is the DCM, and m is the mass. The force produced by propellers acts in the same direction as the Z axis. Propellers are assumed rigid body means there is no flapping or flexibility. This thrust force is dependent on revolution per minute (rpm) and propeller dimensions. Thrust force is figured out in the equation (3.5).

$$F_T = \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix}$$

(3.5)

Drag forces act on the quadcopter when it flies. The drag force's direction is opposite to the moving direction. It is generated by the natural effect of friction in the air [30]. Drag force can be implemented as shown in equation (3.6).

$$F_D = \frac{1}{2} \, . \rho \, . V^2 \, . S \, . C_D$$

(3.6)

Where F_D is the drag force, ρ is the air density, V is the velocity, S is the reference area, and C_D is the drag coefficient. Moments acting on the center of mass are defined in equation (3.7)

$$M_{XYZ} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = I \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} x \, I_b \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

(3.7)

Where M_{XYZ} is the total moment matrix, L, M, and N moments on X, Y, and Z axis respectively. p, q, and r angular rates in roll, pitch, and yaw. Euler angles and angular rates relation is given below,

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\phi c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

(3.8)

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & s\phi tan\theta & c\phi tan\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi sec\theta & c\phi sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

(3.9)

Where tan = tanjant and sec = secant.

Assume the quadcopter is rigid, and symmetric in the x-z plane final equation of motion mathematical model can be implemented as follows,

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw & -gs\theta & \frac{X}{m} \\ pw - ru & gs\phi c\theta & \frac{Y}{m} \\ qu - pv & gc\phi c\theta & \frac{Z}{m} \end{bmatrix}$$

(3.10)

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{I_{XZ}}{I_{XX}}(\dot{r} + pq) & \frac{I_{YY} - I_{ZZ}}{I_{XX}}qr & \frac{L}{I_{XX}} \\ \frac{I_{XZ}}{I_{YY}}(r^2 - p^2) & \frac{I_{ZZ} - I_{XX}}{I_{YY}}pr & \frac{M}{I_{YY}} \\ \frac{I_{XZ}}{I_{ZZ}}(\dot{p} - qr) & \frac{I_{XX} - I_{YY}}{I_{ZZ}}pq & \frac{N}{I_{ZZ}} \end{bmatrix}$$

(3.11)

3.1.3. Propulsion Model

Rotating propellers are the most important parts for generating force and torque for quadcopters. To calculate thrust, the correlation factor between thrust and torque should be implemented. Basically, the torque and thrust model created by the propulsion part can be modeled as follows.

$$T = k_f \Omega^2$$

(3.12)

$$Q = k_t \Omega^2$$

(3.13)

Where T and Q are the thrust and torque, as well as Ω, k_f, and k_t are the rpm, thrust, and torque coeffientiens respectively.

3.1.4. Inertial Measurement Units

An inertial measurement unit (IMU) is a system that completely models the inertial navigation system to calculate vehicle dynamics. It measures data about 3D space dynamics such as magnetometer, accelerometer, and gyroscope information on the three-axis. Gyroscope is able to estimate the angular velocities of the vehicles. It protected against noise but can drift [31]. That drift issue can be corrected with the reference point. The angle can be evaluated by integrating angular velocity gathered IMU as below.

$$\sigma_X = \int \dot{\sigma_X} dt$$

(3.14)

The accelerometer part inside the IMU can calculate both gravitational and dynamic acceleration. Its data highly depended on the white Gaussian noise [32]. It measures perpendicular vectorial components to the earth. The absolute rotation can be indicated in the below equation.

$$\sigma_X = arctan \frac{Ac_X}{\sqrt{{Ac_Y}^2 + {Ac_Z}^2}}$$

(3.15)

The magnetometer inside the IMU calculates the magnetic field of Earth. Its measured field has an angle to the gravitational field of 90 degrees. Magnetometer measurements aid the accelerometer's lack of orientation. It does not directly indicate angles but can be evaluated as below.

$$\sigma_X = arctan \frac{Mag_X}{\sqrt{{Mag_Y}^2 + {Mag_Z}^2}}$$

(3.16)

3.1.5. Quadcopter Physical Modeling with Matlab Simscape

The Simulink environment becomes an efficient tool for constructing physical system models with the use of Simscape. In Simscape, physical component models are designed through direct physical connections with block diagrams and other modeling paradigms. Many models of various systems, including actuators, electric motors, and mechanical systems, by assembling components into a schematic. To further increase the complexity of the components and analytical capabilities, Simscape add-on products are available. The main components of Simscape are given in Figure 9.

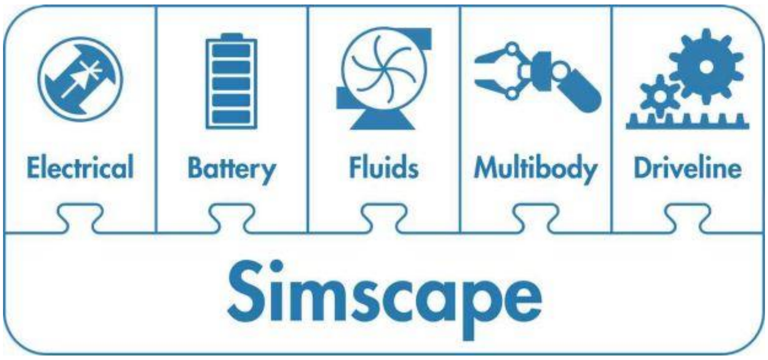

Figure 9. Matlab Simscape toolbox main components [33]

A brief explanation of that component is below.

- Simscape: Multi-domain Physical system simulation and modeling.
- Battery: Modeling the energy storage capabilities of various battery types.
- Driveline: Rotational and translational mechanical representation model.
- Electrical: Mechanic, electronic, and electrical properties tools.
- Multibody: Design multibody systems.

With the help of Simscape tools, it can be easy to visualize any system with its part. Additionally, Simscape can be used as a digital twin environment.

3.1.5.1. CAD model translation

In Simscape Computer-Aided Design (CAD) model can be inserted equivalent Simscape blocks. Simscape uses extensible markup language (XML) format to represent the multibody components. It supports both import and export operations. In export parts, CAD files are converted to an XML markup language file, and additional STL or STEP file. The process of the conversation is given in Figure 10.

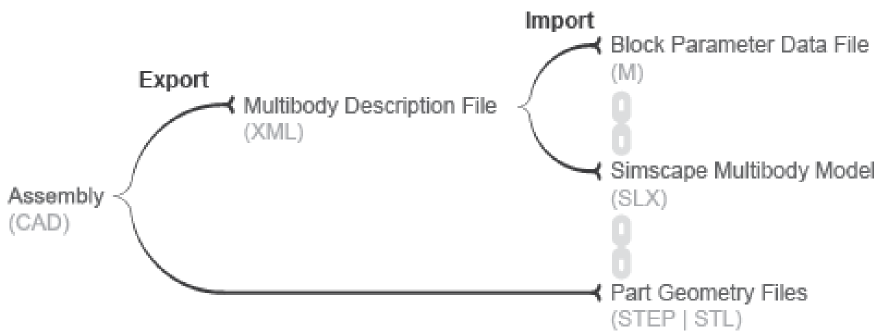

Figure 10. CAD translation steps [34]

To share CAD files with the Simscape environment Simscape multibody link add-ons should be installed. Inside The SolidWorks CAD software environment, it can be added inside the toolbar top of the graphical user interface (GUI) in the tools segment. After clicking on the add-on button it can be found there. By the way, successfully passing this stage, CAD files can be exported to the Simscape environment. Now, inside the tools section, there is a multibody link segment appeared. The export process is given in Figure 11.

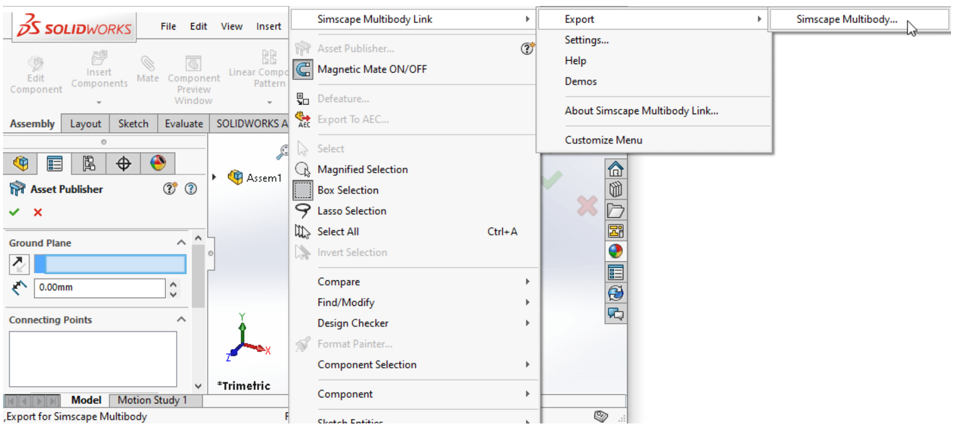

Figure 11. Export SolidWorks CAD files to Simscape Multibody [35]

To represent solid parts of the quadcopter geometries, materials, and visual properties coming from CAD software, solid element components of the Simscape with properties derived from an external file should be used. An example of the quadcopter arm with solid external file representation is given in Figure 12. With that feature, inertia and material properties can be easily modeled and calculated. Also, it can be updated immediately when any change occurred the CAD model.

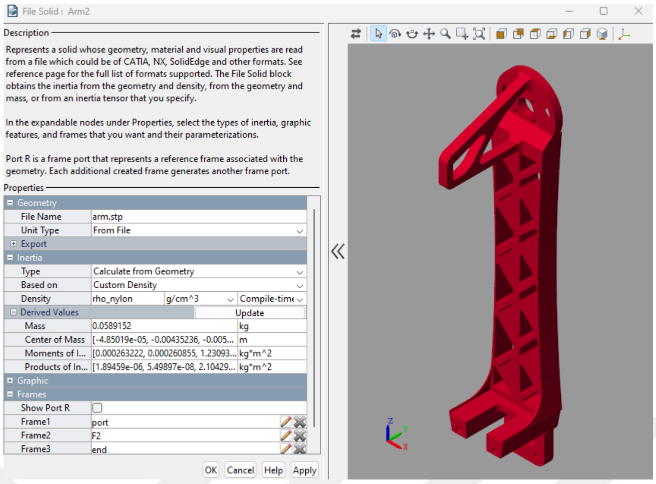

Figure 12. Solid arm external source model

All CAD geometries for this work based on the quadcopter were collected from the GrabCad community [36]. With a solid modeling perspective, any vehicle can be modeled with high fidelity and it gives a detailed view of the model to users. Also importing the model should be an issue. Every part may need inspection or correction carefully. In Figure 13, the brushless direct current motor (BLDC), propeller, and cap parts solid integration with the Simscape is given.

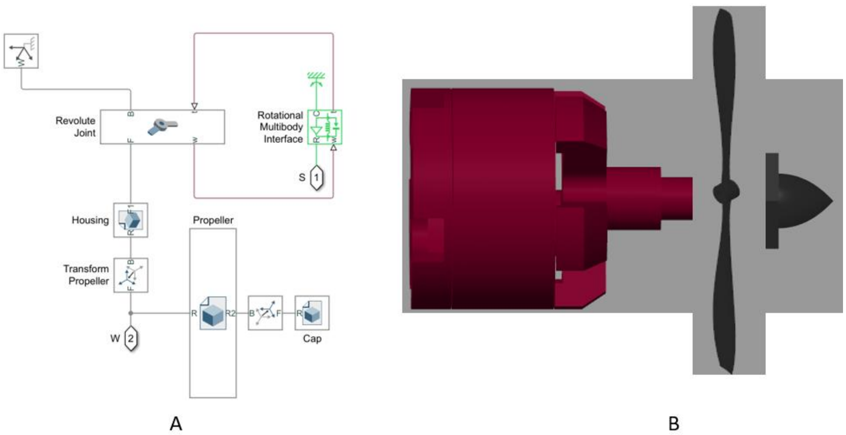

Figure 13. Multibody integration of motor and propeller of the quadcopter.

As seen in Figure 13, how easy to understand and convert solid modeling into a simulation environment. The revolute joint indicates the rotation relation between the motor and propeller. It's connected with world frame reference properties to get active together with the reference object added before it. Transform properties are for the orientation of the solid object. Every single part has a solid part. After the propeller parts, again transform object is added to simulate the orientation of the cap relative to the propeller.

3.1.5.2. Propeller model

The propeller is one of the most critical parts of the quadcopter. It generates thrust force and moves the quadcopter in the air. Thus it should be calculated correctly. The thrust force mostly depends on the propeller's rotating speed, thrust coefficient, propeller diameter, and environment [37]. The mathematical calculation of the thrust force for the propeller is given below.

$$T = C_T \boldsymbol{\rho} n^2 D^4 \tag{3.17}$$

Where T is Thrust force, C_T is thrust coefficient, ρ is the air density, n is the revolution per second, and D is the propeller's diameter.

Table 2. Quadcopter propeller properties

Parameters	**Values**
Thrust Coefficient (C_T)	0.1072
Diameter (D)	0.254 m
Density(ρ)	1.225 kg/m3
Weight	25 grams

With reference to Table 2 propeller data, thrust can be calculated. The simscape model for that calculation is given in Figure 14.

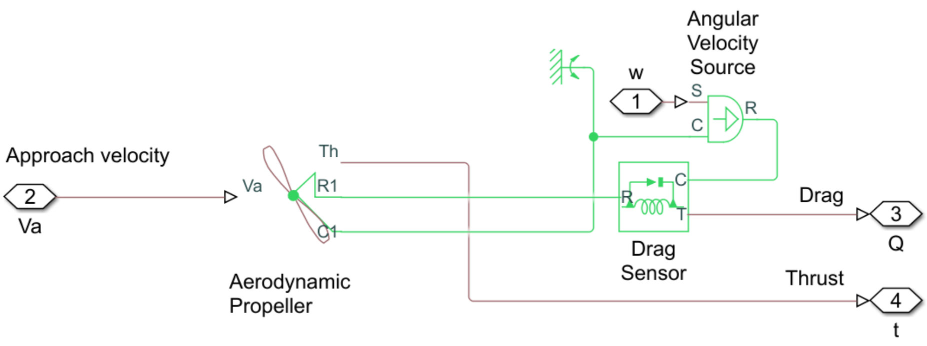

Figure 14. Simscape propeller thrust calculation model

The aerodynamic propeller block is responsible for the thrust calculation. Va is the advanced velocity coming into the propeller. With the help of a drag sensor block, the ideal drag force can be calculated directly.

3.1.5.3. Brushless electrical motor model

Many applications like quadcopters need motor components. Brushless direct current motors come at that point due to their reliability and effectiveness. In BLDC there is no contact surface like brushed one. Thus, there are no wearing or over-heated parts. These motors are really high-performance, fast, and have smaller form factors in terms of torque and speed. The simple model of BLDC motors is given in Figure 15.

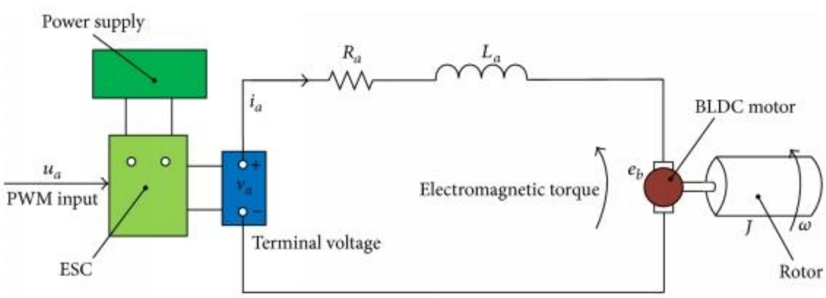

Figure 15. BLDC motor model [38]

Pulse-with-modulation (PWM) signal is applied to the electronic speed controller (ESC) and the controller drives the magnetic motor [39]. In BLDC, motor torque ripples are the most common problem to solve with the control solution [40]. Additionally, motor inertia and other motor properties can be affected by the model of BLDC. The motor and its specifications are given in Figure 16 and Table 3.

Figure 16. 2212 Brushless Motor [41]

Table 3. 2212 Brushless Motor Specification

Parameters	**Values**
Motor Kv	920 rpm/Volt
Stator Diameter	22 mm
Stator Thickness	12 mm
Weight	56 grams
Stator Slots Number	12
Magnet Poles Number	14
No Load Current	0.5 Amps
Internal Resistance	0.015 Ohms
Max. Continuous Power	155 Watts @ 3S Li-Po

In the Simscape blocks, the BLDC motor can be modeled from scratch or used directly by a ready-to-use BLDC item. A scratch model of BLDC is given in Figure 17.

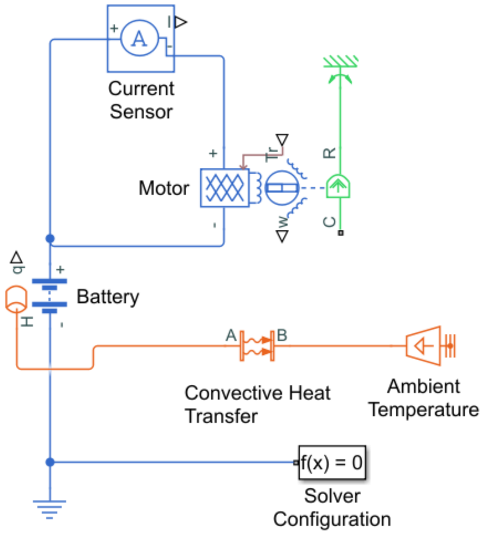

Figure 17. BLDC model from scratch.

Where users can add ambient temperature and connect with the heat transfer block to indicate the temperature effect on the battery. Battery block gives the energy to the system and simulates the voltage source in the system and is connected to the ground source. Motor block act as simple electrical motor behavior. A current sensor can be added to the system to measure current and send it to the control part. Detail BLDC motor model is given in Figure 18.

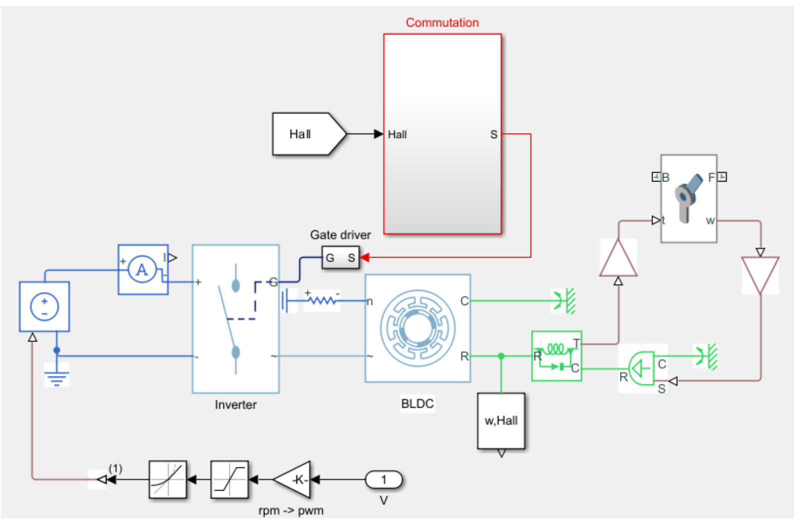

Figure 18. Detail BLDC model.

Where the Simulink signal to the physical signal (PS) converter is used to convert desired driving velocity to the PWM signal and sent to the voltage source. BLDC model indicated an electrical motor. It is connected to an ideal torque sensor, hall sensor, and inverter. Hall sensor is sensing the position and velocity. Torque sensor connected with angular velocity source and revolute joint. The Inverter block represents a six-pulse three-phase controlled converter consisting of three bridge arms each with two switching devices like a metal-oxide-semiconductor field-effect transistor (MOSFET) [42]. The commutator part reverses the flow of current to make sure that the magnetic field generated on the rotor is always opposed by the field on the stator [43]. Moving objects like propellers, motors, and other solid parts are connected to the revolute joint.

3.1.5.4. 6-DoF sensor model

To indicate IMU attitude some Simscape blocks should be used such as a 6-DoF joint, transform sensor, world frame, and gravity. The 6-DoF joint measures rotational and translational degrees of freedom between two frames. It allows unconstrained rotation and 3-D translation. The first part called the follower translates relative to the base part. Then the follower frame can rotate freely, with its origin as the pivot. A transform sensor calculates a time-dependent relationship among that two frames. It passively senses derivatives and time-varying transformations in 3D among the two frames. An example of these as a Simscape block is shown in Figure 19.

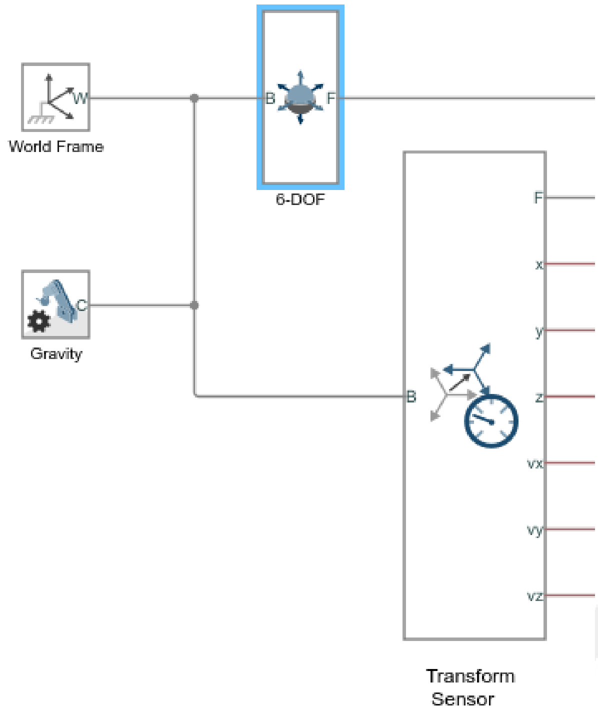

Figure 19. Orientation and 6-DoF modeling blocks in Simscape.

Additionally, gravity and world reference frame should be added to simulate the environment truly. The gravity is uniform. The world frame gives access to the ground frame. It is orthogonal, a unique motionless, and right-handed coordinate frame predefined in any mechanical system model.

3.1.5.5. Controller and overall quadcopter model

In that quadcopter proportional-integral-derivative (PID) controller was used. There are a variety of control technics in the industry, but PID is the most common one. The nature of the being tunable and applicability for the not identified plant is the strongest part of it [44]. The general mathematical implementation of PID is shown in the below equation.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{3.18}$$

The error value indicated as e(t) is the difference between the actual value and the reference set-point. The main goal of the controller is to minimize the error. Kp, Ki, and Kd are the proportional, integral, and derivative gains respectively. The control signal has different effects on the response of the system due to the PID gains [45], [46]. A simple block diagram of the PID controller is given in Figure 20.

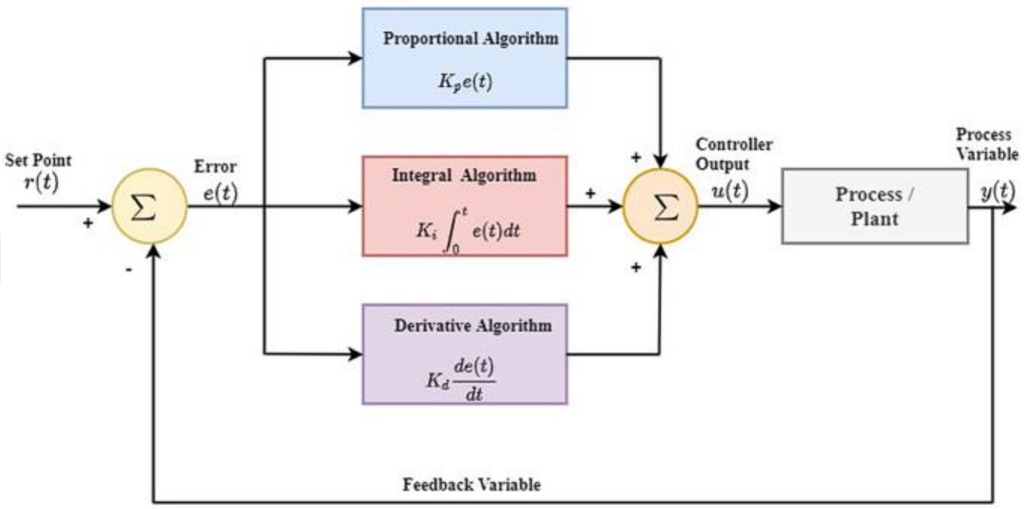

Figure 20. PID controller block diagram [47]

The proportional gain Kp creates a rapid change in the error value. It's increasing the overshoot and may lead to the system being unstable. The integral term Ki is related to the previous value of the error. It is a sum of the errors and increasing the settling time. The derivative term Kd is related to the error change rate over time. It is contributing to the settling of the system. For the noise environment filter solution may need for the derivative part. For roll, pitch, and yaw control in other words to simulate attitude control discrete PID controller was used as shown in Figure 21.

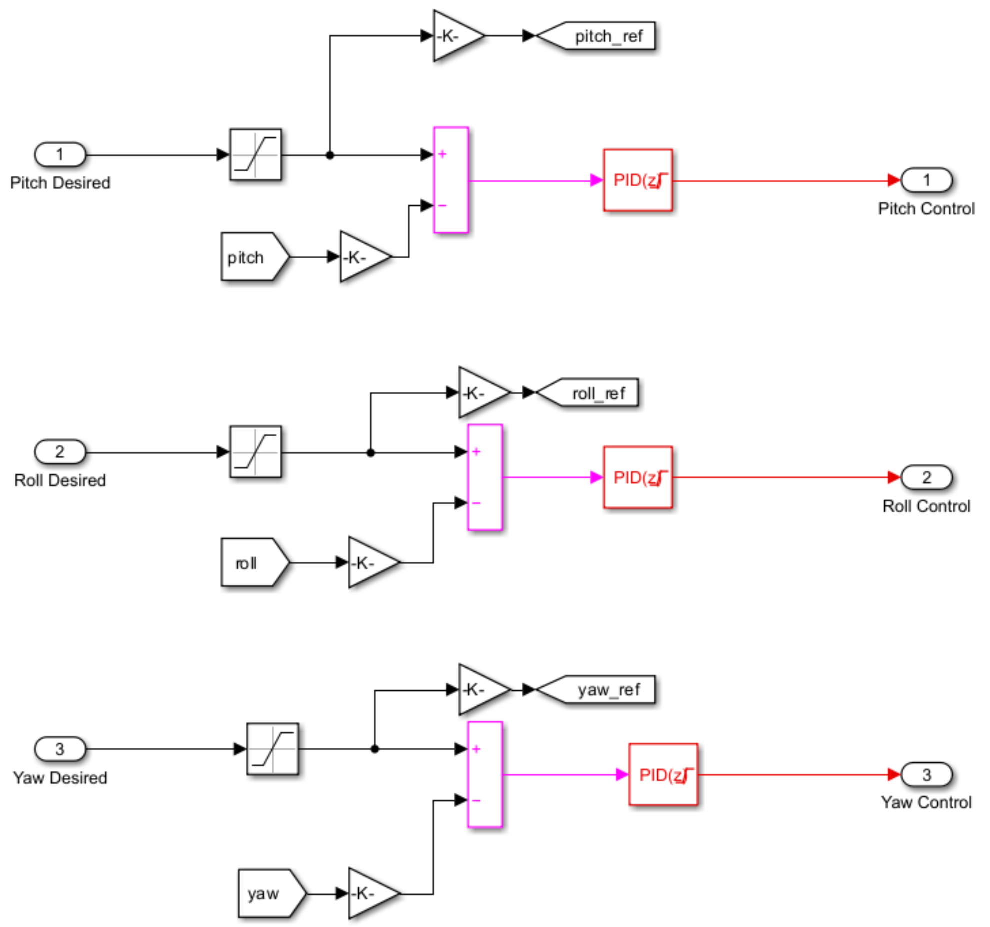

Figure 21. Discrete PID controller for attitude control

The overall control schematic and plant model of the quadcopter for this work is given in Figure 22. The position data in X, Y, and Z directions are coming from the transformed sensor. Also, relative linear velocities are gathered from the transformed sensor. That information sends to the controller part. The controller block calculates the desired action and control signal to control the BLDC motors. The motor inputs are collected from the quadcopter plant block and converted to the PWM signal to drive motors. After controlling the quadcopter, attitudes are changing and the 6-DoF attributes are recalculated again.

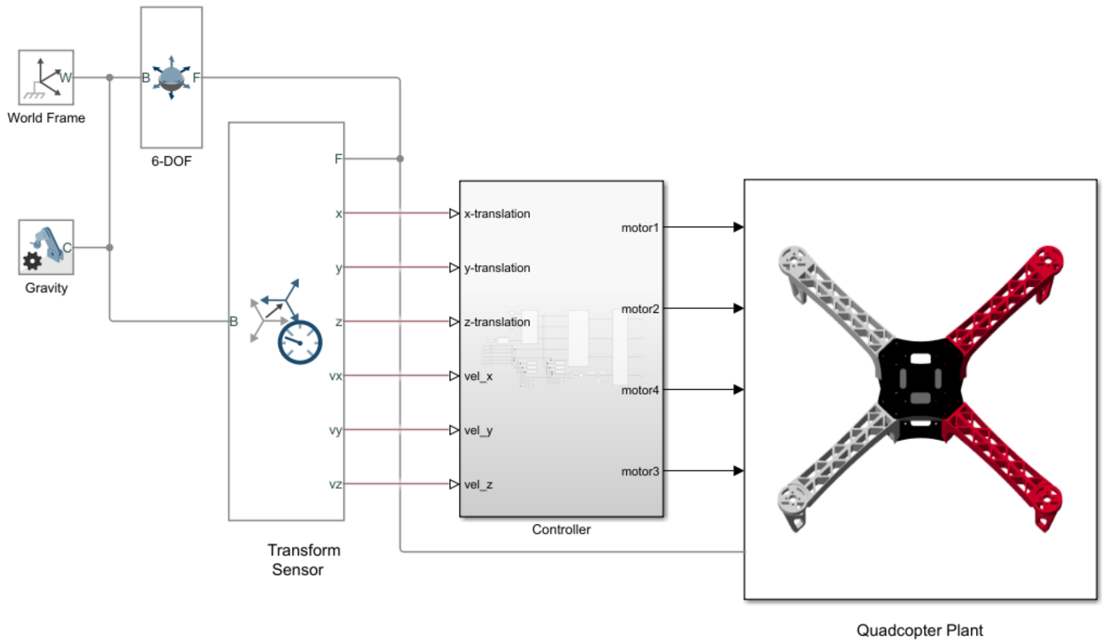

Figure 22. Quadcopter overall Simscape model.

3.2. Design a Virtual Environment with the Unreal Engine

The cost of developing and testing algorithms is quite high. For this reason, it is becoming more and more common to use realistic simulation environments. Today's game engines are very advanced in real-environment modeling. Today's game engines are very advanced in real-environment modeling. What makes Unreal engine powerful is that it is based on C++ software language and the render engine is very powerful. It is possible to make realistic and physics-appropriate simulations in unreal engines [10]. It is necessary to have a very good rendering engine to get realistic images. In this regard, the Unreal engine provides superior features compared to its competitors [48], [49]. Some usage areas are given below.

- Games.
- Film & Television
- Architecture
- Automotive & Aviation
- Simulation
- Digital Twin

It also gives the ability to high-fidelity modeling. Every aspect of the design can be fully controlled with the open source and robust C++ application programming interface (API). The Unreal engine gives huge support for the integration of other tools such as Maya and Blender. These tools were developed for 3D and material design. Characters, assets, and material can be drawn here first, then exported studies can be imported into the unreal engine environment quickly. As an example, drawings are shown in Figure 23.

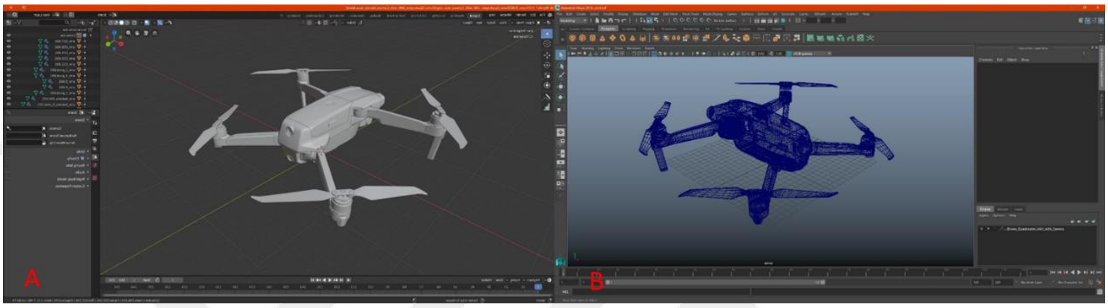

Figure 23. A quadcopter design A) Blender and B) Maya

3.2.1. Landscape Model

In a 3D space environment, modeling a ground level is very important. Different kinds of land can be designed for simulation. Forrest, rocks, hills, mountains, rivers, and so on are some of those. The Unreal engine landscape system has a collection of huge outdoor environments. The landscape system is working in layer logic. Each layer can be modeled on several types and contain unique properties. Layers are textures that can store both color data and height maps. The landscape system function can be found under the modes menu as landscape. It has three modes manage, sculpt, and paint as shown in Figure 24.

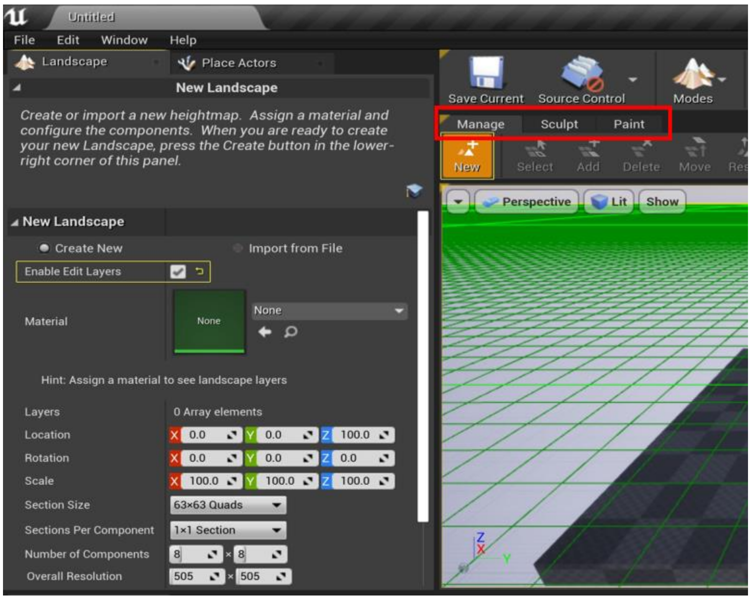

Figure 24. Landscape modes [50]

In manage mode, the terrain can be created, sized, modified, deleted, and moved. Also, it can be imported from external tools or software. Editing layers, hiding, or unhiding can be done in this section. Lock and unlock properties are much better when modifying and working multiple layers. In sculpt mode, it paints height values with a variety of brush types. Scaling, increasing, and decreasing height map. The region of interest can be defined as to work locale area. In Figure 25, the tool's capabilities are shown. Paint mode gives the modifying ability to apply material layers selectively.

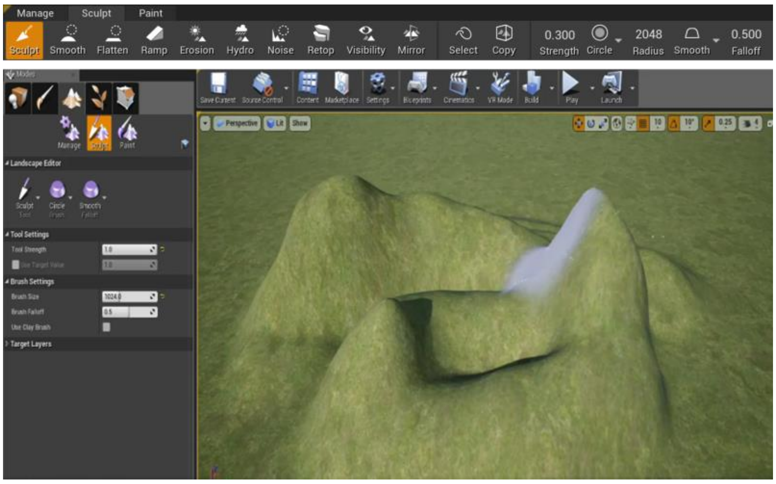

Figure 25. Sculpt Mode [50]

3.2.2. Foliage Model

The foliage enables quick and smart painting. In a condition that static mesh already has, then many of them can be added to the maps without any effort. An example of the usage is given in Figure 26.

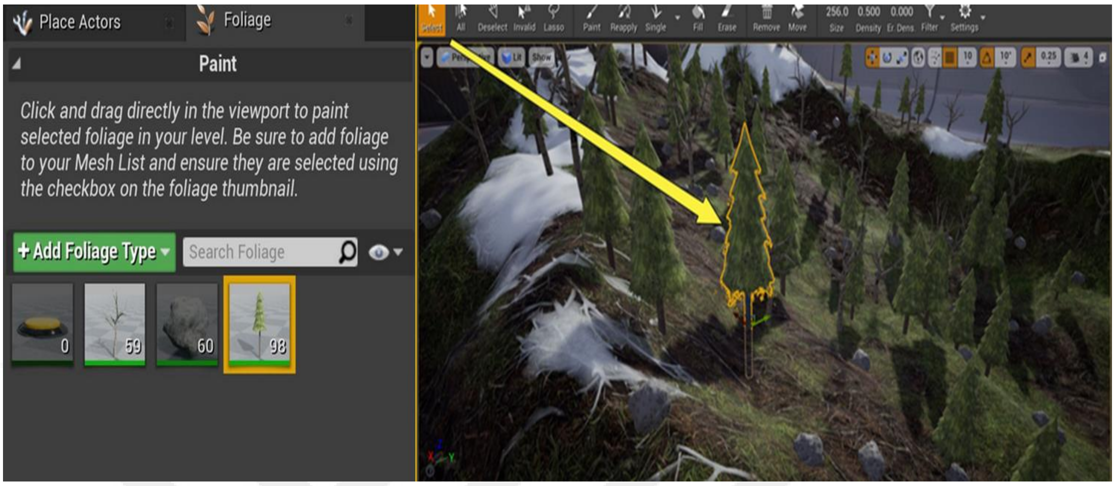

Figure 26. Foliage Tool [50]

Trees, rocks, flowers, grass, and any kind of meshed object can be added simultaneously anywhere on the maps. Also, their properties can be changed at once. The tool has some extra filters to separate different kinds of objects.

3.2.3. Light Model

There are different kinds of lights inside the unreal engine. Spotlight, point light, and directional light are given in Figure 27. Their orientations, intensity levels, and colors can be adjustable. Lights also have movable properties. Static, stationary, and movable. Static lights don't have overhead on the game. Stationary lights can change their brightness and colors, but can't move or change their orientations inside the game. Movable lights are free to change and use the full capability of it during the game.

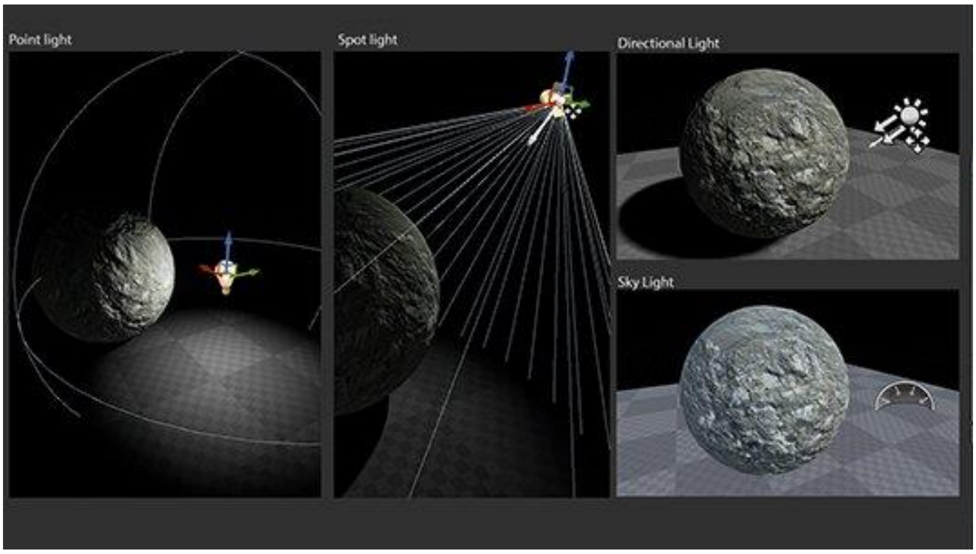

Figure 27. Light types [51]

3.2.4. Fog Model

The main aim is to use fog to change the ambiance of the environment. There are three types of fog inside the unreal engine. Atmospheric fog, exponential fog, and volumetric fog. Height-based, distant fog model can be done with an exponential fog model. For a more realistic look to design the atmosphere in the scene, atmospheric fog should be selected. Depending on the sun's altitude, the sky color is changing. Volumetric fog can be used both locally and globally. It's changing the density and light. The example for these three fog models is given in Figure 28.

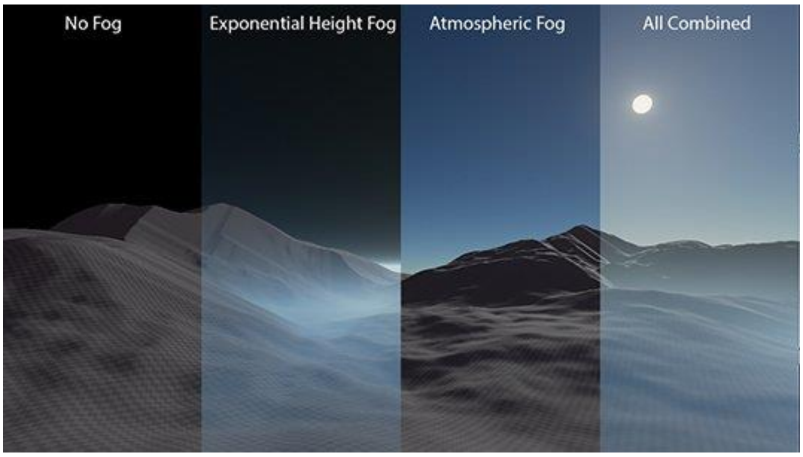

Figure 28. Fog types [51]

3.2.5. Custom Environment Model

The aim of the design new working environment is to simulate how to affect image processing and object detection processes in different weather and space conditions. With the help of the Unreal engine modeling tools and some readymade assets, these different scenes are modeled very fast. In the Unreal engine, there is also a plugin segment to install different software packages to make easy to design process.

In the Unreal ecosystem, marketplace and asset management are have their own value. This is because some of the design environments needs huge elements and items. To be able to combine all of them such as land, rocks, trees, sky, lights, and so on. Every single assets has their own development process. For example a tree can be static or movable. It can be more detail when the character is close to it and low detail when going far. One of the most popular asset management tools integrated with the Unreal engine is Quixel. You can find rock, colors, textures, maps, and many inside it. Another great tool is Niagara visual effects (VFX) system. It is the primary visual effect tool for the Unreal engine. It can be done with it such as snow, rain, and other particle effects.

3.3. Image Processing with Yolo

Today, studies on image processing are increasing considerably. With the contribution of artificial intelligence (AI) to these studies, algorithms with very high success rates are being developed. Image processing can be used in drones, land vehicles and underwater applications. Facial recognition, video analysis, and autonomous vehicles are common areas of image processing. Object detection, object classification, and object tracking can be done by image processing. These techniques help to identify real-time environments and objects with digital images as inputs. Image classification sends the image to a classifier to deal with it. It takes into consideration the whole image not where that image occurred. On the other hand, object detection is more advanced than classification. It makes a boundary box around the classified object. Object tracking is related to deep learning. It tracks the object’s movements. In other words, it can predict or estimate the object’s relevant properties or where the object is. A simple example of classification and detection is given in Figure 29.

Figure 29. Classification and object detection [52]

The object detection work can be turned into semantic segmentation or instance segmentation. After detecting the object, it can be marked all pixels related to that object in the image or segment each of the objects and detect a region of interest in the image. The difference between them is given in Figure 30.

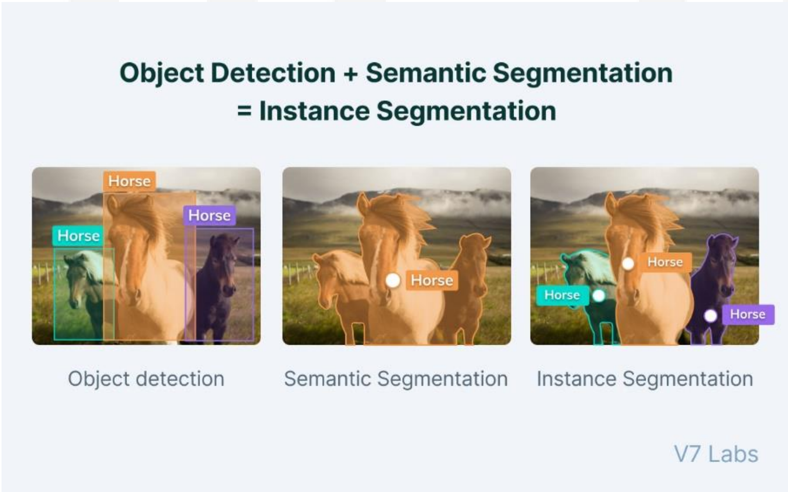

Figure 30. Detection and segmentation [52]

It can be used variety of different approaches to perform detection of objects. There are two main categories to conduct those approaches. These are two stage detectors and one stage detectors. Generally the common detectors in the field uses deep learning based techniques. You Only Look Once (YOLO) and Single Shot Multibox Detector (SSD) are in the one stage part. Regional based techniques are in the two stage part such as Region with Convolutional Neural Networks (R-CNN), Fast R-CNN and so on. These are given in Figure 31.

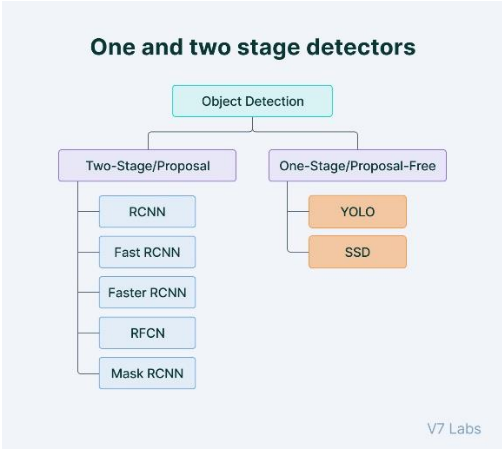

Figure 31. Detector stage and types [52]

In the R-CNN-based approaches, firstly it is identifying the region or sub-region of the object whether is contained or not. Then next, classify the object in that region. This technique is more accurate; however, it is slower than the one stage. R-CNN based model is given in Figure 32.

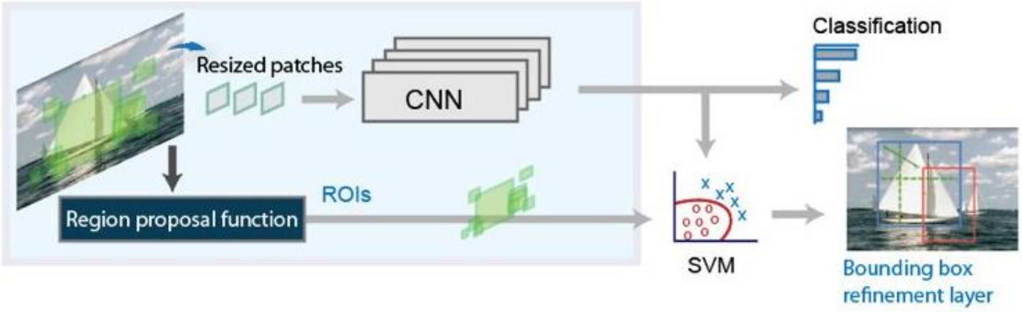

Figure 32. R-CNN-based detector model [53]

Firstly, the R-CNN detector creates a region proposal taking an algorithm like Edge Boxes [54], [55]. Resized images are sent to the Convolutional Neural Network (CNN) part. Images are trained in that section and sent to the Support Vector Machine (SVM) part to bound the images refined by SVM.

In the single-stage part, YOLO is one of the most popular applications in object detection. Prediction is done with the network produced by CNN. The full image section is using anchor box. The one-stage network is much faster than the two-stage, but it can't reach the same model accuracy when compared to the two-stage. The simple model of the YOLO-based network is given in Figure 33.

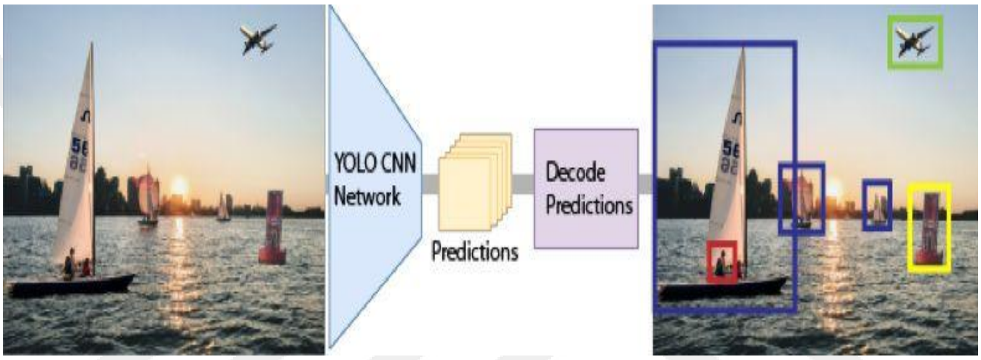

Figure 33. YOLO-based network model [53]

The model splits the input image into a grid of cells, where these cells are responsible for predicting a bounding box. The bounding box containing the x, and y coordinates is estimated by the grid cell. Height and confidence knowledge are gathered. Also, the estimation of the class relies on these cells. R-CNN or YOLO can use a pre-trained network or a custom-trained network. To be able to use a custom train type, a huge dataset can be obtained. Layers and weights should be adjusted manually. Thus, the results should be remarkable [56]. For the trained type, the process starts with the pre assumption then it should be fine-tuned. This way is much quicker than the custom one. Some of the popular pre-trained networks are sown in Figure 34.

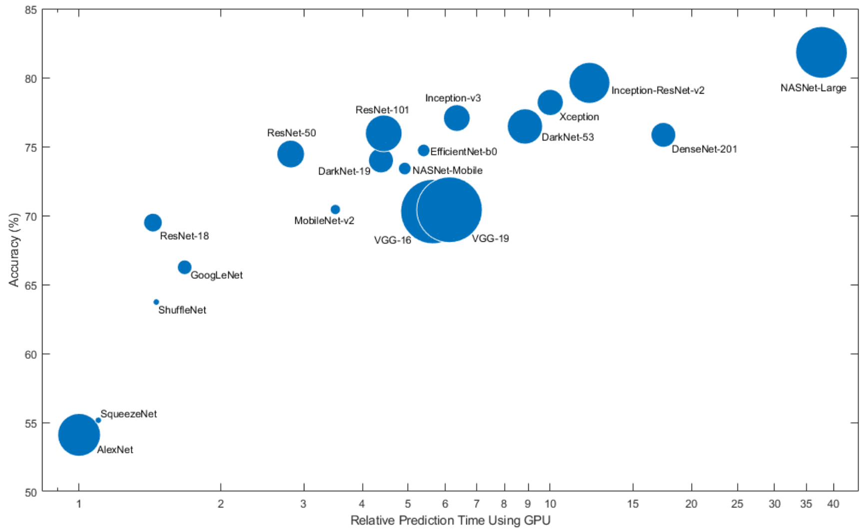

Figure 34. Pretrained neural networks comparison [57]

AlexNet, ResNet-50, GoogleNet, and DarkNet-53 are the common pre-trained networks. There is a threshold between the graphical processing unit (GPU).

3.3.1. Image Labeling and Anchor Boxes

After carefully selecting the dataset, the region of interest (ROI) and anchor boxing process should be performed. In Matlab software there is a tool called image labeler is used to prepare the ROI of the pig images. A simple example of that procedure is given in Figure 35.

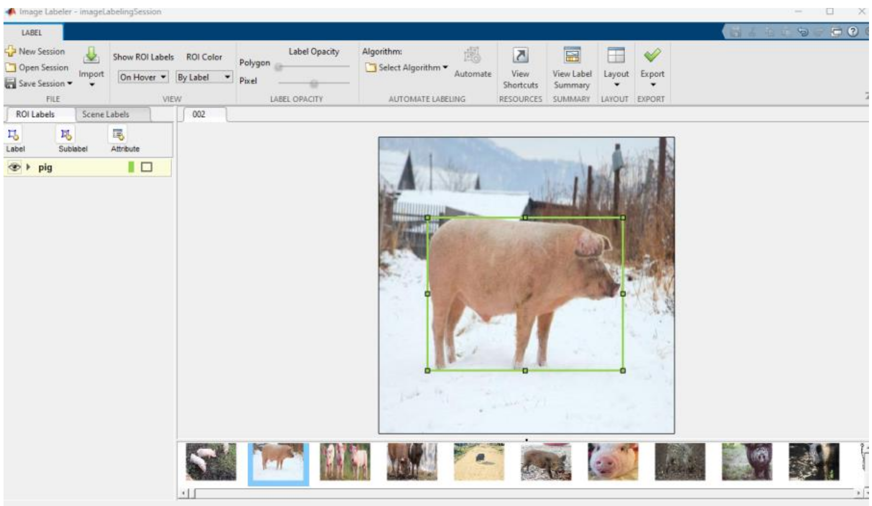

Figure 35. Image labeler in Matlab

When all the images are inserted into the image labeler, the image directly appears in the tools. It is possible to delete or add new images, pass some of the images, or do other stuff. With the label button, it can be given the name of the label part. With the help of the mouse, ROI areas can be quickly drawn and saved. Also, it is available to develop an algorithm to do all the labeling staff at once. After labeling the images you can save the session and export the data to the Matlab workspace with one click. That session file can be used later to continue with the latest saved point. In the exported data you can find a ground truth table. It is possible to label multiple instances and multiple ROI.

Another crucial part of object detection is to define anchor boxes. Object detection is conducted under a deep neural network. In other words, it needs more computation. To overcome that problem we need to predefine the location of the images. To increase efficiency and speed, anchor boxes should be used. An anchor box can be defined briefly as a box that has a certain height and width. A simple representation of it is given in Figure 36.

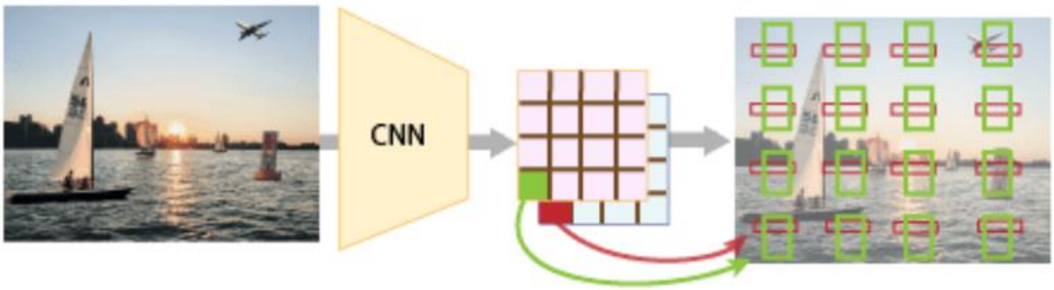

Figure 36. Anchor box [58]

By mapping the location of the network output back to the input image, an anchor box can be defined. The process is repeated for every output. The result creates a set of tiled anchor boxes across the whole image. Each anchor box simulates a specific prediction of a class. The anchor box can be calculated easily by using image label data and the Matlab anchor box estimation algorithm.

3.3.2. Train Dataset with YOLOv4

YOLO is the one stage neural network model for the regression and classification. It uses a few anchor boxes. DarkNet neural network is powering the YOLO at the background. YOLOv4 actually upgrade version of the YOLOv3 model. YOLOv3 is a deeper feature detector and taking roughly 30ms per interface. On the other hand YOLOv4 brings new approach that separating detection task into two part. Basic structure of the YOLOv4 is given in Figure 37.

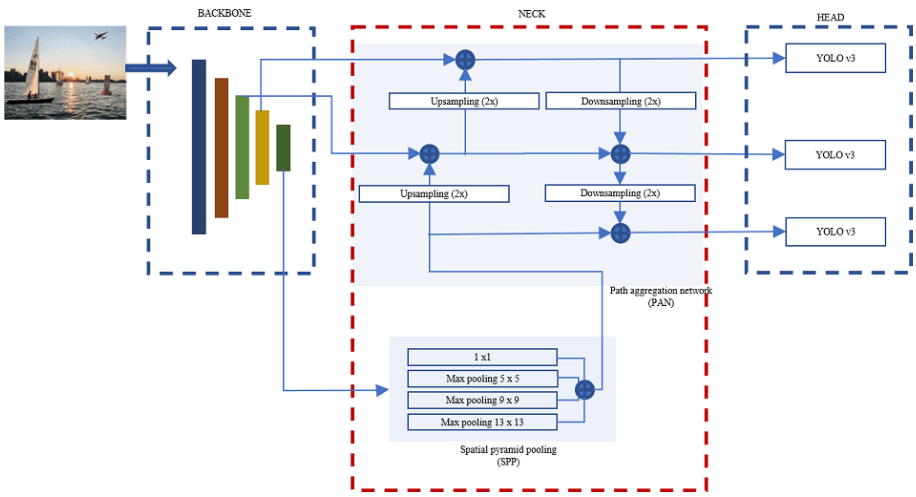

Figure 37. YOLOv4 architecture [59]

The architecture of YOLOv4 consists of three main parts. Backbone in the first, neck in the middle, and head at the end. In the first part as a backbone, input images are used to extract the feature maps. Inside the CNN side, VGG16 or CSPDarkNet53 can be used. In the dataset part, COCO or ImageNet can be used. The backbone consists of five residual modules. The output of this section is feature maps. These maps fused with the neck section of the YOLOv4. In the middle part, there is a neck section connecting the backbone and head side. It has two parts inside the path aggregation network (PAN) and spatial pyramid pooling (SPP) modules. It takes feature maps from different layers and concatenates them. Then send it to the head. In the SPP, the max-pooling process is performed. In its kernel, 1x1, 5x5, 9x9, and 13x13 max-pooling operations can be done. In the PAN section, it takes high-resolution maps to apply up-sampling and down-sampling work. Lastly head section act as a prediction section by processing the packed features. In that part, YOLOv3 is used. In this training section simple road map is used which is given in Figure 38.

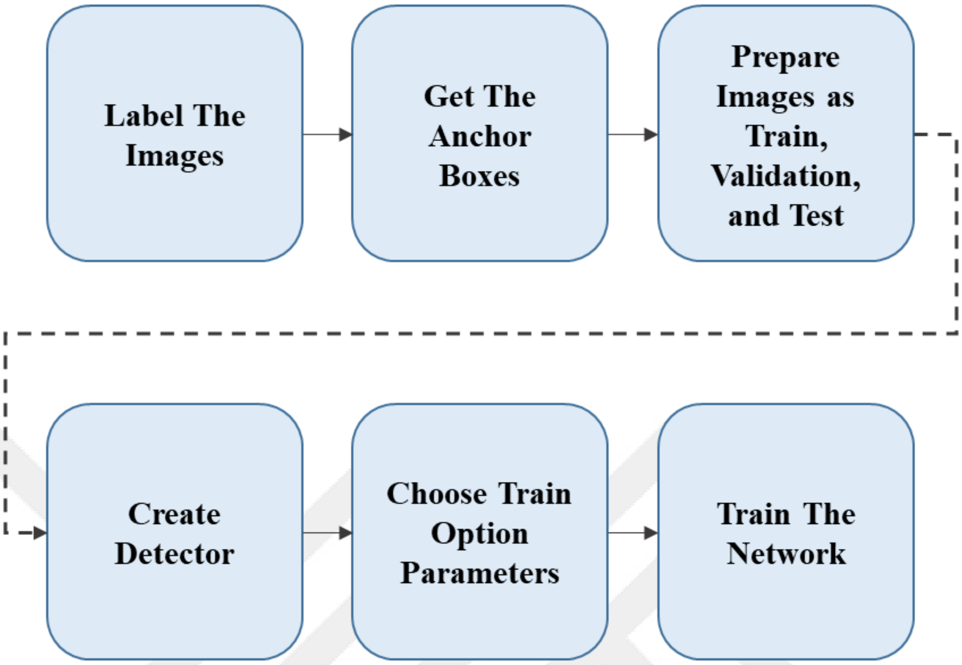

Figure 38. Training road map for YOLOv4 network

Image labeling and anchor boxing are shared in the above section. To tell about separating image datasets into three parts, it's a common phenomenon that divides data train, validation, and testing. Data is divided randomly like 10% for validation, 60% for training, and the rest of the data for testing [60]. It is important to separate test data and training data to avoid trained data testing failure.

In the detector creation part, using the "yolov4ObjectDetector" function inside the Matlab detector can be easily created. It can be defined pre-trained neural network here. In this work, CSP-DarkNet53 neural network architecture is selected. This network is applied to the COCO dataset and is already trained. The name inside the Matlab is csp-darknet53-coco. It is a CNN-based network and consists of 53 layers deep. It is often used in YOLO applications. The Cross-stage-partial (CSP) structure in YOLOv4 is given in Figure 39.

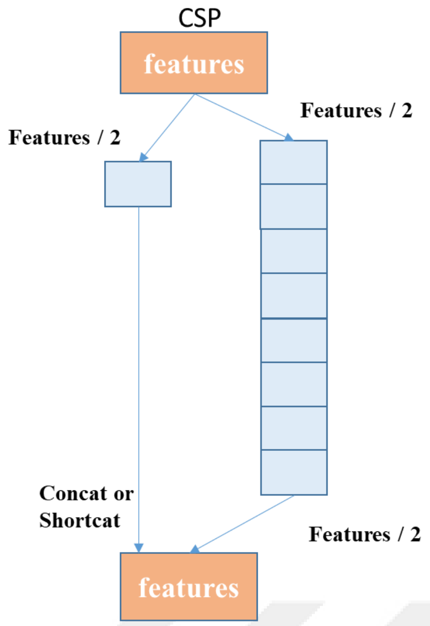

Figure 39. CSP architecture in YOLOv4

Features are divided into two. Half of the features follow the main patch to create more semantic information. The other half goes bypass to preserve extensive information. The data augmentation part is used to improve accuracy. Using the transform function inside Matlab following operation is done with the help of the “augmentData” helper function. Color jitter augmentation, random horizontal flip, and random scaling are done.

In the training option part, the configuration of the train solve is set up. An adaptive moment estimation (Adam) solver is chosen for the training process with a hundred epochs. It keeps an element-wise moving average of both the parameter gradients and their squared values.

$$m_\iota = \beta_1 m_{\iota-1} + (1-\beta_1)\nabla E(\theta_\iota)$$
$$v_\iota = \beta_2 v_{\iota-1} + (1-\beta_2)[\nabla E(\theta_\iota)]^2 \tag{3.19}$$

β1 and β2 are decay rates,. They are using the “SquaredGradientDecayFactor” and “GradientDecayFactor” parameters in the option part. Adam solve updates the CNN parameters by using the moving averages as below.

$$\theta_{\iota+1} = \theta_{\iota} - \frac{\alpha m_l}{\sqrt{v_l} + \epsilon}$$

(3.20)

If gradient noise is low the moving average is becoming low. So the updates becomes low too [61]. Learning rate α parameter can be used for optimization algorithm. It is called “InitialLearnRate” in the option parameter part.

To be validate data more, “ValidationFrequency” parameter can be reduce. To select the hardware source, “ExecutionEnvironment” parameter should set. Default parameter is auto which is perform GPU operation.,

4. EXPERIMENTAL PROCEDURE AND RESULTS

In this study all the workload are arraneged in an order to cleary determined the roadmap. Lenovo ThinkStation P340 model desktop computer was used for all analyzes in this study. The computer components are Intel Xenon W-1270 3.40 GHz central processing unit (CPU) with 8 cores, 32 GB ram, solid state disk and Nvidia Quadro P2200 5 GB GPU.The UAV type and parameters are selected first to be able to reperesent a flying object. Then phsyical model of it is developed based on the dynamic equations with the Matlab Simscape tool. The requirement of this study is get the virutal video images and detect an object. For this purpose virtual environmet for UAV is created in the Unreal engine. The connection between virtual envirınment and the Matlab Simscape tool is established. The UAV flights and the video images are collected virtually and the images are saved. To be able to detect objects in these images, trained neural network should be created. For this part of works, dataset for the detection object shoul be generated and collected. By the external dataset provider, images are collected and prepaired such as quality selection and get rid of unconvinent content. For object detection pig animal and person are selected and dataset is collected regarding of these topics. Pre-stage of training such as image labeling and anchor boxing is done into the dataset. Dataset is splited into three part, validation, traning, and test to conduct train process succesfully and check the training quality. After that image processing framework and pretrained CNN network are selected. Selecting the right option parameters, the network is run and trained in GPU supported computer machine. The last aciton is test the virtual video images on the trained network and compare the results. All these steps are given in Figure 40.

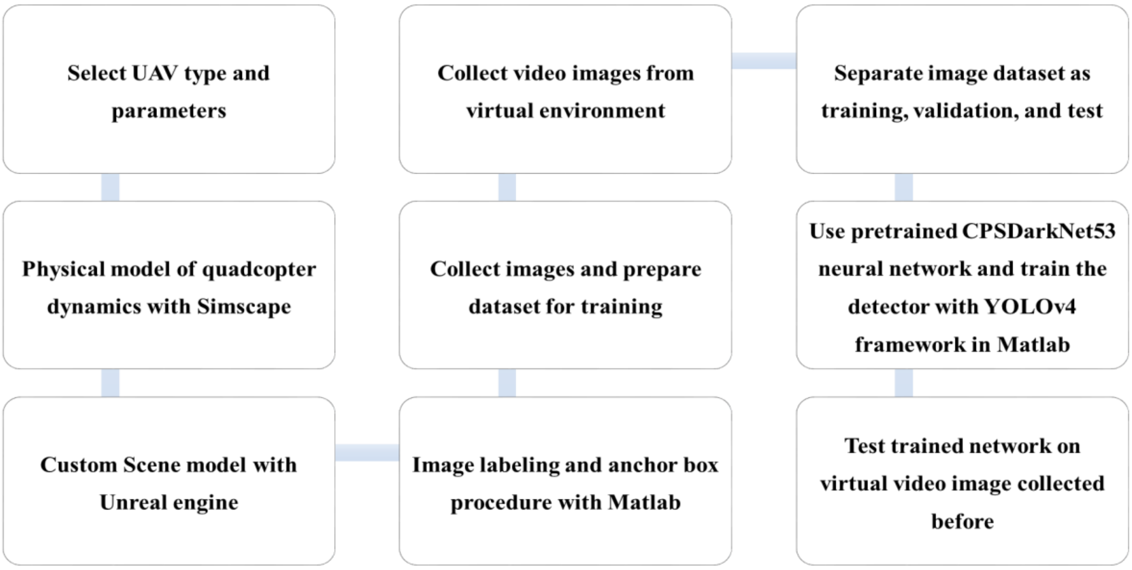

Figure 40. Flow chart of the experiment of the study

Since the primary purpose of this study is to lead the projects to be made in the defense industry, the objects to be determined are determined as pig and human in this concept. Here, the wild boar was chosen because it is important to detect both the damage it can cause to the equipment in the field or the military elements (wounded soldiers) in a rural area.

In addition, improvements to be made in simulation systems provide an innovative perspective in many areas. In studies conducted in this field, simulations are often preferred because they are more advantageous than many. It is thought that the study brings a different approach in this regard.

Quadcopter model used in the study is selected the DJI F450 quadcopter frame. In Table 4 some parametric information about that frame such as mass, inertia, and length is given. This type of quadcopter are common to use in the research area due to the stability, ability to conduct real test with the kits which is already in the market.

Table 4. DJI F450 quadcopter frame parametric information

Parameters	Values	Symbols
Arm Length	225 mm	L
Thrust coefficient	0.0087	K_{th}
Drag coefficient	$0.0055x10^{-2}$	K_d
Total mass	1.776 kg	m
Moment of inertia	0.0035; 0.0035; 0.0055 kg.m^2	I_x, I_y, I_z
Rotor inertia	2.8x10-6 kg.m^2	J_r

Getting reference to that frame, physical model is built in the Matlab Simscape environment. All the component such as BLDC, propeller, frame, orientation sensors and controller part are done. Addition to the model, it is needed to connect the model with the virtual environment. For this purpose Matlab UAV toolbox is used. An example of the setup is given in Figure 41.

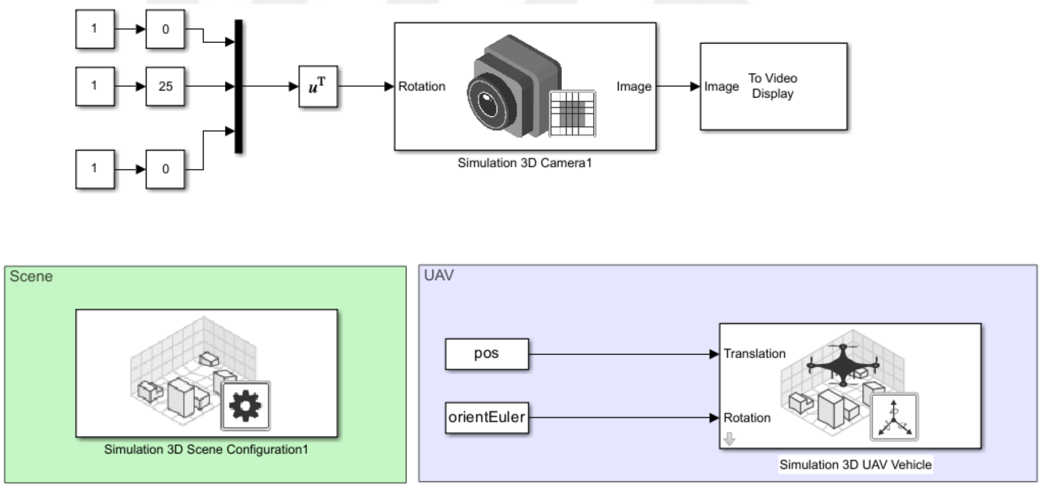

Figure 41. Connection setup between virtual environment and Matlab software

The Simulation 3D scene configuration block make connection with the Unreal engine. It starts the model and Unreal engine software. Then, Simulink run button should be pressed the Matlab side. On the other hand, when the Unreal engine play button pressed, simulation connection is done and the UAV is started to fly. All the fly conditions like translation and positional parameters are sending to the Unreal engine via the Simulation 3D UAV vehicle block. On the UAV there is a camera model to responsible for taking video images. These image inside the Unreal virtual environment are capturing with the

Simulation 3D camera block. The camera orientation, resolution, and position can be changed with that block. In the development environment camera mounted to the UAV with the fixed orientation parameters. There is no gimbal effects are considered in the simulation. The camera properties related this work is given in Table 5.

Table 5. The camera properties

Parameters	Values
Orientation (degree)	x:0, y:0, z:25
Focal length (pixel)	1109, 1109
Optical center (pixel)	640, 360
Image size (pixel)	720, 1280
Relative translation (X, Y, Z) (m)	0.1, 0, 0
Relative orientation (Roll, Pitch, Yaw) (degree)	0,0,0

The next step for the experiment in this study is to create custom virtual environment. In this study, two different applications are conducted. In the firt part, three custom scenes and its variants are designed and integrated with Matlab Simulink software. The UAV flies in 15 meter above the ground and all the images captured by the virtual camera at that altitude. In the Scene 1, there are three vairants. All variants have road, winter landscape, threes with snow textures, and pig animals which are placed tor the map. In the first case, model is built under normal winter condition with sunny, low snow, and clear sky view. In the second case, winter condition is changed to the heavy snow, and again sky view clear and sunny. In the last one, winter condition is low and the sky is clear like case one, but, heay fog condition is added to map. All these cases are shown in Figure 42.

Figure 42. Scene 1 design with three variants

In the Scene 2, again three variants are created. The difference in this environment, stony land environment is imagined. All variant model’s background has rocks, trees, stones, and ground landscape like the Arizona. The pig animal characters are again inserted to the area. In the first case the weather is normal, sunny, and the sky view is clear. In the second case, heavy rain and fog is modelled. In the last case, there is no rain. Only fog is added to the environment. Three cases are shown in Figure 43.

Figure 43. Scene 2 design with three variants

In the Scene 3, city model is taken as a reference point. Buildings, trees, and other things are created with the Unreal engine. For the weather condition, skylights are very low and cloudy like the sunset. The pig animals are inserted to the map. Only one scenario is created. The reason is to see how city model effect on the object detection. The Scene 3 environment is shown in Figure 44.

Figure 44. Scene 3 design

For the three different scenes and their variants, the weather conditions, creating snow, rain, and fog effects are dependent some parameter. The Unreal enigne VFX tool is giving ability to create these effects. Some of the parameters are range, velocity, altitude, density and so on. All these parameters can be adjustable. The fog parameters table in Scene 1 and Scene 2 indide third variants is shown in Table 6.

Table 6. Fog parameters are in the Scene 1 and Scene 2

Parameters	Scene 1 Variant 3	Scene 2 Variant 2	Scene 2 Variant 3
Fog Density	0.05	0.1	0.1
Fog Height Falloff	2	0.001	0.001
Fog max opacity	1	1	1
Start Distance	5000	5000	5000
Fog cutoff distance	0	20Million	20Million

Also rain parameters which are belongs to the Scene 2 in the second variant are shown in Table 7.

Table 7. Rain parameters are in the Scene 1 and Scene 2

Parameters	Scene 2 Variant 2
Lifetime mode	random
Lifetime min	1.12
Lifetime max	1.75
Mass min	0.75
Mass max	1.25
Sprite size mode	non-uniform
Sprite size	x:3, y:50
Velocity	x:10000, y:10000, z:5000
Spawn rate	x:0, y:0, z:-1000

After that, the snow parameter which are belongs to the Scene 1 in the second variant are shown in Table 8.

Table 8. Snow parameters are in the Scene 1

Parameters	Scene 1
	Variant 2
Lifetime mode	random
Lifetime min	10
Lifetime max	15
Mass min	1
Mass max	5
Sprite size mode	random uniform
Sprite size	x:9.6, y:6.7
Velocity	x:0, y:0, z:-1000
Spawn rate	50000

In the second part of the environment design, a virtual environment containing the forest, hikes and so on is designed. Inside the map some military person figures are added and flight environment is adapted for the UAV. In that part four different cases are designed with different fog level for each cases. Case 1 has no fog effect and the case 4 has the most fog effect. Only view distance fog parameter is changed according to the cases inside the Unreal engine. Case 1, case 2, case 3, and case 4 have 0, 1000, 3000, and 10000 meter view distance fog parameter respectively. The UAV altitude is configured at 30 meter above the ground. The virtual camera which is on the UAV is fixed with the orientation parameter given in the Table 5. All the cases belong to the second part are given in Figure 45 and Figure 46.

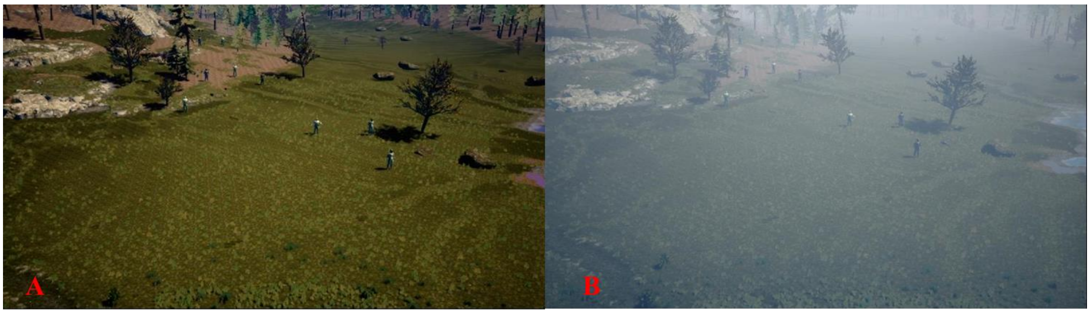

Figure 45. Virtual environment with military person. A) Case 1 and B) Case 2

Figure 46. Virtual environment with military person. A) Case 3 and B) Case 4

Obtaining datasets in neural network applications is a critical part. Image datasets can be taken from some sources such as ImageNet, MS COCO, Stanford Dogs Dataset, and so on, or can be created on your own dataset for computer vision applications. In this work, for the military person detection application we used only pretrained network for detection. MS COCO dataset has contained person categories. Because of the MS COCO dataset don't have pig class inside, different kinds of pig images are collected for the image dataset from a ready dataset source [62]. These pictures are resized 224 x 224 pixels for computation issues. The sample of the dataset is given in Figure 47.

Figure 47. Pig image samples from the dataset

The dataset contains 1093 pig images. With training this dataset, pigs can be detected inside the virtual environment in which the video is taken from a UAV's camera. For clarity, some of the images that have low quality aren't added to the training dataset. After collecting and preparing the dataset, it can be passed to the next step. It is the neural network training step. For this purpose, YOLOv4 real time object detection framework is chosen. Previous version of YOLO is not capable of the real time object detection, but with the improvement YOLOv4 is published. It is the first efficient version for real time object detection. Also, there are many new versions and customizations on it. The Matlab has also YOLOv4 framework. Mostly the Matlab software and its tools are used in this study. Thus, going with the supported YOLO version is simplified the total work effort. In the training option parameters should be correctly chose. Adam solver is faster than the others. Option parameters are used in the detector training is given Table 9.

Table 9. Training option parameters.

Parameters	**Values**
Solver Name	Adam
GradientDecayFactor	0.9
SquaredGradientDecayFactor	0.999
InitialLearnRate	0.001
LearnRateSchedule	none
MiniBatchSize	4
L2Regularization	0.0005
MaxEpochs	100
BatchNormalizationStatistics	moving
DispatchInBackground	TRUE
ResetInputNormalization	FALSE
Shuffle	every-epoch
VerboseFrequency	20
ValidationFrequency	1000
CheckpointPath	tempdir
ValidationData	validationData

Training process takes approximately 10 hours. The Scene 1 and its variants detection result as shown in Figure 48. These detection results are measured from the one of the video frames. So, the results given are indicating that specific point detection result.

Figure 48. Training results for the Scene 1 with three variants

In Figure 48, it is evaluated that fog is the main property to affect the confidence. Although there is a heavy snow in variant 2, it is not affecting more the result. However, in variant 3 has fog and it changes the result. There is no detection occurred at the same frame as variant 1 and variant 2. Detection just occurs in the near frame to the pig animal. The Scene 2 and its variants training results are given in Figure 49.

Figure 49. Training results for the Scene 2 with three variants

As shown in the Scene 2 results, variant 1 has the higher confidence value result compare to the variant 1. By the way, there is no detection occurred at the same frame in variant

3 which is simulate the fog case. The detection results of the Scene 3 are given in Figure 50.

Figure 50. Training results for the Scene 3

In the Scene 3, it is aimed that the city environment is effect negatively or positive. Maximum confidence result reached in the training is 96%. One of the most accurate evaluation metrics for the detection algorithms is the F1 score parameter. F1 score is an alternative machine learning evaluation metric that assesses the predictive skill of a model by elaborating on its class-wise performance rather than an overall performance as done by accuracy. F1 score combines two competing metrics precision and recall scores of a model, leading to its widespread use in recent literature. F1 score can be calculated with harmonic mean of the precision and recall values. The values of the F1 are changing between 0% and 100%. Higher the percentage of the F1 score means the better quality of the classification. For the both cases in this study which are pig detection and person detection applications, F1 score is considered to measure the detection quality. F1 scores for all these Scenes and variants related to pig detection are given in Table 10.

Table 10. The Scene 1, the Scene 2, and the Scene 3 F1 scores

Cases	Scene 1 F1-score	Scene 2 F1-score	Scene 3 F1-score
Variant 1	0.64	0.66	0.47
Variant 2	0.55	0.63	No variant
Variant 3	0.48	0.52	No variant

As seen in the F1-scores table, the Scene 3 has the lowest F1 score. In the Scene 1 and Scene 2 fog effect is added in third variants has the lowest F1 score. Rainy and snowy conditions are less effective than the fog effect. Lastly for the military person detection part single frame detection results are given in Figure 51 and Figure 52.

Figure 51. Person Detection A) Case 1 and B) Case 2

Figure 52. Person Detection A) Case 3 and B) Case 4

When the weather conditions are changed the detection results are getting worse. In that scenario only the fog parameters are changed. Case 1, case 2, case 3, and case 4 have 0, 1000, 3000, and 10000 meter view distance respectively. Increasing the view distance parameters means make the more fog effect to the environment. To look at overall view of the detection performance F1 score table for the person detection application which is performed with different fog levels is given in Table 11.

Table 11. F1 scores for the person detection application

Parameter	0 meter fog level	1000 meter fog level	3000 meter fog level	10000 meter fog level
F1-Score	0.77	0.67	0.64	0.60

As seen in the F1 score table, for the person detection application, increasing the view distance meter for implementing fog level is decreasing the F1 score and the detection precision. In 0 meter case F1 score is reached 0.77. Also, for the 0 meter case max detection accuracy in that specific frame is 91.7%.

5. CONCLUSION

In this thesis, object detection was performed on pig images and military person images using virtual environment flight conditions. Images were taken from environments designed using a quadcopter physically modeled digital twin. The configurations of each component and related controllers in the quadcopter were modeled. Pigs and military person in the designed virtual environments were tried to be detected with deep learning architecture. Matlab Simscape / Simulink and Unreal engine software are used and specified. YOLOv4 real-time object detection architecture is implemented to detect pig images and military person. In this way, a high-quality quadcopter model was created and virtual environment modeling was successfully carried out in the Unreal engine part. The combined use of this basic digital twin structure and environment is intended to enable us to collect virtual digital images and test flight characteristics. Analyzes were made for two different applications. In the first application weather conditions created in three different scenes. In the second application four different fog level was applied to the one Scene. The YOLOv4 convolutional neural network framework is used. Therefore, the previous version on these architectures cannot perform real-time object detection. The dataset is prepared for the training network depend on the first application. The training phase is well executed with the option parameters selected and takes about 10 hours. The pre-trained neural network is used to speed up the process because the pre-trained network has many classes already learned. All virtual images captured in scenes are tested with the trained network. In the pig detection application, maximum 64% F1 score is reached in the best weather condition. And also evaluated both applications separately, fog effect are most effective parameters than snow and rain for pig detection application. In the military person detection application which is second part, increasing the fog level with view distance parameter makes the detection algorithm lowest accuracy. Maximum 77% F1 score is reached for this case.

Some troubleshooting has occurred in this run. For example, the GPU compute resource is out of power for high-resolution training. Thus, 224x224 pixel images are given to the model. The YOLOv4 framework has a rendering resolution of 608x608 pixels. Another issue is that there are new frameworks release for the YOLO architecture. They have better real-time performance than YOLOv4. In any case, the YOLOv4 training procedure must be more complex and fine-tuned at layers within the network to achieve more precise detection. In addition to all these, nearly a thousand images are used in the data set. It is the lowest value for doing CNN training. However, it is not easy to collect the dataset from an external source. In the later stages of the study, the effects on accuracy can be observed by increasing the number of images in the dataset.

In the study, object detection was carried out from the images taken by the UAV from the created virtual digital environment, which is the criteria for the study. This basic model will be used for future studies. Swarm algorithms, simultaneous localization and mapping (SLAM) algorithms, new version of YOLO frameworks or R-CNN based frameworks will be implemented with it. In addition, it is predicted that the accuracy rate will be increased on both real-time recordings and images by using the parametric perspective to create virtual assets. It is also thought that it will contribute to the literature for the studies to be done for the images and recordings taken over the digital twin.

It is foreseen that the application made in the study will be the basis for the projects to be carried out especially in the defense industry. For this reason, pig and human were chosen. As it is known, wild boars are at the forefront of deceptive or damaging factors for some systems in rural areas. In addition to this, it is important to detect the wounded soldiers or danger elements with the drones. It is seen that the accuracy rate in the object detection studies made on images taken from unmanned aerial vehicles in the literature is lower than other non-real-time object detection applications.

In future studies, it is aimed to eliminate the image distortions caused by weather conditions by using image enhancement methods and to improve the performance of the created network. In addition, it is thought that making applications for different scenes and conditions and datasets in order to improve the study will contribute to the studies in this field.

REFERENCES

[1] F. Mohamed, “Efficient data communication in unmanned aerial vehicles,” M.S. Thesis, Department of Networking, United Arab Emirates University, 2015. [Online]. Available: https://scholarworks.uaeu.ac.ae/all_theses/35

[2] M. Hassanalian and A. Abdelkefi, “Classifications, applications, and design challenges of drones: A review,” Progress in Aerospace Sciences, vol. 91, pp. 99–131, May 2017, doi: 10.1016/J.PAEROSCI.2017.04.003.

[3] A. Fuller, Z. Fan, C. Day, and C. Barlow, “Digital Twin: Enabling Technologies, Challenges and Open Research,” IEEE Access, vol. 8, pp. 108952–108971, 2020, doi: 10.1109/ACCESS.2020.2998358.

[4] Y. Yang, W. Meng, and S. Zhu, “A Digital Twin Simulation Platform for Multi-rotor UAV,” 2020 7th International Conference on Information, Cybernetics, and Computational Social Systems, ICCSS 2020, pp. 591–596, Nov. 2020, doi: 10.1109/ICCSS52145.2020.9336872.

[5] L. Li, S. Aslam, A. Wileman, and S. Perinpanayagam, “Digital Twin in Aerospace Industry: A Gentle Introduction,” IEEE Access, vol. 10, pp. 9543–9562, 2022, doi: 10.1109/ACCESS.2021.3136458.

[6] H. Meyer et al., “DEVELOPMENT OF A DIGITAL TWIN FOR AVIATION RESEARCH,” Sep. 2020.

[7] W. Meng, Y. Yang, J. Zang, H. Li, and R. Lu, “DTUAV: a novel cloud–based digital twin system for unmanned aerial vehicles,” The Society for Modeling & Simulation International (SCS), vol. 99, no. 1, pp. 69–87, Aug. 2022, doi: 10.1177/00375497221109575.

[8] O. H. Alvarez, L. B. Gutierrez Zea, C. Bil, M. L. Fravolini, and M. Napolitano, “Digital twin concept for aircraft system failure detection and correction,” AIAA Aviation 2019 Forum, 2019, doi: 10.2514/6.2019-2887.

[9] N. Grigoropoulos and S. Lalis, “Simulation and Digital Twin Support for Managed Drone Applications,” Proceedings of the 2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications, DS-RT 2020, Sep. 2020, doi: 10.1109/DS-RT50469.2020.9213676.

[10] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “AirSim: High-Fidelity Visual and Physical Simulation for Autonomous Vehicles,” Springer Proceedings in Advanced Robotics, vol. 5, pp. 621–635, May 2017, doi: 10.1007/978-3-319-67361-5_40.

[11] M. Y. BİYİK, M. E. ATİK, and Z. DURAN, “Deep learning-based vehicle detection from orthophoto and spatial accuracy analysis,” International Journal of Engineering and Geosciences, vol. 8, no. 2, pp. 138–145, Jul. 2023, doi: 10.26833/IJEG.1080624.

[12] A. Sharma, N. Jain, and M. Kothari, “Lightweight Multi-Drone Detection and 3D-Localization via YOLO,” Feb. 2022, Accessed: Apr. 23, 2023. [Online]. Available: https://arxiv.org/abs/2202.09097v1

[13] A. Jawaharlalnehru et al., “Target Object Detection from Unmanned Aerial Vehicle (UAV) Images Based on Improved YOLO Algorithm,” Electronics (Switzerland), vol. 11, no. 15, Aug. 2022, doi: 10.3390/ELECTRONICS11152343.

[14] C. Chen et al., “YOLO-Based UAV Technology: A Review of the Research and Its

Applications," Drones 2023, Vol. 7, Page 190, vol. 7, no. 3, p. 190, Mar. 2023, doi: 10.3390/DRONES7030190.

[15] S. P. Kao, Y. C. Chang, and F. L. Wang, "Combining the YOLOv4 Deep Learning Model with UAV Imagery Processing Technology in the Extraction and Quantization of Cracks in Bridges," Sensors 2023, Vol. 23, Page 2572, vol. 23, no. 5, p. 2572, Feb. 2023, doi: 10.3390/S23052572.

[16] S. Caputo, G. Castellano, F. Greco, C. Mencar, N. Petti, and G. Vessio, "Human Detection in Drone Images Using YOLO for Search-and-Rescue Operations," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 13196 LNAI, pp. 326–337, 2022, doi: 10.1007/978-3-031-08421-8_22/COVER.

[17] L. Zhu, J. Xiong, F. Xiong, H. Hu, and Z. Jiang, "YOLO-Drone:Airborne real-time detection of dense small objects from high-altitude perspective," Apr. 2023, Accessed: Apr. 23, 2023. [Online]. Available: https://arxiv.org/abs/2304.06925v1

[18] M. Shafto, M. C. Rich, D. E. Glaessgen, C. Kemp, J. Lemoigne, and L. Wang, "DRAFT MoDeling, SiMulATion, inFoRMATion Technology & PRoceSSing RoADMAP Technology Area 11," 2010.

[19] T. H. J. Uhlemann, C. Schock, C. Lehmann, S. Freiberger, and R. Steinhilper, "The Digital Twin: Demonstrating the Potential of Real Time Data Acquisition in Production Systems," Procedia Manuf, vol. 9, pp. 113–120, Jan. 2017, doi: 10.1016/J.PROMFG.2017.04.043.

[20] "The UK approach to unmanned aircraft systems (JDN 2/11) - GOV.UK." https://www.gov.uk/government/publications/jdn-2-11-the-uk-approach-to-unmanned-aircraft-systems (accessed May 07, 2023).

[21] "Quanum Trifecta Mini Foldable Tricopter Frame (KIT)." https://hobbyking.com/en_us/quanum-trifecta-mini-foldable-tricopter-frame-kit.html (accessed May 07, 2023).

[22] "Flame Wheel ARF KIT - Features | DJI." https://www-v1.dji.com/flame-wheel-arf/feature.html#Huge-Assembly-Space (accessed May 07, 2023).

[23] "Connect ESCs and Motors — Copter documentation." https://ardupilot.org/copter/docs/connect-escs-and-motors.html (accessed May 07, 2023).

[24] A. Yurtman, B. Barshan, and B. Fidan, "Activity recognition invariant towearable sensor unit orientation using differential rotational transformations represented by quaternions," Sensors (Switzerland), vol. 18, no. 8, Aug. 2018, doi: 10.3390/S18082725.

[25] A. A. A. Alawsi, B. H. Jasim, and S. M. Raafat, "Improved Positioning of Delivery Quad copter Using GPS - Accelerometer Multimode Controller and Unscented Kalman Filter," IOP Conf Ser Mater Sci Eng, vol. 745, no. 1, p. 012018, 2020, doi: 10.1088/1757-899X/745/1/012018.

[26] A. Tayebi and S. McGilvray, "Attitude stabilization of a VTOL quadrotor aircraft," IEEE Transactions on Control Systems Technology, vol. 14, no. 3, pp. 562–571, May 2006, doi: 10.1109/TCST.2006.872519.

[27] A. Benallegue, A. Mokhtari, and L. Fridman, "Feedback linearization and high order

sliding mode observer for a quadrotor UAV," Proceedings of the 2006 International Workshop on Variable Structure Systems, VSS'06, vol. 2006, pp. 365–372, 2006, doi: 10.1109/VSS.2006.1644545.

[28] S. L. Waslander, G. M. Hoffmann, J. S. Jang, and C. J. Tomlin, "Multi-agent quadrotor testbed control design: Integral Sliding Mode vs. Reinforcement Learning," 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, pp. 3712–3717, 2005, doi: 10.1109/IROS.2005.1545025.

[29] T. Dierks and S. Jagannathan, "Output feedback control of a quadrotor UAV using neural networks," IEEE Trans Neural Netw, vol. 21, no. 1, pp. 50–66, Jan. 2010, doi: 10.1109/TNN.2009.2034145.

[30] David. Allerton, "Principles of flight simulation," p. 471, 2009, Accessed: May 15, 2023. [Online]. Available: https://www.wiley.com/en-us/Principles+of+Flight+Simulation-p-9780470754368

[31] "Advances in Aerospace Guidance, Navigation and Control," Advances in Aerospace Guidance, Navigation and Control, 2011, doi: 10.1007/978-3-642-19817-5.

[32] A. Al-Radaideh and L. Sun, "Self-localization of a tethered quadcopter using inertial sensors in a GPS-denied environment," 2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017, pp. 271–277, Jul. 2017, doi: 10.1109/ICUAS.2017.7991436.

[33] "Simscape - MATLAB." https://www.mathworks.com/products/simscape.html (accessed May 18, 2023).

[34] "CAD Translation MATLAB & Simulink." https://www.mathworks.com/help/sm/ug/cad-translation.html (accessed May 19, 2023).

[35] "Enable Simscape Multibody Link Plugin in SolidWorks - MATLAB & Simulink." https://www.mathworks.com/help/smlink/ref/linking-and-unlinking-simmechanics-link-software-with-solidworks.html (accessed May 19, 2023).

[36] "DJI F450 Quadcopter Drone | 3D CAD Model Library | GrabCAD." https://grabcad.com/library/dji-f450-quadcopter-drone-1/details?folder_id=1691767 (accessed May 19, 2023).

[37] G. Andria et al., "Design and performance evaluation of drone propellers," 5th IEEE International Workshop on Metrology for AeroSpace, MetroAeroSpace 2018 - Proceedings, pp. 407–412, Aug. 2018, doi: 10.1109/METROAEROSPACE.2018.8453604.

[38] C. Xiang, X. Wang, Y. Ma, and B. Xu, "Practical modeling and comprehensive system identification of a BLDC motor," Math Probl Eng, vol. 2015, 2015, doi: 10.1155/2015/879581.

[39] P. Manda and S. K. Veeramalla, "Simulation and Modelling of a Brushless DC Motor Using the MATLAB/ SIMULINK Application Environment," ISMEE 2021 - 2021 3rd International Symposium on Material and Electrical Engineering Conference: Enhancing Research Quality in the Field of Materials and Electrical Engineering for a Better Life, pp. 207–210, 2021, doi: 10.1109/ISMEE54273.2021.9774052.

[40] M. S. Pillai and K. Vijina, "Efficient Commutation and Torque Ripples Minimization in BLDC Motor for Hoist Applications," 2018 International Conference on Control,

Power, Communication and Computing Technologies, ICCPCCT 2018, pp. 405–409, Dec. 2018, doi: 10.1109/ICCPCCT.2018.8574281.

[41] "A2212 Brushless Motor Pinout, Wiring, Features, Specs & Datasheet." https://components101.com/motors/2212-brushless-motor (accessed May 21, 2023).

[42] S. Srita et al., "Modeling, Simulation and Development of Grid-Connected Voltage Source Converter with Selective Harmonic Mitigation: HiL and Experimental Validations," Energies 2022, Vol. 15, Page 2535, vol. 15, no. 7, p. 2535, Mar. 2022, doi: 10.3390/EN15072535.

[43] Y. Liu, J. Gao, P. Zhan, and S. Wang, "The Influence of PWM Modulation Mode on the Commutation Performance of Brushless DC Motor," Proceedings - 2020 7th International Forum on Electrical Engineering and Automation, IFEEA 2020, pp. 18–23, Sep. 2020, doi: 10.1109/IFEEA51475.2020.00012.

[44] V. A. Omar, "Modelling and control of a UAV-USV collaboration scheme for fluvial operations," Mar. 2017, Accessed: May 21, 2023. [Online]. Available: https://e-archivo.uc3m.es/handle/10016/26163

[45] F. Candan, A. Beke, and T. Kumbasar, "Design and Deployment of Fuzzy PID Controllers to the nano quadcopter Crazyflie 2.0," 2018 Innovations in Intelligent Systems and Applications (INISTA), pp. 1–6, Jul. 2018, doi: 10.1109/INISTA.2018.8466284.

[46] S. YILDIRIM, N. CABUK, and V. BAKIRCIOGLU, "OPTIMAL PID CONTROLLER DESIGN FOR TRAJECTORY TRACKING OF A DODECAROTOR UAV BASED ON GREY WOLF OPTIMIZER," Konya Journal of Engineering Sciences, vol. 11, no. 1, pp. 10–20, Mar. 2023, doi: 10.36306/KONJES.1172687.

[47] R. P. Borase, D. K. Maghade, S. Y. Sondkar, and S. N. Pawar, "A review of PID control, tuning methods and applications," Int J Dyn Control, vol. 9, no. 2, pp. 818–827, Jun. 2021, doi: 10.1007/S40435-020-00665-4/METRICS.

[48] "Real shading in unreal engine 4 by Brian Karis, Epic Games" https://cdn2.unrealengine.com/Resources/files/2013SiggraphPresentationsNotes-26915738.pdf (accessed May 21, 2023).

[49] R. A. Rosu and S. Behnke, "EasyPBR: A lightweight physically-based renderer," VISIGRAPP 2021 - Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications, vol. 1, pp. 245–252, 2021, doi: 10.5220/0010268902450252.

[50] "Landscape Edit Layers | Unreal Engine 4.27 Documentation." https://docs.unrealengine.com/4.27/en-US/BuildingWorlds/Landscape/Layers/ (accessed May 21, 2023).

[51] "25 tips for Unreal Engine 4 - Ειδήσεις από τον χώρο του Design και Hosting - Design Host." https://designhost.gr/topic/6700-25-tips-for-unreal-engine-4/ (accessed May 21, 2023).

[52] "Object Detection: Models, Architectures & Tutorial [2023]." https://www.v7labs.com/blog/object-detection-guide (accessed May 24, 2023).

[53] "What Is Object Detection? - MATLAB & Simulink." https://www.mathworks.com/discovery/object-detection.html (accessed May 24,

2023).

[54] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 580–587, Nov. 2013, doi: 10.1109/CVPR.2014.81.

[55] C. L. Zitnick and P. Dollár, “Edge boxes: Locating object proposals from edges,” Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8693 LNCS, no. PART 5, pp. 391–405, 2014, doi: 10.1007/978-3-319-10602-1_26/COVER.

[56] B. Zhou, A. Lapedriza, A. Torralba, and A. Oliva, “Places: An Image Database for Deep Scene Understanding,” J Vis, vol. 17, no. 10, p. 296, Aug. 2017, doi: 10.1167/17.10.296.

[57] “Pretrained Deep Neural Networks - MATLAB & Simulink.” https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html (accessed May 24, 2023).

[58] “Anchor Boxes for Object Detection - MATLAB & Simulink.” https://www.mathworks.com/help/vision/ug/anchor-boxes-for-object-detection.html?s_tid=srchtitle_anchor_2#mw_8aaafc00-5deb-4b5e-9ffd-4ce033e0eadf (accessed May 24, 2023).

[59] “Getting Started with YOLO v4 - MATLAB & Simulink.” https://www.mathworks.com/help/vision/ug/getting-started-with-yolo-v4.html (accessed May 24, 2023).

[60] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” Apr. 2020, Accessed: May 24, 2023. [Online]. Available: https://arxiv.org/abs/2004.10934v1

[61] D. P. Kingma and J. L. Ba, “Adam: A Method for Stochastic Optimization,” 3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings, Dec. 2014, Accessed: May 25, 2023. [Online]. Available: https://arxiv.org/abs/1412.6980v9

[62] “Download Pig labeled image classifcation dataset labeled image dataset.” https://images.cv/dataset/pig-image-classification-dataset (accessed May 24, 2023).