T.C.
İSTANBUL GEDİK UNIVERSITY
INSTITUTE OF GRADUATE EDUCATION

ADAPTIVE BLENDED VISUAL LOCALIZATION SYSTEM BASED ON ARTIFICIAL NEURAL NETWORKS FOR UNMANNED AIR VEHICLES

MASTER'S THESIS

Oğuzhan YILMAZ

Department of Defence Technologies

JANUARY 2022

T.C.
İSTANBUL GEDİK UNIVERSITY
INSTITUTE OF GRADUATE EDUCATION

ADAPTIVE BLENDED VISUAL LOCALIZATION SYSTEM BASED ON ARTIFICIAL NEURAL NETWORKS FOR UNMANNED AIR VEHICLES

MASTER'S THESIS

Oğuzhan YILMAZ
(191202003)

Department of Defence Technologies

Thesis Advisor: Prof. Dr. Halit Hami ÖZ

JANUARY 2022

T.C.
İSTANBUL GEDİK ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ MÜDÜRLÜĞÜ

Yüksek Lisans Tez Onay Belgesi

Enstitümüz Savunma Teknolojileri Tezli Yüksek Lisans Programı 191202003 numaralı öğrencisi **Oğuzhan YILMAZ**'ın tez çalışması, 21.01.2022 tarihinde yapılan tez savunma sınavında aşağıdaki jüri tarafından oy birliği ile 90 (doksan) almıştır.

Öğretim Üyesi Adı Soyadı

1) Tez Danışmanı: Prof. Dr. Halit Hami ÖZ

2) Jüri Üyesi: Prof. Dr. Asaf VAROL

3) Jüri Üyesi: Dr. Öğr. Üyesi Ahmad Mostafa Karim KARİM

DECLARATION

I, Oğuzhan Yılmaz, hereby declare that the work presented herein is original work done by me and has not been published or submitted elsewhere for the requirement of a degree programme. Any literature date or work done by other and cited within this thesis has given due acknowledgement and listed in the reference section. (08.01.2022)

Oğuzhan YILMAZ

FOREWORD

In this study, a visual positioning system for UAVs independent of GPS is described. In the first stage, position information is obtained from two sources, deep learning and handcrafted features, and the final position is calculated by adaptive blending through the trained artificial neural network.

With my best regards to my wife, Nagehan Önder Yılmaz, and my family, who were with me both technically and spiritually during this work process.

This work was supported by the Turkish Aerospace Industries, Inc.

January 2022 Oğuzhan Yılmaz

TABLE OF CONTENTS

ABBREVIATIONS

ANN	Artificial Neural Network
BC	Before Christ
CNN	Convolutional Neural Network
DEM	Digital Elevation Map
DL	Deep Learning
DoG	Difference of Gaussian
FAST	Feature-Based Algorithm
FNN	Feedforward Neural Network
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HF	Handcrafted Feature
HM	Handcrafted Method
ID	Identification
INS	Inertial Navigation System
IPS	Instrument Pointing System
İHA	İnsansız Hava Aracı
LIDAR	Light Detection and Ranging
LORAN	Long Range Navigation
LSTM	Long-short Term Memory
MPR	Map Verisimilitude Rate
MSE	Mean Square Error
NCN	Neighbourhood Consensus Network
NET	Network
NN	Neural Network
RADAR	Radio Detection and Ranging
RELU	Rectified Linear Unit
RGB	Red Green Blue
RNN	Recurrent Neural Network
ROT	Rate of Turn
SIFT	Scale-Invariant Feature Transform
Sp-OSM	Super pixel-Based Ordered Spatial Matching
Sp-SGS	Super pixel-Based Sample Graded Strategy
SSM	Semantic Shape Matching
SURF	Speed-Up Robust Feature
UAV	Unmanned Air Vehicle

VBN	Visual-Based Navigation
VGG-Net	Visual Geometry Group Network
YSA	Yapay Sinir Ağı

LIST OF TABLES

LIST OF FIGURES

ADAPTIVE BLENDED VISUAL LOCALIZATION SYSTEM BASED ON ARTIFICIAL NEURAL NETWORKS FOR UNMANNED AIR VEHICLES

ABSTRACT

The recent developments in aviation unmannization have brought along systems that aim to limit the operational capability of unmanned platforms. The primary targets of the systems are generally communication and navigation functions. Most widely used Global Positioning System (GPS) signals are vulnerable to jamming and inaccurate localization is almost inevitable in strategic areas where electronic warfare is at an advanced stage. To cope with this challenge, the search for alternative positioning solutions independent of the Global Navigation Satellite System (GNSS) continues. In this thesis, an Unmanned Aerial Vehicles (UAVs) geolocalization framework based on visual resources is presented. It consists of two independent pipelines receiving data from the same source, handcrafted and artificial neural network based. On the side of deep learning, with trained convolutional neural network (CNN) features extracted from satellite and aerial images are used to localize UAV in GPS denied environments. Since the raw outputs from the neural net are not suitable for high-fidelity matching, they are optimized by three successive methods. The first one modifies the feature map obtained from the characteristics of the two images according to the final potential position of the UAV. The matching coefficients are multiplied by an exponentially decreasing function from the nearest pixels to the farthest ones. The second one joins the process after the first three position predictions and uses them to create an imaginary perspective. If the new location determined in each iteration follows the same direction compared to the previous ones, the spectrum narrows and the outputs become more precise. Otherwise, the angle of the corridor will widen, allowing the direction of the new predictions to change in subsequent ones. The third and final one draws an inertial path during the flight of the UAV, taking into account the technical specs of the platform (max. bank angle, max./min. speed). The altitude-independent horizontal plane is divided into 8 segments, and the coefficient of inertia of each segment increases as the aircraft moves over them. Through these applied methods, the flight characteristic of the UAV refines the textures extracted from the images to be matched before the next step, where the best coefficients are taken, and the others are eliminated. In the last stage of the neural network, clustering is performed with Dbscan, and outliers are removed. Scale-invariant feature transform (SIFT) is used for the handcrafted computer vision side. As well as their working mechanisms the efficiencies of the neural network and SIFT-based sources under variable circumstances such as the amount of light, vegetation and seasonal changes also differs. A real-time adaptive blended localization system is achieved by training the long short-term memory network (LSTM) with the dataset obtained by comparing the outputs of these two locator algorithms with ground truth data for different weather conditions and trajectories. As a result, the proposed system reduces the error by 12 percent compared to the blended ones with constant coefficients. This

result is achieved by force of a new blending method without any change in hardware or content of navigation systems.

Keywords: *Artificial neural network, computer vision, visual based navigation, blended navigation*

İNSANSIZ HAVA ARAÇLARI İÇİN ADAPTİF HARMANLANMIŞ YAPAY AĞ TEMELLİ GÖRSEL NAVİGASYON SİSTEMİ

ÖZET

Havacılıkta insansızlaştırma alanında yaşanılan son gelişmeler söz konusu insansız platformların operasyonel faaliyetlerini kısıtlamayı amaçlayan sistemleri de beraberinde getirmektedir. Bu sistemlerin öncelikli hedefleri genellikle haberleşme ve navigasyon fonksiyonlarıdır. En yaygın kullanıma sahip Küresel Konumlandırma Sistemi (GPS) sinyalleri karıştırmaya karsı savunmasızdır ve elektronik harbin had safhada olduğu stratejik bölgelerde hatalı konumlandırma neredeyse kaçınılmazdır. Bu güçlükle başa çıkmak için Küresel Navigasyon Uydu Sistemi'nden (GNSS) bağımsız alternatif seyrüsefer çözümleri arayışı devam etmektedir. Bu tezde görsel kaynaklardan beslenen İnsansız Hava Aracı (İHA) konumlandırma sistemi sunulmaktadır. Sistem bilgisayarlı görme ve Yapay Sinir Ağı (YSA) tabanlı olmak üzere aynı kaynaktan beslenen iki bağımsız ardışık düzenden oluşur. Derin öğrenme tarafında Eğitimli Evrişimli Sinir Ağı vasıtasıyla uydu görüntüsü ve havadan çekilen görüntülerden elde edilen özellikler GPS olmayan ortamlarda konumlandırma için kullanılır. Sinir ağından elde edilen ham çıktılar yüksek doğrulukta eşleştirme için uygun olmadığından üç sıralı yöntemle optimize edilirler. İlki, iki görselden elde edilen özellik haritasını İHA'nın son potansiyel konumuna göre modifiye eder. Eşleşme katsayıları yakın piksellerden uzak olanlara doğru üstel azalan bir fonksiyon ile çarpılır. İkinci metot döngüye ilk üç tahminden sonra girerek hayali bir perspektif oluşturmak için bu tahminleri kullanır. Her bir iterasyon için belirlenen konumun doğrultusu bundan önce belirlenenlerinkiyle ortaksa spektrum daralır ve sonraki çıktılar daha kesin hale gelir. Aksi takdirde, koridor açısı genişleyecek ve sonraki adımlarda yeni tahminlerin yönlerinin değişmesine izin verecektir. Üçüncü ve son metot ise İHA'nın uçuşu sırasında teknik özelliklerini (maksimum yatış açısı, maks. / min. hız) dikkate alarak bir atalet yolu çizer. İrtifadan bağımsız yatay düzlem sekiz parçaya bölünmüştür ve her parçanın atalet katsayısı, uçak üzerlerinde hareket ettikçe artar. Uygulanan bu yöntemler sayesinde bir sonraki safha olan en iyi katsayıların seçilip diğerlerinin elendiği eşleme safhasından önce görsellerden elde edilen modeller İHA'nın uçuş karakteristiği ile rafine edilir. YSA'nın son safhasında Dbscan kullanılarak regresyon yapılır ve aykırı veriler çıkarılır. Bilgisayarlı görme tarafında SIFT kullanılır, ışık miktar bitki örtüsü ve mevsimsel değişiklikler gibi değişken koşullar altında çalışma mekanizması farklı olan YSA ve SIFT tabanlı kaynakların verimliliği de farklılık göstermektedir. İki ayrı kaynaktan alınan çıktılar Uzun Kısa Vadeli Hafıza Ağı vasıtasıyla değişken senaryolar için eğitilip reel değerlerle karşılaştırılarak gerçek zamanlı adaptif harmanlanmış konumlandırma sistemi elde edilir. Sonuç olarak önerilen sistem sabit katsayıyla harmanlanmış olanlara göre hata oranını yüzde 12 düşürmektedir. Bu sonuç donanımda veya navigasyon sistemlerinin içeriğinde bir değişiklik yapılmadan yeni bir harmanlama metodu sayesinde elde edilmektedir.

Anahtar kelimeler: *Yapay sinir ağları, bilgisayarlı görme, görsel tabanlı navigasyon, harmanlanmış navigasyon*

1. INTRODUCTION

While autonomization is one of the most studied and developing topics of the century, when humankind's centuries-long passion for exploring the sky is added to this, UAVs become an area where the limits of technology are drawn. Positioning systems are at the forefront of the factors that allow unmanned platforms to sustain themselves. Existing positioning systems are heavily dependent on satellite-based service mechanisms. These setups have disadvantages such as inaccurate measurement and vulnerability to electronic jamming and spoofing, especially in harsh conditions. Although precautions are taken against electronic jamming, there is no permanent solution. Under all these conditions, alternative solutions are sought for the localization of UAVs. Visual, celestial, and magnetic anomaly-assisted systems can be given as examples. Revolutionary developments in the field of computer vision and artificial intelligence in the last few decades make visual-aided ones more assertive than others. The basic working logic of navigation systems based on visual resources is to unsupervised evaluation the images taken from the ground or compare them with the stored in the database. The outputs obtained can give partial or complete information about the possible location of the platform. The position is obtained by matching real-time images from the ground with those of the satellite. Image matching is constitutively done in two ways: Handcrafted Methods (HM) and Artificial Neural Network (ANN) methods. Although the basic working logic is the same, the image feature detection and description techniques used in these frames differ. The parameters that determine the features obtained from the images to be matched are predefined in HM and are calculated by the hidden layers in ANN. Apart from the ways in which these two systems obtain datum, their efficiency also differs under certain conditions. Data are taken simultaneously from several independent sources, even on aircraft using the most reliable source positioning configurations. Multi-sourcing is more crucial for localization architectures that are still proving themselves. In this study, an adaptive blended visual positioning system is described. During blending, two independent architectures, HM-based and ANN -based, are operated. SIFT is used in the HF-

based algorithm. The ANN-based system consists of the trained CNN architecture VGG-16 and a set of sequential methods. These two sources are tested at different seasonal and daytime periods, and an LSTM-NET is trained with the outputs. During the flight, the trained network manages the weighting of the two resources according to the characteristics of the terrain in real-time. Thus, the blending that will give a better result according to the current image in the flying region is done and the final location output is obtained.

1.1.Motivation

Unmanned platforms are increasing their effectiveness in all industries. As in every field, aviation continues to be the industry where bleeding-edge technologies are applied. However, due to its nature, it is more complicated to predict the contingencies that flying platforms will encounter. For this reason, it takes more time and more challenging for innovations to prove themselves. Visual navigation is one of the most needed of these innovations as it eliminates the dependency of UAVs on satellite-based structures. Therefore, every step towards the autonomization of visual-based systems to be taken in this field will help us to overcome the challenges.

1.2.Objective of the Research

The purpose of using multiple resources in critical systems of existing air platforms is usually to increase redundancy and prevent false outputs. In such structures, as long as the sources produce valid data, no comparison is made between them to increase the accuracy. This situation causes in-limit noise or deviations in any of the outputs to affect the final result negatively. In order to make the comparison, it is necessary to predict under which conditions the efficiency of the systems differs.

This study aims to increase the accuracy of the final output of a system that receives visual localization data from dual sources by adaptive blending. To provide this prediction, the LSTM network is trained using the outputs obtained by simulating two positioning algorithms in different scenarios, and this network is used for priority weighting of the resources in real-time.

1.3.Scope of Thesis

This thesis is organized into eight chapters. The first chapter is the introduction. The purpose and motivation of the study are explained in this chapter.

Literature review is represented in the Chapter 2.

The first of the two positioning systems included in the study, NN-based is described in Chapter 3. This chapter consists of 4 parts: dataset preparation, which describes the four stages of data set preparation, feature detection and description, which describes feature extraction, improving methods, which describes three development methods, and image matching, where image matching is performed.

Chapter 4 describes the second positioning system, the Handcrafted Method Based Navigation system.

Chapter 5 covers the simulation of testing two positioning systems and the training of LSTM and ANN networks with post-test data.

Chapter 6 presents real-time adaptive blending tests and results of positioning systems using trained networks.

Open to interpretation issues encountered throughout the study are detailed in Chapter 7.

In Chapter 8, the results obtained at the end of the study and future work are mentioned.

2. LITERATURE REVIEW

2.1.Navigation Systems

Exploring has always been one of the most primary impulses of human beings. The history of wayfinding methods developed for use during explorations dates back to 3000 (National Geographic Society, 1994). Primitive people observed the movements of the stars and the sun to find direction. The year was not yet 1000 when they started using the first magnetic field-sensitive instruments (Needham, 1986). 500 years from that, they could not only form an opinion about their direction but also their position. The most important proof of this is the astrolabe shown in Figure 2.1. It is the oldest navigation tool found and is thought to belong to the 15th century (Morelle, 2017). The astrolabe was used to predict the movements of the sun, to measure the height of any elevation, and to find direction. The sextants, which can be measured more precisely and are still used for navigation, are three centuries old, as shown in Figure 2.2. Later, sextants started to be used together with the chronometers, which were invented in the 18th century and the purpose of which was to accurately measure the time of a known fixed position.

Figure 2.1.Astrolabe

Figure 2.2.Sextant

Technology is shaped according to the needs of humanity and as long as human curiosity continues, the search for its place in the universe will continue. The fact that nature observations cannot be made under all conditions, the instruments that are observed need to be calibrated and their sensitivity is low, have caused the place and direction-finding habits based on nature observation to be replaced by artificial systems. In other words, the age of radio has begun _(Norman, 2012). Decca was the first navigation system established in 1936 using radio signals. Working in the low-frequency band, it had a range of up to 450 km and could give accurate results up to 200 meters. Loran, one of the antennas of which is shown in Figure 2.3. and started to be used ten years later, had a sensitivity of up to 180 meters in the ultra-low frequency band. For radio-based systems to perform positioning, it is sufficient to receive signals from two slave stations at the same time, whose location is fixed. The intersection point of the circles centred on the signal sources represents the current position. Although the sensitivity of these structures was satisfactory, the search for systems with global coverage continued, as they were only suitable for regional use. In addition, existing systems inherently had disadvantages such as inaccurate and noisy measurement. Day-to-day drift was a major source of problems for Doppler radars _(Burns, 1963). Omega was founded, which finally became operational in 1971 and will remain in use for 26 years. Omega's two km sensitivity was low, but it had a coverage area of 10,000 km (Asche, 1972).

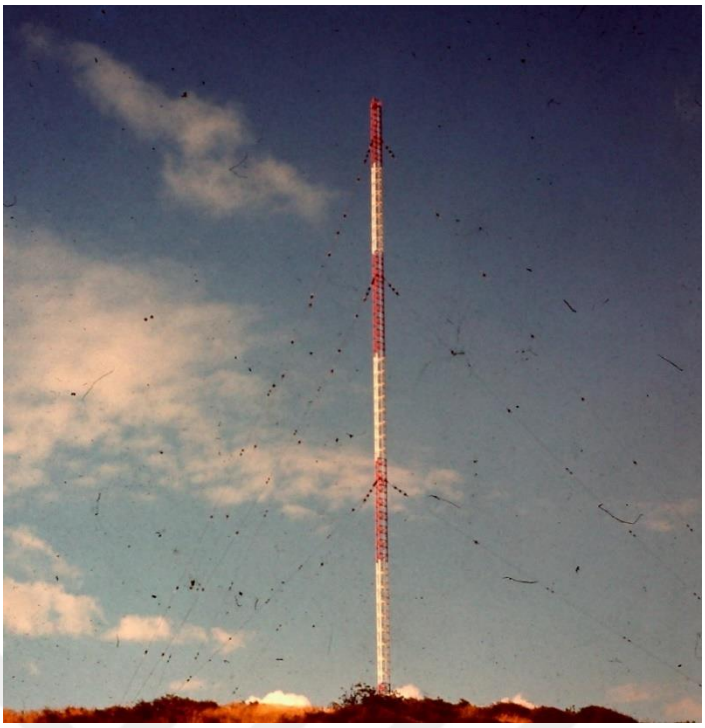

Figure 2.3. A typical LORAN antenna (1972)

With the effect of the transition to the space age and the cold war, a new page in the history of humanity opened in the name of navigation after 1950, it's called artificial satellites. The world's first satellite, Sputnik 1, shown in Figure 2.4. was put into orbit in 1957. In the following seven years, approximately 400 satellites were launched.

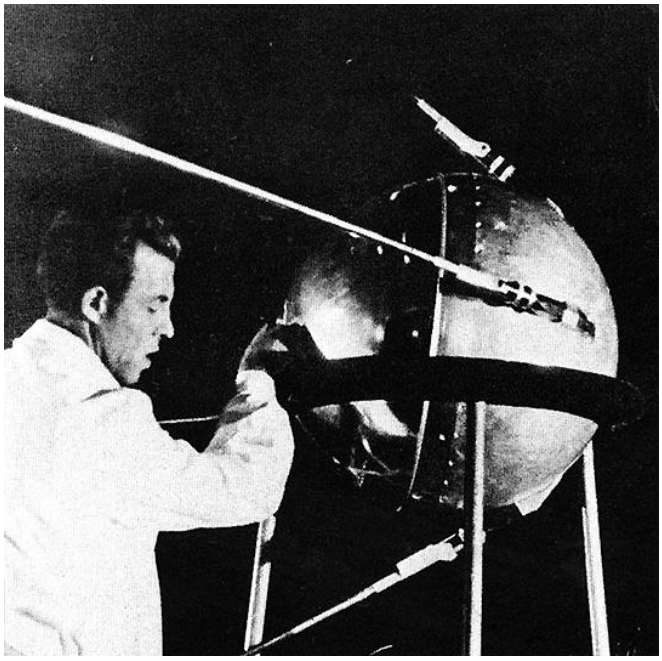

Figure 2.4. Sputnik 1

By 1964, the first global satellite-based navigation system, Transit, was launched. It had an accuracy of 400 meters and the coverage was the best. The project, which started with the name Navstar in 1973, replaced Transit with its 24 active satellites in 1996. The system, which was later named GPS, is still in use. Today, there are a total of 150 active satellites in orbit around the world _(Mohanta, 2021). GPS transmit signals continuously and have a one-way broadcast stream. They have signals in five different frequency classes. L1 is the first used signal type, it is split into two separate parts and is slower than the others. One is for general use and the other is for military use. The frequency of L2 is faster than L1 and is also split for two separate uses. L3 frequency is used as nuclear explosion detection system while L4 transmission is not available. L5 reserved for Safety of life. It is a civilian use and serves aviation safety. Since they do not have a direct connection with GPS devices, they do not have an upper limit of service. Four or more GPS satellites must be visible for accurate results. This is at least five for aircraft. Because the earth's surface is referenced for its vertical position. GPS positioning is represented in Figure 2.5.

Figure 2.5. GPS Satellite System

Alternative frameworks to the existing positioning system are sought, developed, and even used as an auxiliary resource today, as it always is. Examples of these are celestial, radar-aided, lidar-based, magnetic anomaly aided, INS, visual-based navigation systems, and Locata.

2.1.1. Celestial Navigation

The starting point of celestial navigation is to observe celestial bodies. The more calibrated and precise the instrument being observed, the more accurate the output will be. The essential concept of positioning based on observation is to refer to a fixed object or objects. Celestial bodies also constitute the most reliable source in this context. Two or more-star observations are sufficient to determine the position of a moving platform. For celestial navigation, the angular rotation rate of the Earth is relatively slow, so a time reference is not needed for high precision (F. Pappalardi, 2001). In addition, the star tracker celestial localization systems shown in Figure 2.6. are used in explorations made outside the Earth. However, even if there are great improvements in the observation instruments, since the number of photons coming from the stars is constant, no significant improvement in update speed and accuracy is expected in the future (Liebe, 2002). Celestial, which is one of the most primordial navigation systems, could not resist radio-induced formations and remained in the background, since it used relatively primitive tools and had low sensitivity in the 20th century. However, it is known that the Stella program launched for the Navy has a precision of up to 30 meters, even if it is not made available to the general public (Alan D., 1999). Today, modernized versions continue to be used as an auxiliary system, especially in ballistic missiles.

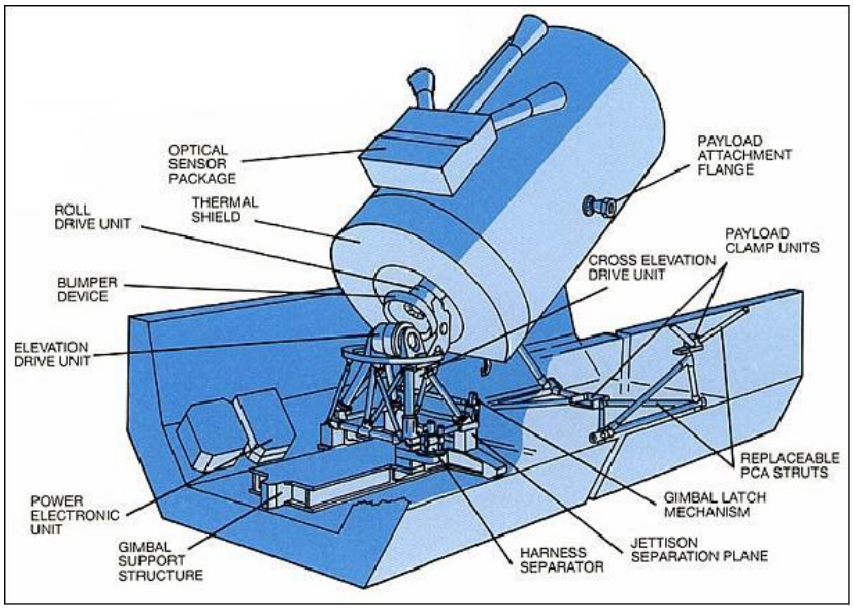

Figure 2.6. Illustration of IPS (Instrument Pointing System), image credit: NASA

2.1.2. Radar-aided Navigation

Although navigation techniques with radar devices are not used autonomously, this technique is used in many situations. The most common method is to refer to the centers that appear in the radar scan and whose location is known. In addition to fixed points, platforms moving along a defined route can also be referenced. The most important example of this is The Franklin continuous radar plot technique, developed by Master Chief Quartermaster Byron E. Franklin, who gave the technique its name. According to this method, it can be determined whether the ship is deviating from its course and how the correction should be made if it has deviated from the route by taking the ships detected on the radar and following a certain path as reference (National Geographic Society, 1994). Another purpose of the use is to increase the aiming capabilities of missiles sent to the target, as shown in Figure 2.7. Correction can be sent for higher accuracy by using the radar reflections of the target and missile (R. S. Ornedo, 1998).

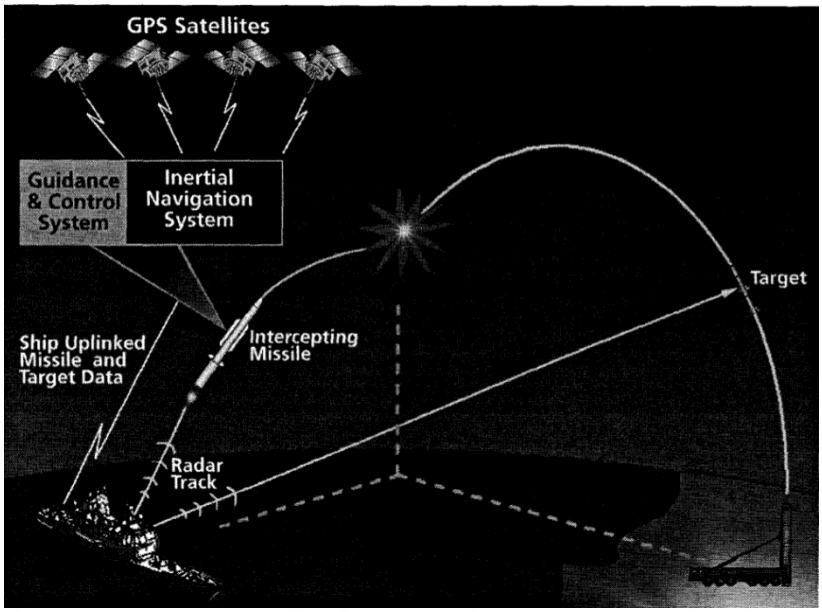

Figure 2.7. A Typical Tactical Ballistic Missile Defence Mission, image credit: [12]

2.1.3. Lidar-based Navigation

Light detection and ranging (LIDAR) is an emerging technology that determines the distance of objects using laser beams. The system sends a beam of light from the aircraft to the ground. The set of points where the rays hitting the ground and rotating are reflected according to their arrival angles and travel times are simulated. There can be more than one return for each unit beam emanating from the laser source. This helps to get an idea of the environment encountered. Even the vegetation diversity of the land is predictable. The extracted map is called point cloud as shown in Figure 2.8. The system, which sends an average of 200 thousand pulses per second, can create a three-dimensional map consisting of millions of points in seconds. Position information can be determined by comparing the obtained map with the help of Digital Elevation Models (DEMs). DEMs are topographic maps of the Earth's surface. Modern lidar sensors can operate with an accuracy of five meters at a distance of five km. Although it cannot be used for aircraft serving at high altitudes, it is used as a navigation system for land platforms and small class UAVs, especially in in-door area service. It is also suitable for use in landing and take-off sequences of aircraft, for example, a lidar-guided UAV can land on a vehicle in motion (J. Kim, 2017).

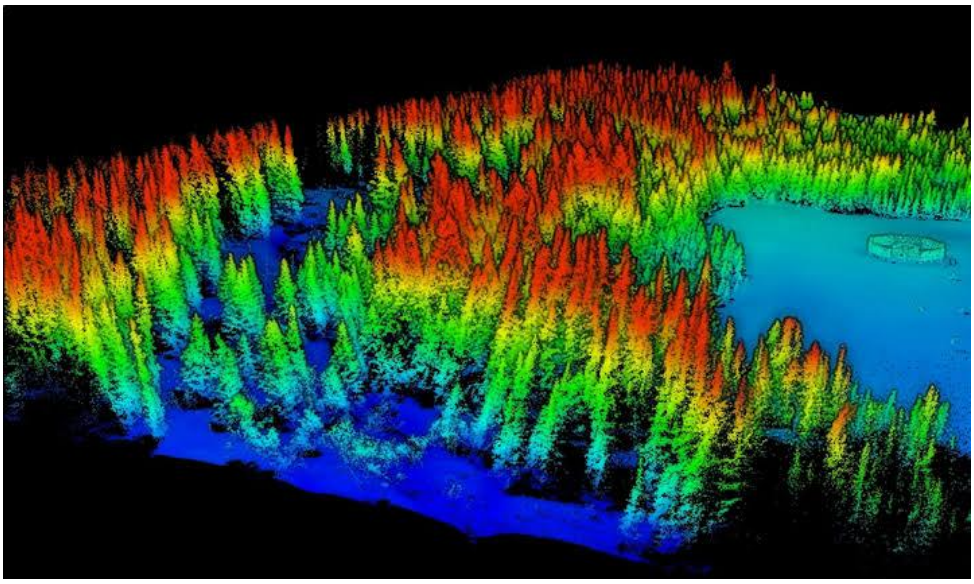

Figure 2.8. Lidar Scanned Terrain

2.1.4. Magnetic Anomaly Aided Navigation

Earth's magnetic field ranges from 25,000 to 70,000 gammas. It is a dipole offset by ten degrees with respect to the axis of rotation. The earth's magnetic field changes over time because the molten iron alloys in the center that make up the domain are affected by the earth's rotation. The magnetic field gradually decreases towards the equator. In addition, landforms change the strength to such an extent that it can be expressed by hundreds. An anomaly is a deviation in the earth's magnetic field. The difference between the measured and the expected one represents the anomaly. As shown in Figure 2.9., there are maps expressing the anomalies around the world. They are created with data obtained from aircraft flying at low altitudes. When the inputs detected during the flight is compared with the maps in the memory, inferences can be made about altitude and location. Since the anomaly effect will decrease as the hight increases, the measurement accuracy decreases at higher altitudes. The reduction of changes in the anomaly also leads to less precise inferences at higher. This study (Raquet, 2017) shows that 13 meters distance root mean square error is obtained by using a high-quality anomaly map at low altitudes. In the future, as the capabilities of the sensors used to improve, it may be possible to position with the same precision at high altitudes.

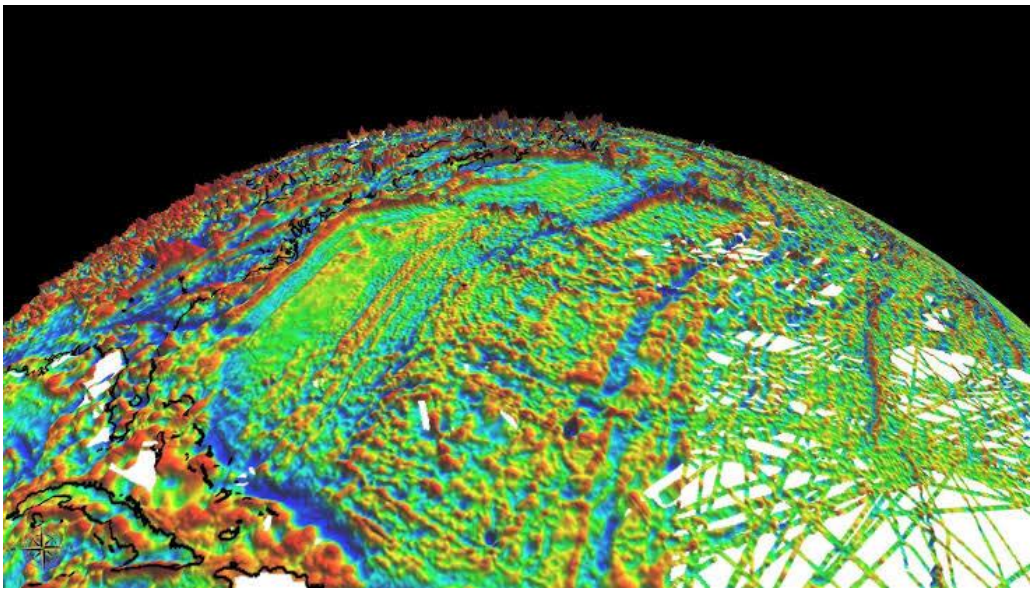

Figure 2.9. Earth Magnetic Anomaly Map

2.1.5. Inertial Navigation

Inertial Navigation Systems (INS) use a computer, motion and rotation sensors to calculate changes in position, orientation, and velocity. The INS is independent of any external references or parameters connected to it. With this feature, it differs from other systems in terms of the measurement technique. It makes these calculations according to the starting point at time t0 and stacks the estimations from there. Since each calculation is made according to the current position obtained in the previous one, it also causes the error made per unit calculation to accumulate. Even the best accelerometers cause a deviation of 50 meters after 17 minutes. This means that at the end of one hour, the measurement of a standard system deviates by 1000 meters as shown in Figure 2.10. Ninety percent of the source of error is due to the system's own error (W. Yang, 2020). For these reasons, INS systems cannot be used as a neolithic navigation framework. They need a second source from which they periodically receive corrections.

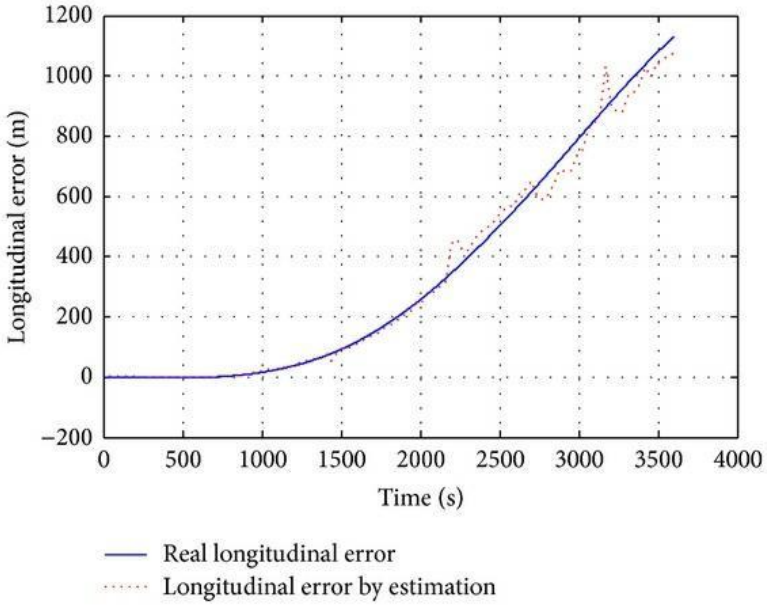

Figure 2.10. Time Dependent INS Error Graph

2.1.6. Visual Based Navigation

Image-aided navigation systems are based on comparing images taken from the earth with those in the database. The higher the resolution of the received and existing images, the better the positioning sensitivities will be. The resulting images are made suitable for use with some computer vision techniques. These corrections may include noise in images, unwanted reflections, and image shifts. In addition, the homography of the resulting image may differ from the one to be used as a reference. In this case, additional correction techniques are used. Then, this data is passed through image interpretation filters and used for matching. As the resemblance rate of the characteristic features obtained from the pictures increases, the probability of matching also increases. Although this positioning technique constitutes the research area of this study, its application area is the widest and developing among the others.

2.1.7. Hybrid and Other Navigation Systems

Other localization systems are sonar/acoustic, visual odometry and laser aided navigation. Sonar/acoustic navigation is based on determining the elevations on the ocean surface and estimating the position used in submarines. The visual odometry method is about determining the distance travelled in unit time according to the movements of the objects in the consecutive images. It is similar to INS in terms of

working logic. Finally, laser aided navigation is the interpretation of elevation patterns obtained using a laser altimeter _(Yilmaz, 2013). Hybrid navigation systems are the use of one or more of the previously mentioned structures. The most important advantage of them is to ensure the integrity of the systems where each other is insufficient. Table 2.1. shows the sensitivity of each system to environmental conditions. Although it is challenging to establish and maintain hybrid systems, their dependence on environmental conditions and external factors is quite low compared to others. The most widely used of these systems are GPS-INS, GPS-INS-VBN.

Table 2.1. Affectability of Navigation Systems

	Cloudness	Flatland	Highland	Dark Field	Jamming/ Spoofing	Electromagnetic Interference
Celestial Navigation	High Vulnerability	-	-	-	-	-
Radar-aided Navigation	-	-	-	-	High Vulnerability	-
Lidar-based Navigation	High Vulnerability	High Vulnerability	-	-	-	-
Magnetic Anomaly Aided Navigation	-	-	-	-	-	High Vulnerability
Inertial Navigation	-	-	-	-	-	High Vulnerability
GPS / GNSS	-	-	High Vulnerability	-	High Vulnerability	-
Visual Based Navigation	High Vulnerability	-	-	Low Vulnerable	-	-
GPS-INS	-	-	Low Vulnerable	-	Medium Vulnerable	Low Vulnerable

2.2. Artificial Neural Networks

Artificial neural networks (ANN) are data processing technique inspired by biological NN. Each unit in which parameters are processed is called a neuron. In the human brain, neurons are connected to each other by synapses. These structures provide the transport of signals that express information. In ANN, these signals are real numbers and the output of each neuron is formed by processing the input with a non-linear function. Each unit has its own weight and this weight affects its output. In addition, each neuron has its own threshold value. No output is transmitted unless

this threshold is exceeded. We can compare this structure to the run-off model. The resource expresses the input and is divided into branches and accumulates in certain places. The deposits formed in these beds are called lakes, which means neurons. When it crosses a certain threshold, it overflows and continues on its way. Eventually, the data coming out of the source and overflowing from different basins converge on the seabed. The final stage is the output of the network. While the depth of the accumulation corresponds to the threshold values of the neurons, the mineral content of each lake bed is different. The river with the highest flow rate among the branches of the river is the most dominant in the content of the sea to be formed as a result. These flow rates also express the weight of the neurons. The output is obtained by processing the data from the source in different branches.

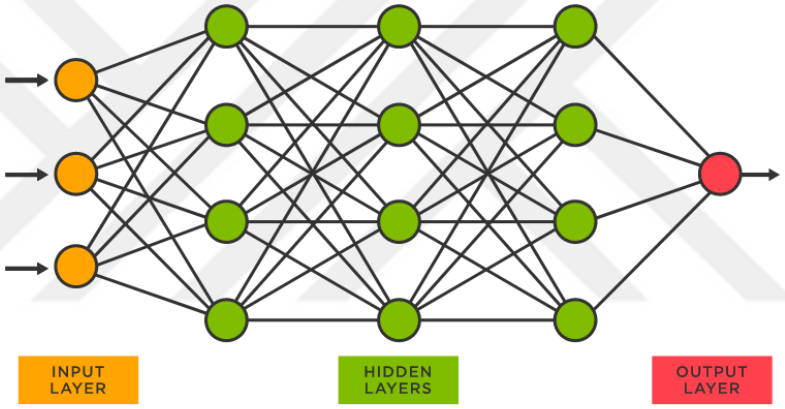

Figure 2.11. Artificial Neural Network Model

A typical ANN model is shown in Figure 2.11. While the input layer is the section where the data is included in the network as input, the part that creates this output is called the output layer. Between these two sections are hidden layers. While the input and output layers can be interpreted, the hidden layers do not have any labels. According to the training approach, ANNs are divided into three as supervised, unsupervised and reinforcement learning. In supervised learning models, feedback is given based on the error of the network. The difference between the targeted output and that obtained from the net is the error. This difference is exploited when training ANNs. The greater the error, the more the weightings are changed. This process is performed many times to ensure that the output converges to the desired value in each iteration. When the adequate result is obtained, the training is completed and the

trained ANN is obtained. Unsupervised models do not receive any feedback to the framework. The network classifies the data using the predefined cost function. There are rewards and punishments in the Reinforcement learning technique. This can also be called the trial-and-error method. It refreshes the weights until the network outputs successfully. The artificial networks used in this study are included in the supervised model and three different types of ANNs are used. These are feedforward neural network (FNN), recurrent neural network (RNN), and convolutional neural network.

2.2.1. Machine Learning Terminology

Under this title, some terms that will be used later and their meanings will be mentioned. Activation function, is to introduce non-linearity into the output of neurons. Layers are neurons that contain an activation function and are in equivalent stages in the architecture. In order to better express the editing in the works, it is called a layer without the activation function. Dense layers are layers in which each neuron is connected with each of the previous ones. Cells are units that contain specialized mathematical operations separate from the familiar neurons. For example, in a CNN network, the filtered unit is expressed as cell. But they are all neurons in general terms. Backpropagation is a supervised learning algorithm using gradient descent. In each iteration, feedback is given with backpropagation and training is performed by changing parameters such as weight. Weights is the number that determines how much the neuron it represents contributes to the process and how much it will be affected in the back propagation phase. Loss function is the difference between the predicted and the targeted output. According to this difference, since the network is trained, it is necessary for learning and measures the penalty. Cost function refers to the average loss of the entire training network. Learning rate is a hyperparameter that determines the effect of the transactions to be made with the result of the loss function. The larger it is, the greater the change of parameters during backpropagation. Epoch refers to a cycle of the whole NN architecture. Iteration is the number of batches sent to the network. Batch is the number of samples sent to the network in each iteration.

2.2.2. Feedforward Neural Network

FNN is one of the simplest forms of ANN. It has activation function. By means of this function, the value to be transmitted to the next neuron is optimized. In addition, activation functions adapt the network to non-linear behaviour and enable the network to get the relationship between input and output. FNNs can be in single-layer or multi-layer perceptron networks. Perceptron is the simplest neural network. It is single-layered and can provide information whether an entry belongs to a certain class or not. Multi-layer perceptrons can solve nonlinear and more complex problems. However, FNNs have difficulties in solving computer vision problems with huge dataset and problems using sequential data. The main reason for these is the necessity of reducing the inputs to one dimension. This requires both an immense amount of processing capacity and loss of spatial properties. In order to overcome all these limitations, RNN and CNN networks have been developed.

2.2.3. Recurrent Neural Network

The feature that distinguishes RNNs from others is their memory qualification. Neurons keep the information they had in the previous step in their memory and use them in the next steps. Thus, they can calculate the relationship of time-dependent or sequential information with each other. They also have varying weights when calculating this relationship. For example, as part of the content censorship project for children, we want to extract violent images from the video recording. The ranges that the normal mesh will determine will usually be where weapons or sharp objects are clearly visible. However, the memory networks trained with the same data will also detect the images from which it obtains clues about the factor of violence. This is because, unlike other networks, they can calculate the interconnection of consecutive scenes. RNNs are widely used in obtaining texts by interpreting images, translation programs and video processing. However, RNNs are insufficient when there are unusual changes in sequential data. For example, RNNs fail to calculate in the presence of time lags greater than 5 - 10 individual time steps between input events and target signals. In addition, the effect of the stored data on the outputs of the neurons either increases excessively over time or decreases and disappears. A recent model, “Long Short-Term Memory” (LSTM) is not affected by this problem (F. A. Gers, 1999)

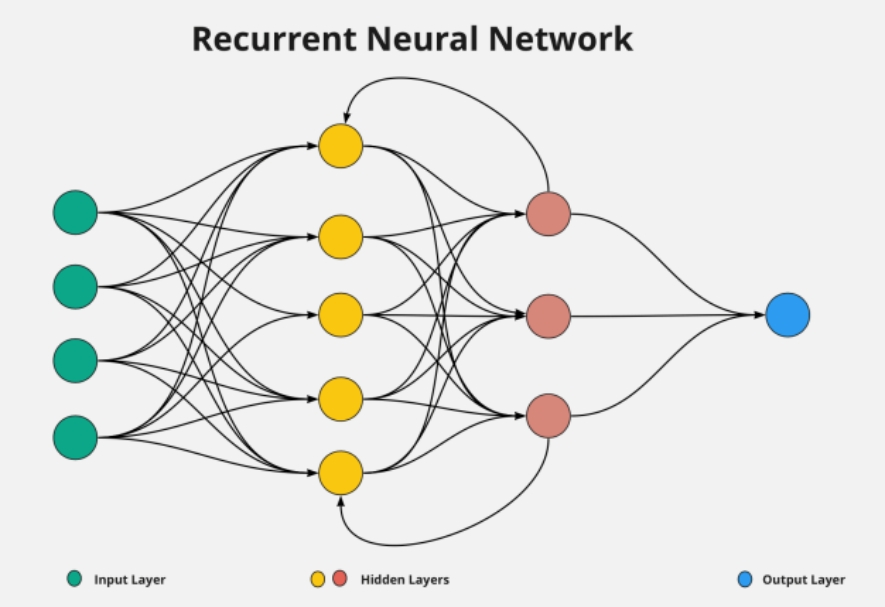

Figure 2.12. Recurrent Neural Network Model

2.2.3.1.Long-Short Term Memory Neural Network

LSTM networks are a type of RNN. It is a complex area of deep learning (DL). It is designed to overcome vanishing and exploding gradient problems _(Hochreiter, 1997). LSTM can perform the sequential processing task where a hierarchical structure may exist but cannot be determined exactly. While standard RNNs contain a single tanh layer, LSTMs contain four different layers in communication with each other as seen in Figure 2.13. These are called cell states and gates. The cell state can be called the memory of the network. Doors, on the other hand, determine which ones are necessary and which ones are unnecessary while this information is being carried. In doing so, they process the data with the activation function. There are three ports: Forget, input and output. Forget gate is the gate where it is determined which information to be forgotten. Input gate cell makes state updates. It decides whether to update the previous and current information according to the sigmoid result. The output gate determines the data to be given as input to the next cell. LSTM networks are used in areas such as voice and handwriting recognition and imitation.

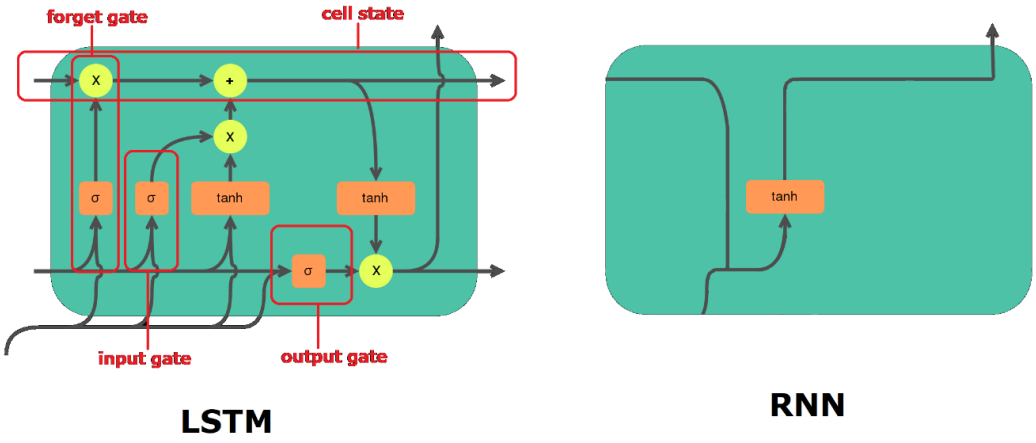

Figure 2.13. LSTM and RNN

2.2.4. Convolutional Neural Network

CNNs are especially used in the field of computer vision, by means of the kernels they contain in their architecture. We can compare kernels to frames. As seen in Figure 2.14., these frames are moved over the input image, thereby reducing the size. "0" values placed around the input are padding. Paddings are used in order not to lose the properties of the pixels on the edges. The kernel filter is multiplied by the stride value, which determines how many pixels the filter will skip at each step. While doing this, the characteristic features of the picture are not lost. The reason why this is needed is to make it easier to process large data such as images and not to lose spatial properties. The formula below expresses the new dimensions of an image whose dimensions are given as $w_1 x h_1 x 3$ by processing with $f_\omega x f_h$ kernel. p_w denotes the padding given on the vertical axis, while S denotes the stride.

$$w_2 = \frac{w_1 - f_\omega + p_w}{s} + 1 \quad \textbf{(2.1.)}$$

$$h_2 = \frac{h_1 - f_h + p_h}{s} + 1 \quad \textbf{(2.2.)}$$

When the formula is applied:

$Input\ size: 3x3x1$

$Kernel\ size: 2x2$

$Padding: 2$

$Stride: 1$

$Output\ size: 4x4$

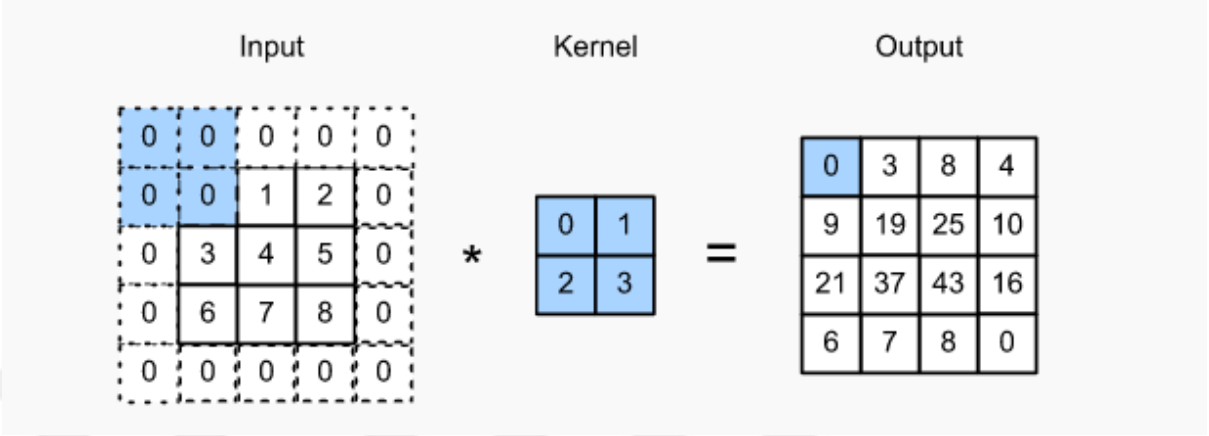

Figure 2.14. Kernel Operation with 1 Padding and 1 Stride

A CNN architecture is shown in Figure 2.15. In the input layer, image data with the size of 28x28x1 is given. The first number represents the width of the image and the second represents the height. Since the picture is black and white, the third dimension is one. If it was coloured, the last number would be represented by three. In the first step, n1 different kernels of 5x5 dimensions are applied and this layer is called the convo. Padding and stride values are as (1.1), (0.2), (1.1), (0.2) for each step, respectively. In the second step, the pooling layer, max-pooling kernel is applied and while the number of kernels remains the same, the data size is halved by virtue of the stride value. The third step is like the repetition of the first step. In the fourth step, the same operations are performed as in the second. As a result, the 28x28x1 size data is reduced to 4x4xn2 size. Here, n2 is the number of applied filters. The more filters are applied, the more different features are obtained from that image. However, this is a factor that increases the processing load. In summary, what is done here is to reduce a 28x28 pixel image to a 4x4 image. While doing this, the picture is reproduced in such a way that certain features come to the fore. We can obtain a total of forty-nine 4x4 images without changing the total data density.

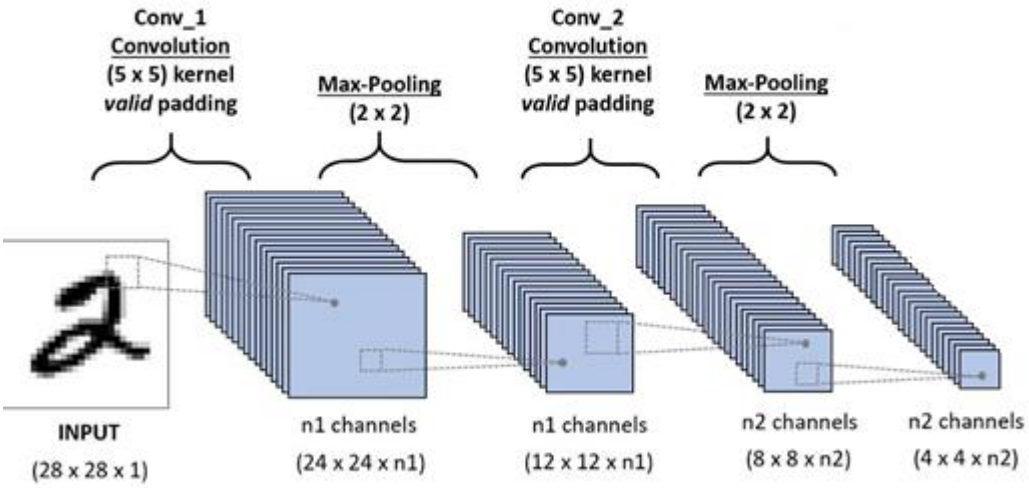

Figure 2.15. Convolutional Neural Network

2.3. Image Matching

Computer vision is important in the field of autonomization as they enable artificial intelligence to recognize their environment and act more consistently. These methods support decision mechanisms with limited data under certain conditions, such as texture recognition, image analysis and remote sensing. The common challenge to overcome is image matching. Although it is simply the matching of two or more pictures, the visuals and the purpose of the study change the methods used. Examples of these are feature matching, dense matching, patch matching, graph matching and point registration. In this study, in order to express the similarities of the picture given for positioning in the big picture that serves as a map, first of all, remarkable features should be determined in the pictures. The first step of this matching is feature detection. The second step is to match the detected features. In order to be matched, they must first be expressed. This is also called the feature description. In the last step, the obtained data is matched. On the other hand, there are some handicaps of the feature-based matching method. Assuming that N feature points are obtained from the images, N! numbers represent potential matches, which is a huge number for high-resolution images. In addition, noises and repetitive motifs in pictures can lead to incorrect matching. Therefore, the use of additional development methods is inevitable.

2.3.1. Feature Detection

As mentioned before, the features are unique to the images themselves. Although there are common shapes in every picture, if they are correctly identified, a unique pattern can emerge. HF-based feature detection methods detect previously outlined shapes. This can be an edge, corner, oval structure or ridge. But the most used feature is the points. Because the pixels are easy to extract and they can come together to form other formations. A useful feature point for matching should be easy to detect and fast to process _(Ma, 2021). Examples of feature detectors are Canny, Sobel, Harris, FAST, SIFT and SURF. On the other hand, NN-based feature detection is done with the parameters obtained by the trained network without any definition or outline. For example, let's say we're doing a project where we match people's faces and want to identify relatives. If we are going to use the HF method, we focus on elements such as eyes, eyebrows, mouth so that we can achieve efficiency. When we use DL, the trained network defines its own elements and it is not possible to give specific names or describe them. Maybe it parameterized glabella[1], but as long as it works efficiently, trying to make sense of it won't help.

2.3.2. Feature Description

After the features in the images are detected, there must be an "identity" describing them so that they can be recognized anyway. This ID may include information such as the type of feature, its location, the distinguishing features of neighbouring pixels. Although there is no standard, each algorithm contains a different descriptor. Description process can be evaluated in three steps: local feature extraction, spatial pooling and feature. First of all, local features detected in images are extracted. The second step is to collect local features to get an overall description. When this process is done, the fingerprint of the image is obtained. The last process is applied to increase the matching performance of the obtained descriptor.

[1] A slight ridge, browbone, interbrow mound, located in the part of the forehead between the two brow ridges and important in anthropology.

2.3.3. Feature Matching

The matching task is to determine the similarity ratio in two images. It plays an important role in the image matching pipeline. Considering the method followed, it is roughly done in two ways. The first one is the direct one in graph matching and point set registration. The way followed here is to match the feature set between the two images according to the similarity ratio as in the image. This method is useful in tasks such as face recognition, human body discrimination. The second method is indirect. In this method, features are matched mutually. For the features in the first image, the similarity ratios for each of the features in the opposite image are determined. Outliers are eliminated and odds above a certain threshold are considered successful matches. The areas that benefit from this method are usually on the matching of fixed structures or shapes under different angles or conditions. The Figure 2.16. shows a structure paired in this way.

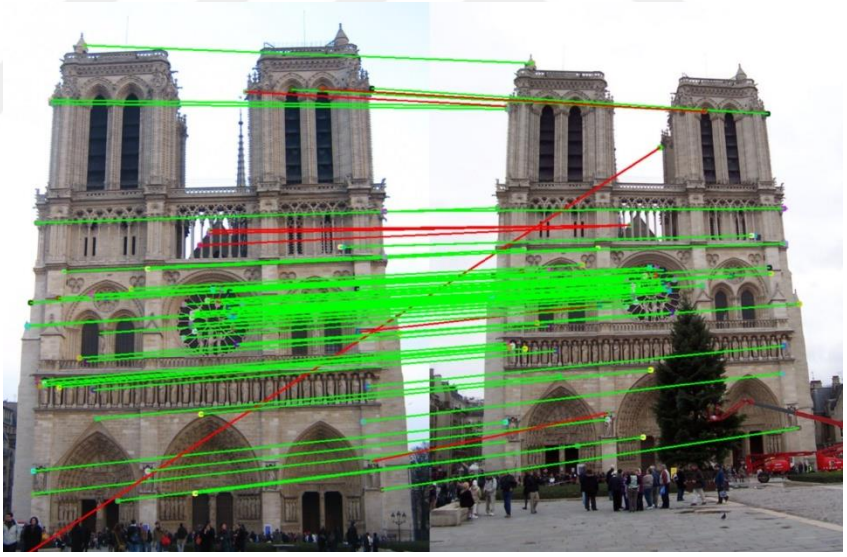

Figure 2.16. Paired Image Features

3. NEURAL NETWORK BASED VISUAL NAVIGATION

NNs have been used frequently in recent years to solve problems that were previously considered much more complex. One of these problems is positioning. The realization of visual aided navigation using NNs dates back to the 90s (Pomerleau, 1993). Pomerleau obtains an autonomous robot driver by training the system he calls ALVINN (Autonomous Land Vehicle in a Neural Network) with the visuals he obtains while driving. Thrun (Thrun, 1998) uses NNs for indoor robot positioning by teaching the network the sonar data it receives from the sensors. NN has also been used for mapping between satellite and aerial images (Chen, 2005) (J. R. G. Braga, 2016). Despite all the advantages of deep learning, it was difficult to process big data effectively until CNNs. By means of CNN, topographic maps were also included in the training (Tamre, 2009) (T. Wang, 2018). The complex nature of remote-sensing images and the distortions they contain make visual-based single-source positioning unsafe. In order to overcome this, some improvement methods have been developed that are integrated into CNN architectures. (Z. Yang, 2018) proposes a feature point registration procedure that uses a gradually expanding selection of inliers. (Costea, 2016) and (Wang, 2019) make more reliable positioning in the city using road maps.[28][29] predict the labels of the regions with the help of CNN. In (W. Ma, 2019), the descriptors obtained by SIFT are used in the CNN network. (H. Zhu, 2019) designs two strategies, namely, super pixel-based sample graded strategy (Sp-SGS) and super pixel-based ordered spatial matching (Sp-OSM) strategy to improve the matching accuracy. (A. Nassar, 2018) aims to increase positioning accuracy in settlements with the semantic shape algorithm. Mughal et al. (M. H. Mughal, 2021) whose open-source dataset is used in this study obtained an end-to-end positioning pipeline using CNN and (I. Rocco, 2018) NCN (Neighbourhood Consensus Network).

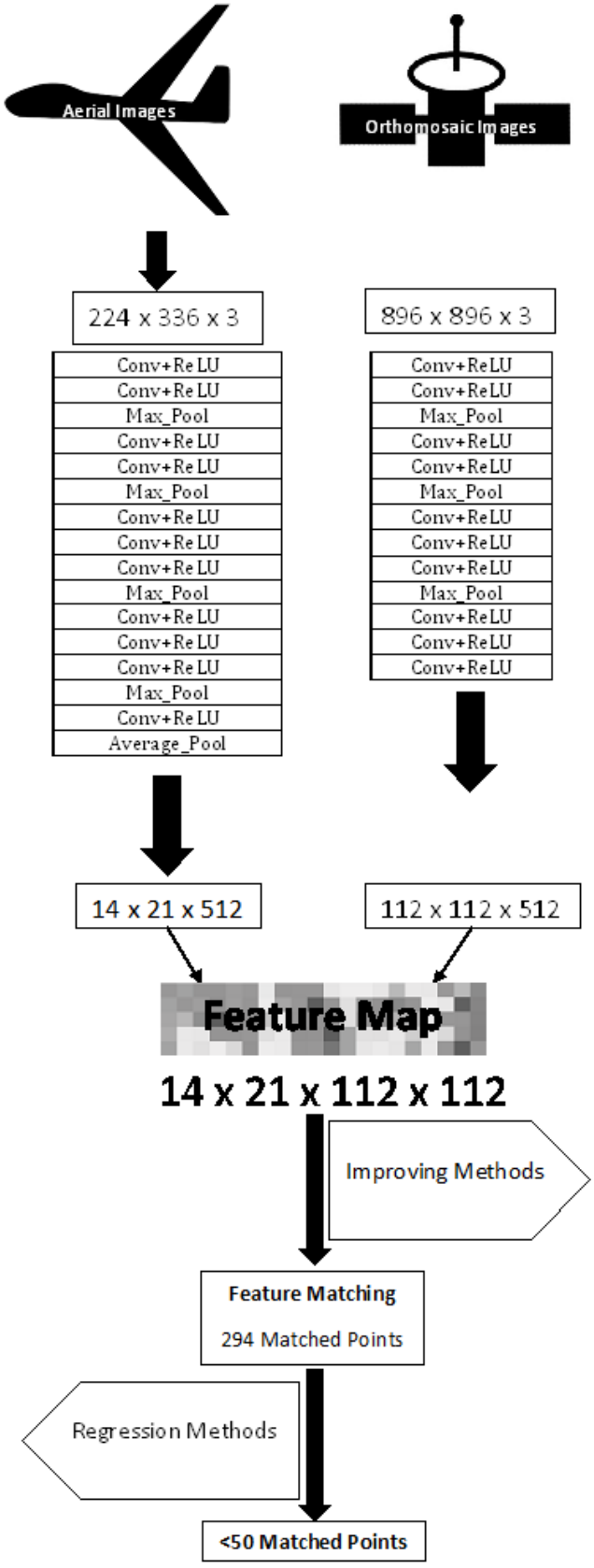

Figure 3. 1. CNN-Based Flow Chart

In this chapter, the visual navigation system, which is made using an end-to-end CNN network, and the flow chart is shown in Figure 3. 1. CNN-Based Flow Chart, is explained. The first step is to obtain the dataset and prepare it for matching. In the second step, features are extracted from the images and feature maps are obtained. In the next step, verisimilitude methods such as map, trajectory, manoeuvre are applied. These methods make improvements on the obtained feature maps by taking into account the flight characteristics of the UAV. This stage is the key to visual positioning. While it prevents illogical pairings, it greatly increases the matching accuracy. After the methods are applied, matching is done using refined maps. In the last step, clustering is applied to the matches made and the data to be used for final positioning is obtained.

3.1. Dataset Preparation

This section describes the pre-processing of images before feature extraction. As a first step, images are resized before they are sent to the network. The reason for this is to increase the processing speed. But there is a trade-off here. Distinctive information may be lost in images that are oversized to achieve greater processing speed. In Figure 3. 2., repeatability and scale factor graph are given for image matching. When non-adapted detectors are used, repeatability decreases as the scale factor increases _(Dufournaud, 2020). In the next step, the reduced images are normalized for better results. Thanks to normalization, duplications and anomalies in the dataset are removed. In the last step, scaling is done before sending it to the network. The purpose of scaling is to eliminate the scale difference between source and target images.

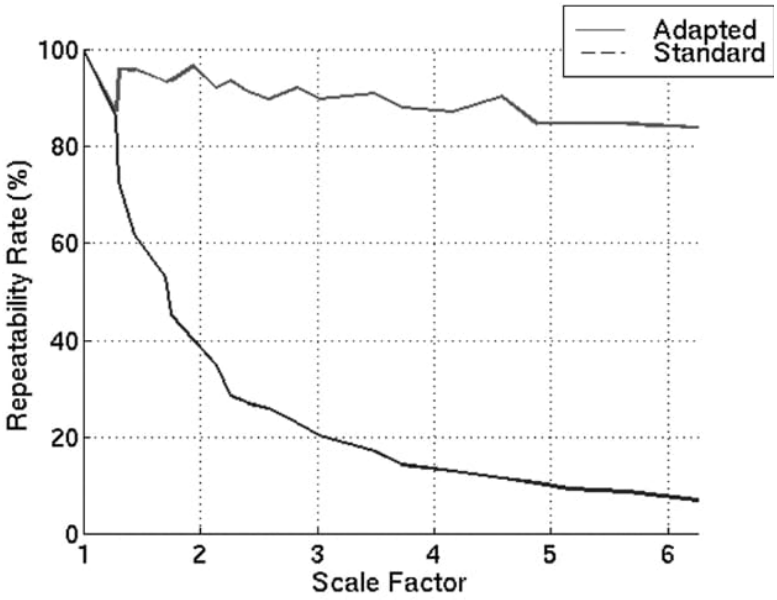

Figure 3. 2. The comparison criterion is the repeatability rate which is displayed as a function of the scale factor.

3.1.1. Dataset

In this study, it is a dataset belonging to the Nust region prepared for the study [34], in which the methods are applied. The reason for choosing this dataset is that each image has data to include ground-true location data. The UAV is also positioned by placing the source images, which are assumed to be instantly reflected on the camera of the UAV, into the target images. Figure 3. 3. shows an image from the dataset and the data it has. In the upper right, there is a sample source image and reference points are shown. In the target image used as a map, point equivalents are available for each reference point in the source. In this study, eight reference points were used in each source image.

Figure 3. 3. Ground True References

The dataset consists of 1200 images 580 square meters each and covers a total area of 522,000 square meters. The images were taken with the DIJI Phantom 4 Pro drone at different times of the day and combined to create a geo-tagged orthomosaic image. On the right is an image taken from the drone, while the geo-tagged image is on the left. For each of the 1200 images, the points distributed to the image are mutually available as given in Figure 3. 3. What makes this data critical is its use in detecting errors in the training of the CNN network. After matching, the ground-true data in the other image is used to obtain an error for the feature expressed by each point. With datasets that do not contain these data, network training should be done with weakly-supervised instead of strongly-supervised [35]. However, the feedback obtained in the calculation of the loss function made in this way will be quite ineffective.

3.1.2. Resizing

While the resolution of the source images in the dataset is 224x336, the target image has a resolution of 1896x1896. The resolution of the target image is reduced to 896x896 pixels, allowing the network to run 30 to 40 percent faster. Before the resize process is applied, an image dataset contains more than ten million data, while after resize it decreases to 2.5 million. The reason for choosing 896 pixels is to keep the greatest common divisor as large as possible. This gives us flexibility in convolutional operations. Resize operation is done with the affine grid generation method. After this process, the pictures are sent to the normalization step.

3.1.3. Normalization

By means of normalization, duplications and anomalies in the dataset are removed. When the parts with high pixel value density are not normalized, it causes erroneous matching. The reason for this is that the parts with anomalies in the mutual images are evaluated as similar by the algorithm even if they do not represent the same location. Figure 3. 4. shows a source image before and after normalization.

Figure 3. 4. Normalized Source Image

3.1.4. Scaling

Images that are sized and normalized are scaled before being sent to the network. The purpose of scaling is to eliminate the scale difference between source and target images. The scaling effect is shown in Figure 3. 5., which includes the reference structure. While the images are reduced gradually in the convolutional network, the number of pixels representing the distinctive elements as in the image decreases. Since a pixel-feature-based matching method is used, the closer the pixel count of the

items between the matched images, the more precise the matching will be. Since the source images in this study were taken from a fixed altitude, the manual scaling ratio was calculated once and applied for all. Where the altitude is variable, adaptive scaling will be required and this will be scrutinized in the discussion section.

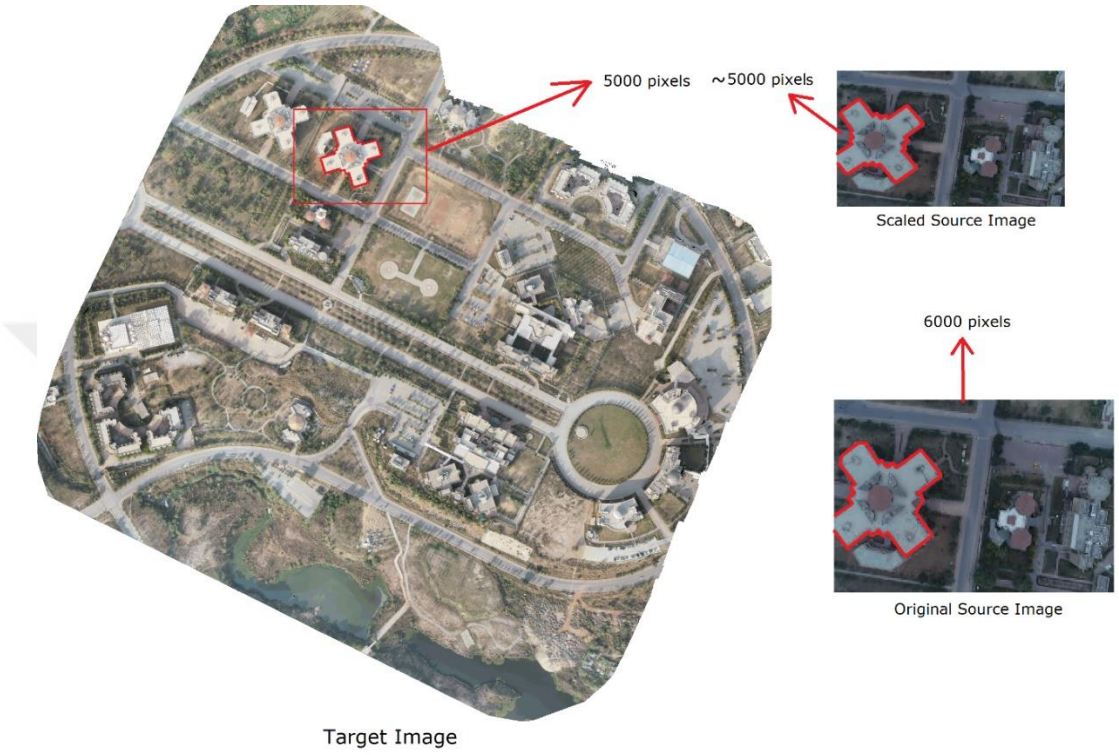

Figure 3. 5. Scaled Image

3.2. Feature Detection and Description

In this study, the detection and extraction of features was done using the trained VGG-16 network. Two separate pipelines are used for source and target images.

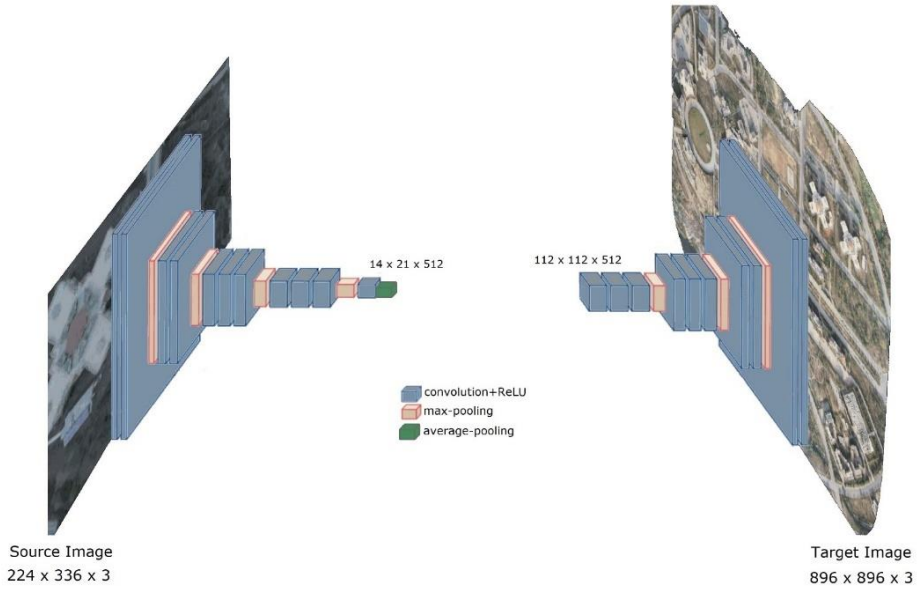

Figure 3. 6. CNN Architecture

The CNN architectures used in Figure 3.6.. are shown. While the source architecture has 16 layers, the target has 13 layers. Although the first layers are common, the methods followed are also the same. The first layers are the convolution layer where 3x3 kernels are applied. As shown in Table 3. 1., the inputs are 224 x 336 x 3 pixels by 896 x 896 x 3 pixels. The first two values represent width and height, while the third value is RGB (Red Green Blue). In the Convolutional layer, the RGB value is replaced by the number of filters. If it is wanted to be explained in a different way, each pixel is expressed with three different color values, while after the first step it is expressed with filters. The first layer contains the conv-section and RELU (Rectified Linear Unit). The picture accesses 64 filters by means of the conv part and an activation function ReLU is applied. In the second conv-layer, the number of parameters increases while the number of filters remains the same. In the third layer, the max-pooling layer, the image is scaled in half. This process is done three more times for the source image and two more times for the target image. Finally, the source image is inserted into the average-pooling layer. Data in the size of 14 x 21 x 512 for source and 112 x 112 x 512 for target are obtained from the VGG-16 network. In the obtained data, there are 512 filter results for each of the pixels for

both images. A four-dimensional feature map is obtained from these two three-dimensional datasets. The resulting map can be called the correlation matrix.

Table 3. 1. Convolutional Neural Network Dimensions

#	Source Image			Target Image			Layer	Stride		Kernel		Padding	
0	224	336	3	896	896	3							
1	224	336	64	896	896	64	conv3-64	1	1	3	3	1	1
2	224	336	64	896	896	64	ReLU	-	-	-	-	-	-
3	224	336	64	896	896	64	conv3-64	1	1	3	3	1	1
4	224	336	64	896	896	64	ReLU	-	-	-	-	-	-
5	112	168	64	448	448	64	maxpool	2		2	2	0	
6	112	168	128	448	448	128	conv3-64	1	1	3	3	1	1
7	112	168	128	448	448	128	ReLU	-	-	-	-	-	-
8	112	168	128	448	448	128	conv3-64	1	1	3	3	1	1
9	112	168	128	448	448	128	ReLU	-	-	-	-	-	-
10	56	84	128	224	224	128	maxpool	2		2	2	0	
11	56	84	256	224	224	256	conv3-64	1	1	3	3	1	1
12	56	84	256	224	224	256	ReLU	-	-	-	-	-	-
13	56	84	256	224	224	256	conv3-64	1	1	3	3	1	1
14	56	84	256	224	224	256	ReLU	-	-	-	-	-	-
15	56	84	256	224	224	256	conv3-64	1	1	3	3	1	1
16	56	84	256	224	224	256	ReLU	-	-	-	-	-	-
17	28	42	256	112	112	256	maxpool	2		2	2	0	
18	28	42	512	112	112	512	conv3-64	1	1	3	3	1	1
19	28	42	512	112	112	512	ReLU	-	-	-	-	-	-
20	28	42	512	112	112	512	conv3-64	1	1	3	3	1	1
21	28	42	512	112	112	512	ReLU	-	-	-	-	-	-
22	28	42	512	112	112	512	conv3-64	1	1	3	3	1	1
23	28	42	512	112	112	512	ReLU	-	-	-	-	-	-
24	14	21	512	-	-	-	maxpool	2		2	2	0	
25	14	21	512	-	-	-	conv3-64	1	1	3	3	1	1
26	14	21	512	-	-	-	ReLU	-	-	-	-	-	-
27	14	21	512	-	-	-	avgpool	1	1	2	2	1	1

This matrix gives the similarity ratio for each pixel in the target image to the pixels in the source image, as in Figure 3. 7. Each of the 294 pixels in the source image contains a value corresponding to each of the 12544 pixels in the target image. These values give the similarity ratio. The higher the value, the greater the similarity. This is because during matrix multiplication, each filter is multiplied by the corresponding filter in the counter matrix. If the pixels contain similar patterns, the value of the mutually identical filters will be higher than the others. However, before pairing, these values should be adjusted according to the flight characteristics of the UAV using development methods.

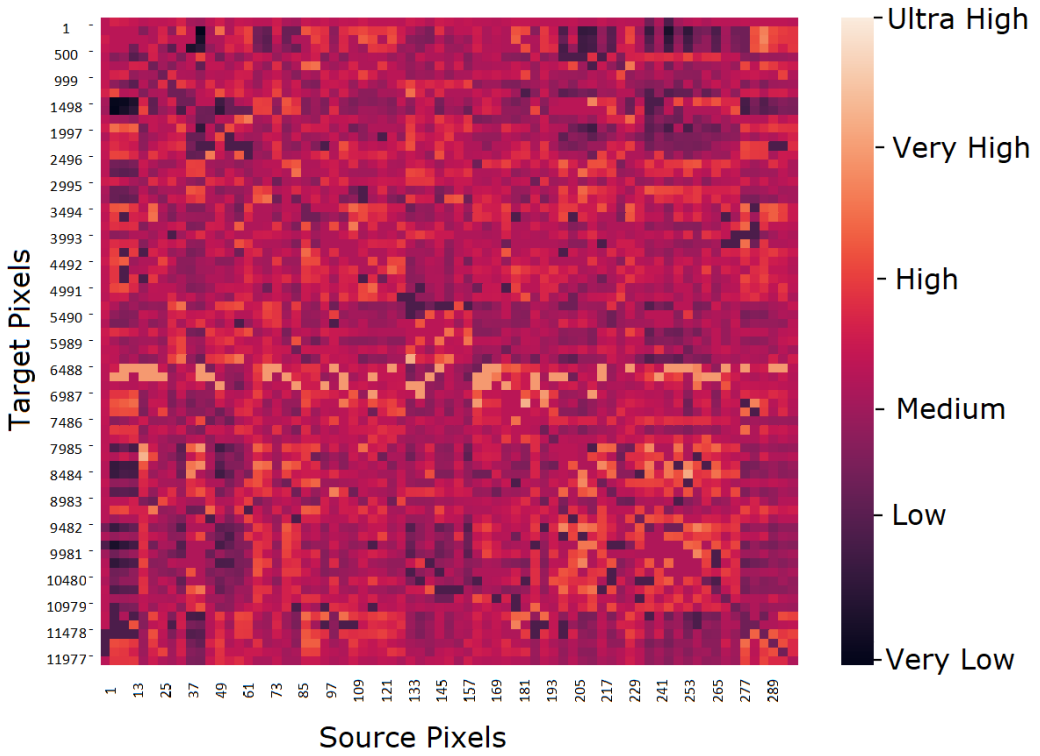

Figure 3. 7. Correlation Map

3.3. Improving Methods

When the resulting feature map is used for matching without any processing, it will lack a presupposition about the nature of positioning. This may lead to inconsistent results and reduce the efficiency of the algorithm. To overcome this problem, _(A. Nassar, 2018) increase the accuracy of geolocalization using the semantic shape matching (SSM) method. Ma et al. reduce the registration error with the location adjustment method in _(W. Ma, 2019). In this study, three different methods, namely map, trajectory, and manoeuvre verisimilitude, are applied respectively to overcome this situation. The purpose of applying the methods is to reduce unsuitable mappings for the flight characteristics of the UAV and to take into account the result of previous estimates in each new position estimate to be made.

3.3.1. Map Verisimilitude Method

This method includes the presuppositions necessary due to the nature of positioning into the pipeline. It creates a prediction for the new predictions that the UAV will obtain using its last location. To do this, apply the MPR (Map Verisimilitude Rate)

formula given in 4.1 to each pixel in the feature map. The distance is expressed by d and its unit is pixel. fs^{-1} is the number of frames the system matches per second. The given ratio decreases with distance from the estimated final position. As the data processing capacity of the system increases, the matching frequency will increase and the ratio will decrease. As the speed (v) of the platform increases, it is inversely proportional to the MPR as more pixels will be covered per unit time.

$$\text{MPR} = \frac{1}{1+\frac{d\, fs^{-1}}{v}} \tag{3.1.}$$

According to this formula, as in Figure 3. 8., as the distance of the pixels to the final location increases, their values in the map will decrease for the next matching. This will reduce the matching probability of these pixels in the matching step.

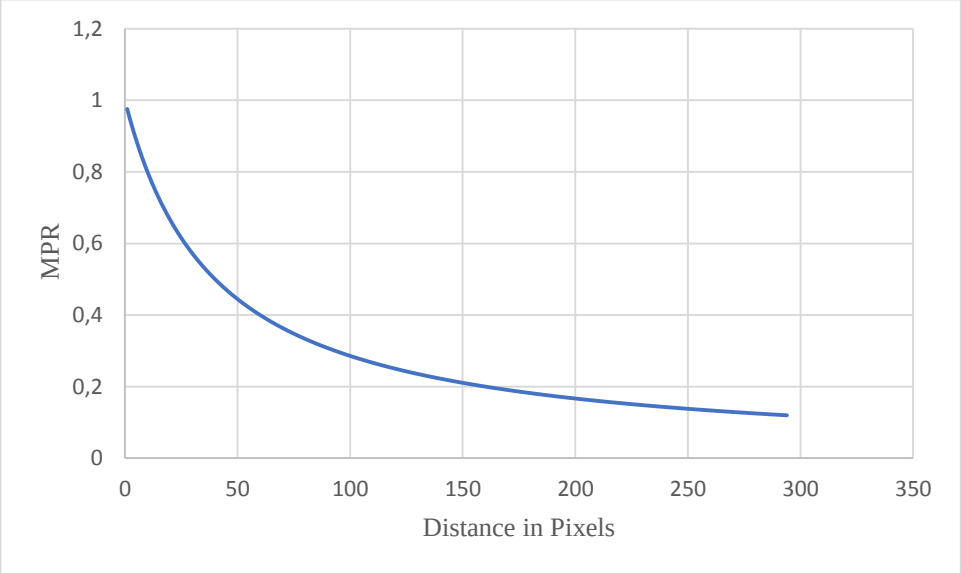

Figure 3. 8. Map Verisimilitude Rate

In Figure 3. 9., MPR maps between two UAVs with cruise speeds of 0.42 and 0.16 Mach are visualized. As the colour gets darker, the probability of being at the point given by the algorithm decreases. The MPR values of the aircraft with the highest

maximum cruise speed are higher than the other. We can attribute this to the fact that it is more likely to be located at farther points per unit time.

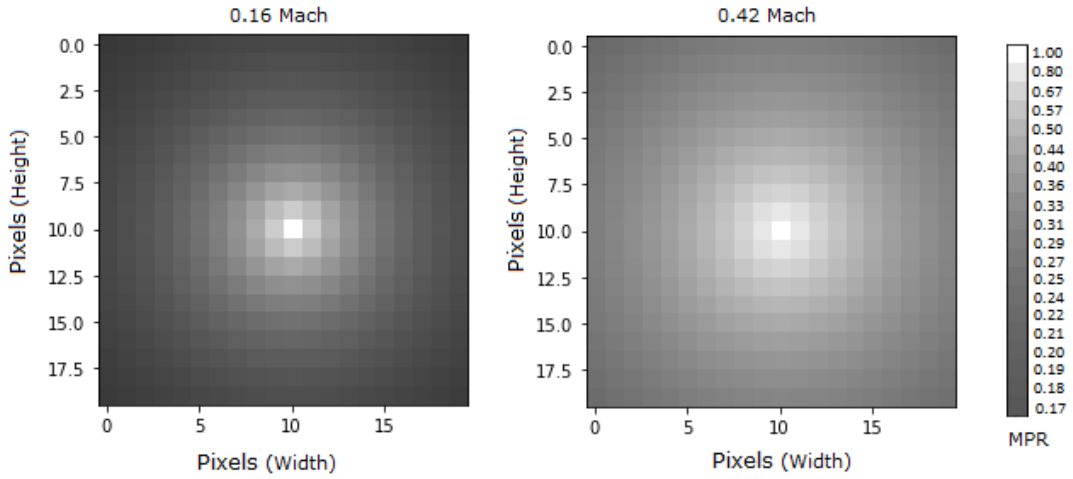

Figure 3. 9. MPR Maps for the UAVs Flies at 0.16 Mach and 0.42 Mach

3.3.2. Trajectory Verisimilitude Method

In the Trajectory method, an algorithm that keeps the flight path of the aircraft in its memory is used. It evaluates the flight space in two dimensions and divides it into eight parts of 45 degrees. As the platform moves in the separated directions, it increases the coefficient of that direction and the directions around this part. As this coefficient increases, the matching coefficient of the pixels in that direction is also increased. This means that there will be more matches in this direction. Since the aircraft will not follow a straight flight line, this coefficient increase is limited enough not to create a very stable state that is effective enough to eliminate the instability. Figure 3. 10. shows the given flight path and the coefficients given to the surrounding pixels at certain intervals.

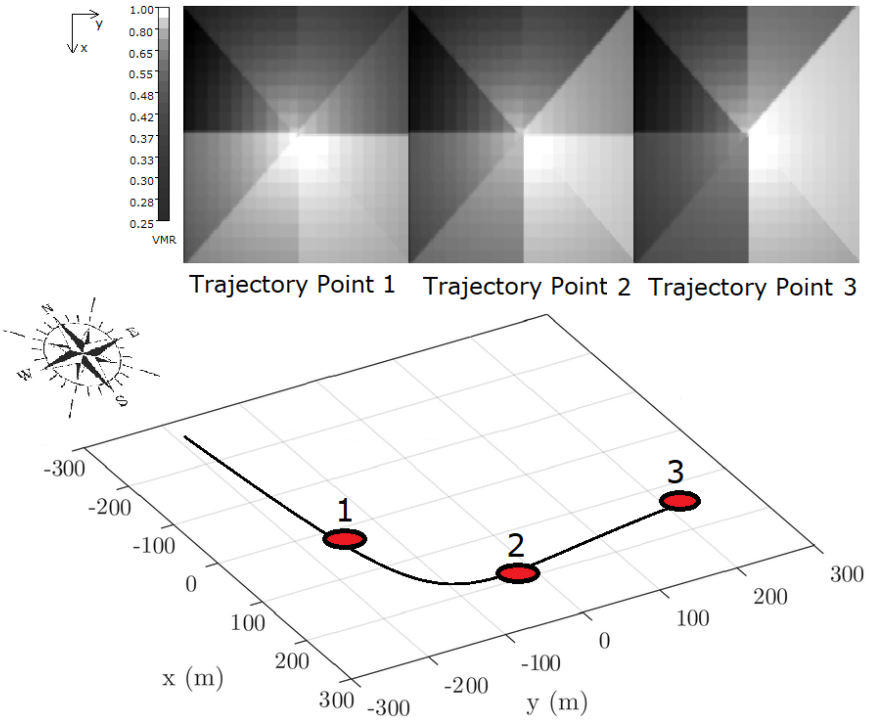

Figure 3. 10. Trajectory Verisimilitude Method

The main purpose is to reduce the probability of matching in pixels located in the opposite direction of flight. However, looking at the movements per unit time in order to evaluate the flight direction will not give healthy results. Even if a direction prediction can be made when examining short sections, it will not be reliable enough to affect future positioning predictions. For these reasons, this method does not come into play for each forecast, but only when successive forecasts indicate a stable trend. The coefficients are limited when traveling in a certain direction for a long time. Otherwise, the subsequent positioning estimates simulated in the algorithm will be too stable. This may cause the system to react later or even not respond to the movements of the aircraft. The constrained coefficients are gradually reduced as the UAV continues to move in another direction.

3.3.3. Manoeuvre Verisimilitude Method

Although the speed of the aircraft is included in the loop in the Map method, the flight characteristics do not have a full effect yet. More precisely, by using the manoeuvring capabilities of the aircraft, the potential matching area is narrowed and thus a higher accuracy positioning is achieved. The Manoeuvre method is applied to the pixels in the area where the platform is located, not to each pixel as in the map. An imaginary corridor is drawn that narrows during flight. The rate of turn (ROT) value and speed of the aircraft are used when calculating the limits. Boundaries having an adaptive structure decrease the aisle angle as the aircraft speed increases. Thus, the values on the feature map of the pixels within the areas scanned by the narrowing corridor are increased and the matching accuracy is increased. In this study, the ROT value, which is the flight characteristic of the aircraft, is included in the algorithm for once and the movements of the flight control surfaces are not observed instantly. This situation will be scrutinized in the discussion section.

3.4. Image Matching

After the feature map acquires the flight and positioning characteristics, it is passed to image matching, which is the final stage for visual-based localization. The main problems encountered in the point or pixel-based image matching methods used in this study are outliers and unstable matching. Outliers are the result of mismatches and if not corrected, they will lead to bias in the final data, especially for positioning. Unstable matches, on the other hand, refer to pixels whose matching results are equal or very close. In Figure 3. 11., contradictory and unstable situations are visualized. Regions one, two and four refer to the feature points in the source image and the others in the target image. Matching 8 and 9 is incorrect for 2, while pairing 4 and 6 is unstable. Similarly, pixels 5 and 7 are unstable matches for pixels 1 and 3. In this study, Dbscan, a clustering method, is used as a solution to the outlier problem, and SoftMax and ArgMax methods are used as a solution to the instability problem.

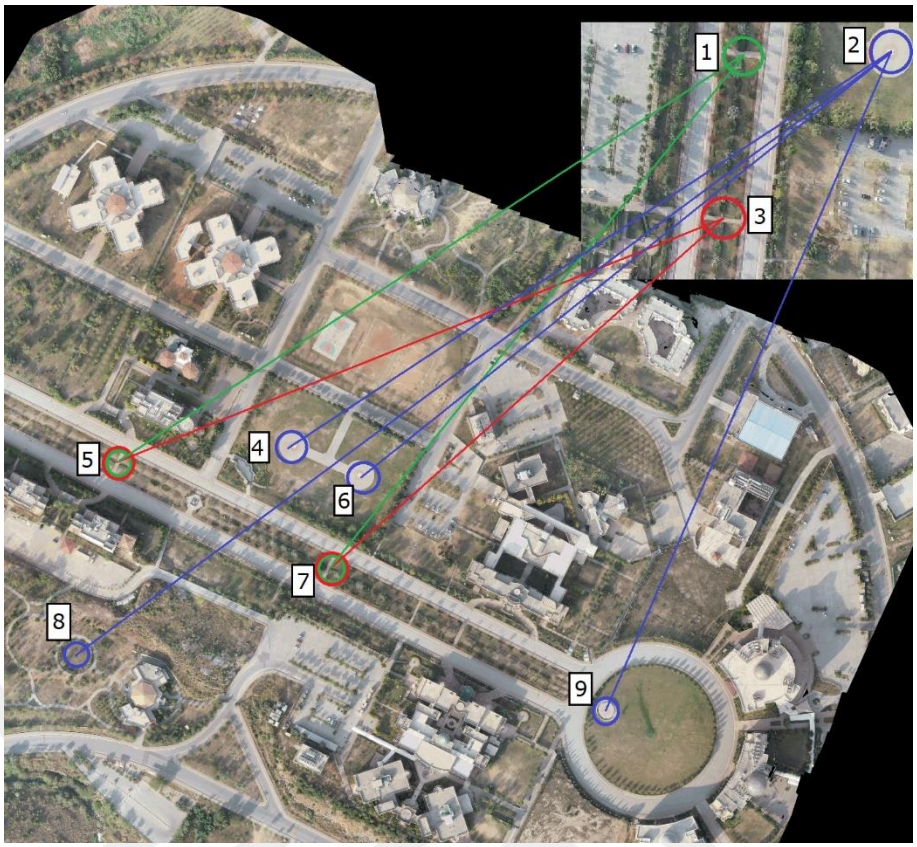

Figure 3. 11. Unprioritized Match Points

3.4.1. SoftMax

SoftMax is a function used to convert numerically expressed values into probabilities. A simplified feature map and inter-pixel matching coefficients are given in Figure 3. 12. Each four squares represent a pixel in the source image. Each of these gives the ratio of similarity to the pixel in the target image. For example, the similarity ratio of the first pixel in the source image and the third pixel in the target is 0.4.

(1,1)	(1,2)	(2,1)	(2,2)
0.2	0.3	0.5	0.3
(1,3)	(1,4)	(2,3)	(2,4)
0.4	0.5	0.2	0.4
(3,1)	(3,2)	(4,1)	(4,2)
0.3	0.4	0.1	0.1
(3,3)	(3,4)	(4,3)	(4,4)
0.3	0.5	0.2	0.5

Figure 3. 12. A Simplified Correlation Matrix Before SoftMax

According to this data without SoftMax applied, it is seen that the 1st, 3rd and 4th source pixels match the 4th pixel of the target image. Similarly, the 4th pixel of the source image gives equivalent similarity rates for the 1st and 2nd pixels of the target image. If the match was made in this state, the result would not be suitable for use as it seems. This is for two reasons. The first reason is that some of the target pixels that the pixels in the source image give the highest similarity are common. Although it is not possible in practice, this situation is caused by the 'multi-pattern' feature of some pixels. It is a kind of summary of the pattern of the general map, as it contains a lot of color and shape transitions within these pixels. These features increase the probability of matching any pixel. The second reason can be defined as the same situation being present in the image used as the source. This happens when a pixel in the source image gives values equal to or very close to two or more of the target pixels.

(1,1)	(1,2)	(2,1)	(2,2)
0.229	0.254	0.309	0.254
(1,3)	(1,4)	(2,3)	(2,4)
0.282	0.256	0.231	0.231
(3,1)	(3,2)	(4,1)	(4,2)
0.253	0.281	0.207	0.208
(3,3)	(3,4)	(4,3)	(4,4)
0.255	0.256	0.231	0.256

Figure 3. 13. A Simplified Correlation Matrix After SoftMax

For each pixel in the target image, the source pixels are inserted into the SoftMax function. After this process, the values are refined as shown in Figure 3. 13. Thus, only one source pixel for each target pixel gives the highest similarity rate in matching.

3.4.2. ArgMax

ArgMax is a function that allows us to obtain the index of the one with the maximum value in the data set in which it is used. After SoftMax is applied, the highest probability values are obtained with ArgMax. After this process, a matrix of size (294 x 2) is obtained. The resulting matrix gives the position of the pixel to which each source image pixel is mapped on the target. Although each pixel matches the one with the most similarity in the opposite image, not all of them have the same similarity ratio. By means of ArgMax, they are ranked with the highest of these ratios in the first place. Although even one point is sufficient for matching, it is necessary to get rid of incorrect matches in order to determine the most accurate location. Clustering methods are used for this.

3.4.3. Clustering

In this study, an unsupervised Dbscan method is used as a clustering method. Dbscan, a density-based spatial method, divides data into two as inlier and outlier.

3.4.4. Prioritisation

After the Dbscan method, less than 294 inlier matches are obtained. Only the first 20 of these are used to obtain positions. There is a trade-off here. Using less points than necessary may cause an incorrect positioning, while using more than necessary will reduce the sensitivity at the detected location. For these reasons, a number that is predicted to include the data in the entire dataset was chosen. Here, a trained network is needed to use different number of matching points in different situation.

4. HANDCRAFTED BASED VISUAL NAVIGATION

In computer vision navigation, Lowe‘s SIFT, a handcrafted method is used as the second image processing source in the system. (Ma, 2021) SIFT extracts key-point as the local extrema in a DoG (Difference of Gaussian) pyramid, filtered using the Hessian matrix of the local intensity values. DoG is a Feature enhancement algorithm. The image is progressively blurred with Gaussian kernels, a blur operator.

$$L(x, y, \sigma) = G(x, y, \sigma)I(x, y) \tag{4.1}$$

I and G represents the image and gaussian operator, respectively. The O parameter determines the blur ratio, and the larger it is, the higher the degree of blurring. DoG is obtained by comparing images blurred at different degrees. Local extremes are determined as keypoints. Because of DoG has a high response for edges they need to be removed. After testing the stability of the keypoints whose scale of detection is known, the next step is the rotation step. At this stage, the gradient size and direction of the neighbouring pixels around the point are calculated. The peak regions in the histogram taken after the calculation are used to calculate the orientation. Since a unique identifier is ultimately required, vectors are defined as in Figure 4. 1.

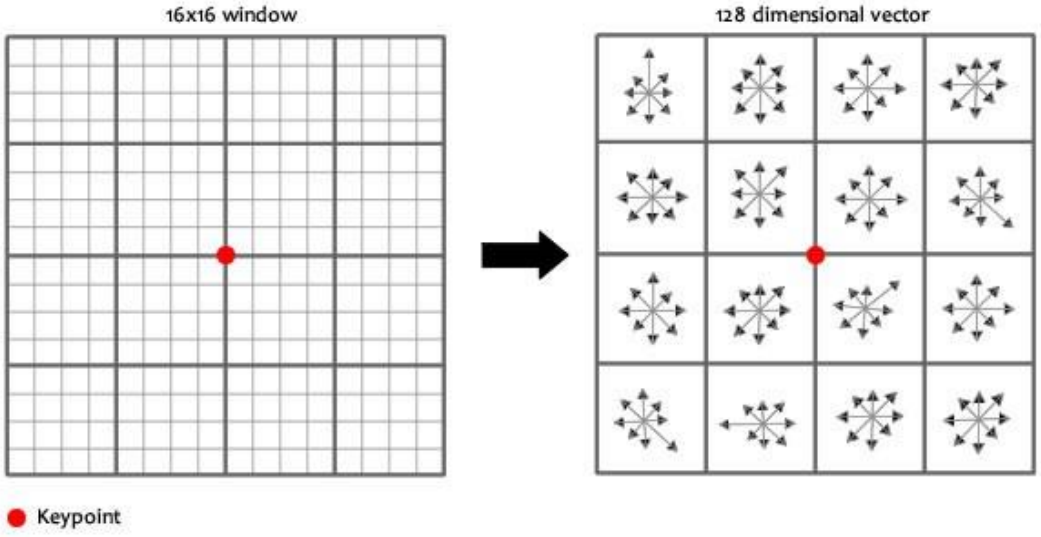

Figure 4. 1.Vectors Around Keypoint

Eight histograms are created for each sub-block. Before the last step where fingerprint is defined, some features need to be added to the features. These features are making the feature independent from rotation and illumination. After this process is done, the matching phase is started. At this stage, the keypoints of the pictures are matched according to their similarity ratios. But sometimes the second most similar match is very close to the first. There may be reasons for this such as noise and reflection in the image. In this case, if the closeness ratio of the first two closest matches is greater than 0.8, they are rejected. Ninety percent of incorrect matches are eliminated with this process, while only five percent of successful matches are lost.

5. ADAPTIVE BLENDING

At this stage, which is the state of art of this study, the adaptive blending of two systems that are dependent in terms of resources and independent in terms of positioning techniques is described. Adaptive blending is re-weighting two separate positioning sources at each iteration in order to be able to position with the highest accuracy, taking into account certain conditions. In order to do this, first of all, it should be determined to which conditions the positioning sources are sensitive and attributes that can be measured with high precision should be selected. Then, the error rates of the systems should be calculated under these conditions. In order to obtain error rates, different flight routes and flight conditions should be simulated and the systems should be tested in these scenarios. A data set is created with the extracted error rates and changing conditions. The generated datasets are trained with appropriate neural networks. The reason for choosing NNs here is that there are many parameters and the relationship between them is complicated. The trained networks will then weight resources to implement adaptive positioning using it in real-time flight. In this way, it is expected that the adaptive blended system will give more accurate results in positioning compared to the sources weighted with predefined fixed degrees.

5.1. Dataset Preparation

The first step when creating a dataset is to determine the attributes. Some of them are listed as in Table 5. 1. Among these labels, the ones that affect the positioning accuracy of the systems should be selected. Otherwise, it will cause unnecessary workload for the algorithm and inefficiency in the training phase. At the same time, the specified parameters should be easily identifiable. It is not possible to use parameters that cannot be determined with the available sensors. In addition, it will not be efficient to select attributes that are difficult to determine and analyze, such as "Control Surface Commands". This issue will be detailed in the discussion section. Another criterion is the sensitivity of the systems to the factors. If the positioning

error of the system is independent of the variable, it will not make sense to use it in blending. The selected variables should both affect the error rate of the systems and be possible to detect. Three different times of the day are simulated in the dataset. To achieve this, the brightness of the source images is gradually changed. The efficiency of computer vision methods decreases in a very dark and bright environment. This is the most vulnerable part of these systems. "Mistiness" is the amount of fog. The efficiency of visual positioning systems in foggy environments decreases as in dark environments. In foggy, cloudy and snowy environments, the efficiency of visual positioning systems decreases as in dark environments. The reason for this is that the ground visibility decreases, as well as the decrease in the quality of vision. Decreased visual quality reduces detail in images and approaches or even falls below the minimum threshold required for matching. As a result, incorrect positioning becomes inevitable. Another criterion is terrain characteristic. This parameter expresses the unique features of the sections corresponding to the route flown. The main factor that determines these features is the color distribution in the images. The features to be deduced by looking at the colors in the images can give information about the vegetation of the region and the amount of urbanization. At the same time, since the responses of the two systems to color diversity differed, it was included in the data set. The selected criteria play a decisive role for both systems and are determinable.

Table 5. 1. Positioning Systems and Criteria

	Availability	Degree of Dependency (NN-Based)	Degree of Dependency (Handcrafted-Based)
Quantity of Light	Yes	High	High
Mistiness	Yes	High	High
Degree of Snowy Vision	Yes	High	High
Terrain Characteristic	Yes	High	High
UAV Speed	Yes	Low	Low
Elevation Changes	No	Very Low	Very Low
Vibration Severity	No	Medium	Medium
Trajectory Characteristic	Yes	Medium	Very Low
Instantaneous Noise or Sparkle	No	High	High
Control Surface Commands	Yes (Hard)	High	High

In addition, the platform speed and trajectory characteristic given in the table were included in the simulation and the effects of the attributes were observed by analyzing the dataset that created the outputs. The Figure 5. 1. shows the feature importance of attributes obtained using gradient boosting. Considering these values, it is seen that the effects on the positioning of the systems are low and they are in line with the expected effects before the analysis. According to the analysis made with 5 thousand data, the selected attributes are “Quantity of Light”, “Mistiness”, “Degree of Snowy Vision” and “Terrain Characteristic”. Another advantage of the selected criteria is that they can be extracted from the currently studied images.

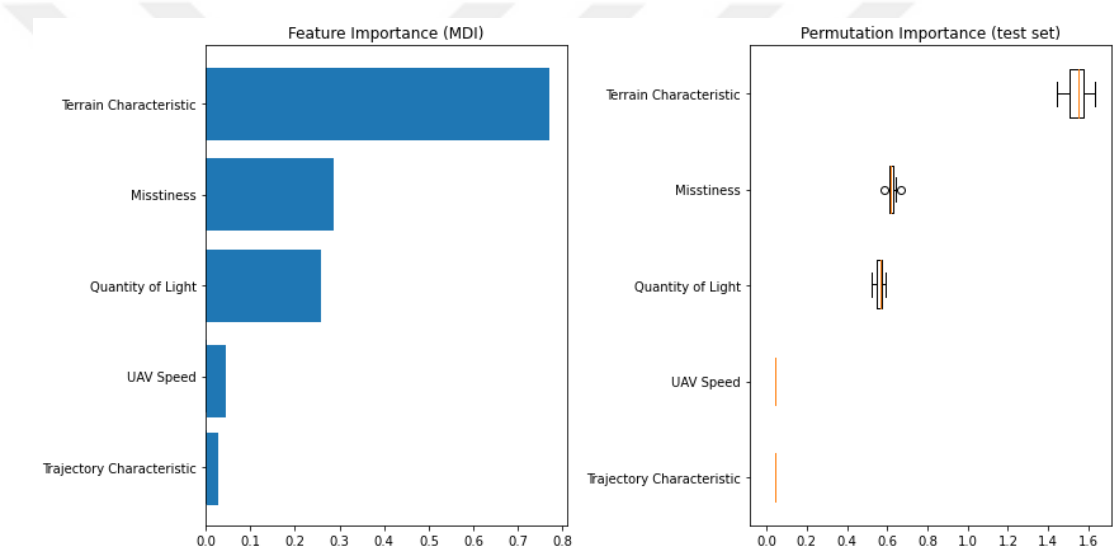

Figure 5. 1. Feature and Permutation Importance

5.1.1. Simulation Model

It is necessary to obtain error amounts by testing two separate systems with the specified qualities. For this, a model must be created to include all combinations of conditions. For example, a foggy evening and a sunny morning should be simulated. Three different brightness levels have been determined for the time of day that will form the light amount label. The evening time scenario was imitated by using the image processing method, since there are not enough low-light images to meet the light levels determined in the existing data set. Similarly, the images are blurred to create a foggy image. Blurred pictures cause the same effect as the reduction of

details in the pictures when the field of view decreases. In order to create the snowy land, the image processing method was used as in Figure 5. 2.

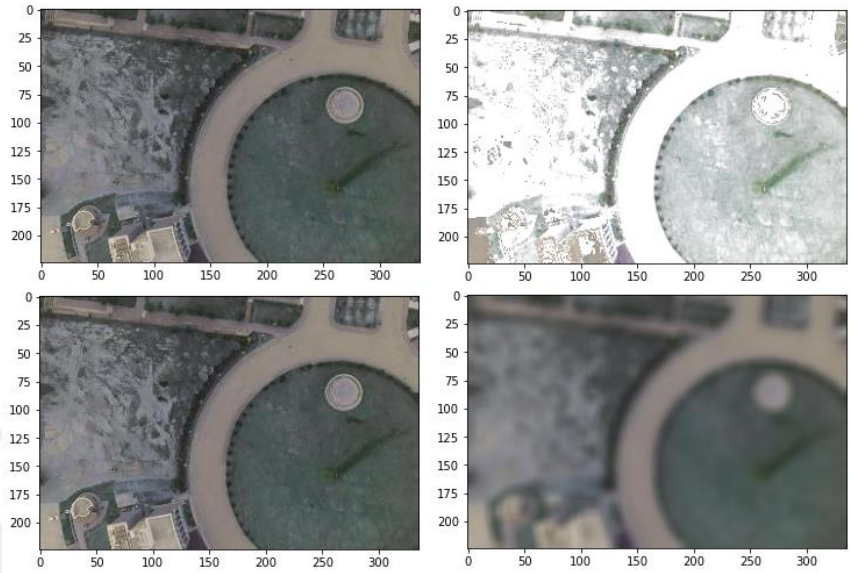

Figure 5. 2. Source Images with Fog and Snow Effect Applied

The three attributes described so far can be combined in a total of 27 different ways. The previously mentioned terrain label will be determined by the features to be obtained from the source images instead of being created artificially. To obtain this attribute, the contrast values in each picture were used. The next step is to create the flight patterns. For NN-based and HM-based systems, this is done differently. When testing the HM-based positioning system, 20960 images are given at once and the outputs are not split. However, while testing the NN-based system, first of all, 1048 different flight patterns consisting of 20 frames are created. Although the tested pictures and their order are the same, the reason why it is divided in this way is because the Network-based system has continuity. With the effect of the development methods described earlier, the predictions that the system will make in the next step are affected by ones made in previous step. Therefore, at the end of each route, the parameters from the development methods are reset and the test is performed. Some simulated routes are shown in Figure 5. 3.

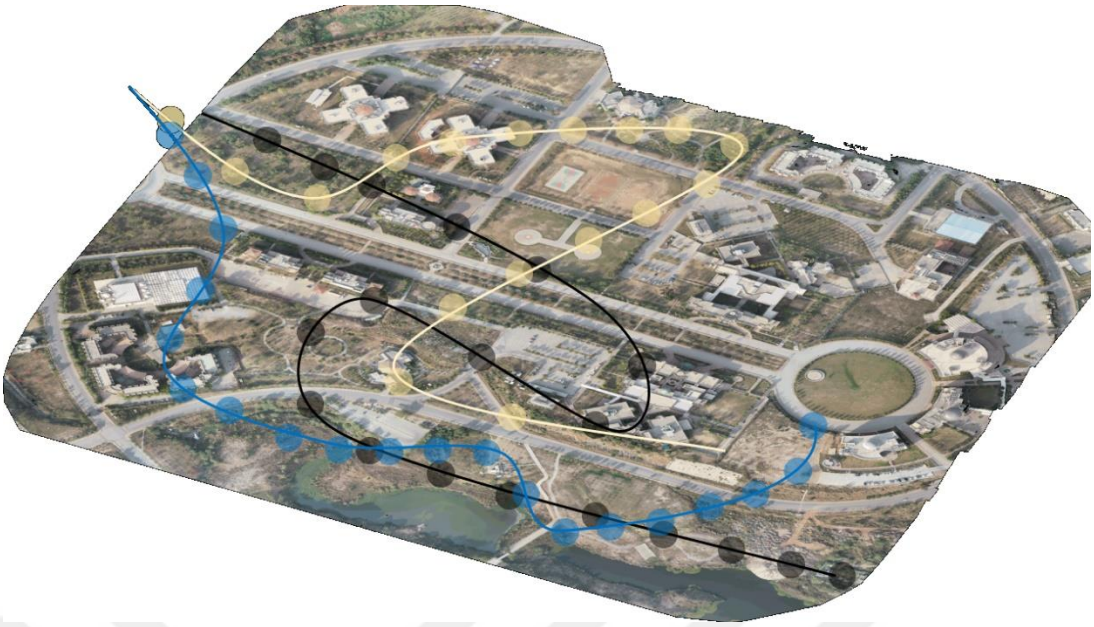

Figure 5. 3. Some Flight Patterns

5.1.2. Data Acquisition

To create the data set, each of 1048 patterns consisting of 20 points is combined with 27 different conditions. Thus, a data set with the size of "[565.920 x 6]" is obtained. The first four columns are the previously selected attributes, the fifth and sixth are the positioning errors of the systems. These errors are the distance from the ground-true values of the point determined by the algorithm on the target image as a result of the matches. The distance is calculated with the latitude and longitude values corresponding to the location marked in the picture. The data set containing the error of the NN-based system is sent to the LSTM network for training, and the HM-based part to the ANN.

5.2. Training Process

The resulting dataset includes positioning errors of CNN-based and HM-based positioning systems under variable conditions. If we want to make an adaptive weighting, we first need to know which system will give more accurate positioning results in real-time observed conditions. To obtain this information, we need to train the network with the dataset containing the test results. To do this, two separate NN architectures were created for the two systems. The data of the CNN-based system is trained with an RNN type, LSTM. The reason for training with LSTM is the ability

of the architecture to learn sequential and time-dependent data. Here, what is meant by time dependent data is that each unit of data is affected by the previous and the next. The reason for such an interaction between CNN-based positioning data is due to enhancement methods. Some of the development methods have their own memory. By means of this memory, it makes a prediction for the others by using each positioning result. For all these reasons, there is a memory dependency in each batch of the dataset obtained by CNN-based positioning. This caused us to benefit from LSTM when choosing the network to be trained with these data. On the other hand, the simpler ANN is preferred for the HM-based system where there is no interaction between the unit positioning data.

5.2.1. LSTM

In order to train the network, first of all, the data must be prepared for training and the appropriate architecture must be set up for the data structure. The dataset, consisting of 1048 different patterns and expressing 27 different conditions for each, contains a total of 28,296 matches, each consisting of 20 steps. While training time-dependent data in the LSTM network, the total unit time is determined and the data is prepared accordingly. Here, our time step value is 20 since each pattern consists of 20 steps. When training the network, we need to make sure that the time dependent data maintains the same order. Therefore, we can express the three-dimensional data set as in Figure 5. 4. The first value represents the number of combinations required for the test. As mentioned earlier, there are three attributes, each simulated for three different situations. For example, for the "quantity of light" feature, the amount of light in the morning, noon and evening hours is given. In short, these three variants combine a total of 27 different states for each of the 1048 flight routes.

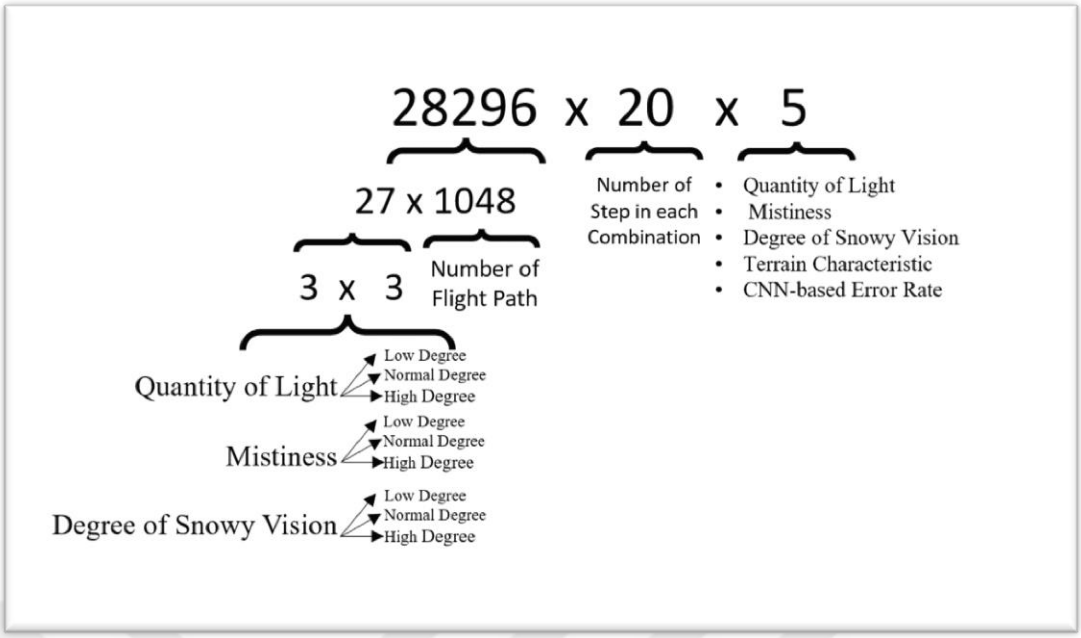

Figure 5. 4. Dataset for LSTM Training

This results in a total of 28,296 states. The second value is time step, while the last value represents the data type. More specifically, the first three of the five data represent simulated current features. The fourth is the data expressing the terrain characteristic mentioned earlier. The final value is the amount of error in meters of the CNN-based system obtained at each match. For network training, data should be separated as train, test and validation. Although there is no standard method for this, in this study, 60 percent of the entire dataset is reserved for training, 30 percent for testing and the rest for validation. Since numbers with multiples of one hundred would be more useful when training the network after separation, the data numbers were rounded to the nearest hundred, so as to delete some of the data. In addition, the first four of the data in the previously mentioned third dimension of the dataset are reserved as input and the last data, the error amount, is reserved as output. The sizes of the data obtained after all these processes are shown in Table 5. 2.

Table 5. 2. Dataset for LSTM Training

	Train Data	Test Data	Validation Data
Input	16,900x20x4	8,400x20x4	2,800x20x4
Output	16,900x20x1	8,400x20x1	2,800x20x1

In addition, since the distribution of the data is high, especially due to the amount of error data, the training is also normalized before being sent to the network to increase efficiency.

After the dataset is prepared, the most important step is to prepare the architecture according to the data and ultimately the output we want to achieve. At this stage, we encountered a challenge related to the structure of networks with temporal perception. In the LSTM networks used in the literature, the first time step is used with the same value in the post-training prediction phase. To explain in more detail, these time-dependent relationships are taught at a certain time interval during the training phase of the network, and when estimating with the trained network, data is given at this time interval and output is obtained. However, it would be useless to establish such an architecture in this study. The reason for this is that we want to give data consisting of 20 units of time as input while training the network, and therefore it is trained in this time interval. But when we want to make a real-time forecast, we should be getting this forecast in any unit of time. For this, we need to train 20 different networks for each of the 20 time slots and use the networks in order in the real-time positioning test, which would be pretty pointless. As a solution to this problem, instead of training many networks, the parameters of a trained network are cloned and used for predictions at different time intervals. To put it more clearly, while an architecture with a timestep of 20 was established for the training phase, estimations were made at different time intervals by using the weights of this architecture during the estimation phase.

The built architecture is shown in Figure 5. 5. The architecture consists of four layers. The first layer is the LSTM layer, which is the input layer. It increases the input size (20,4) to (20,64) by means of the 64 neurons it contains. The second layer is a LSTM layer with 128 neurons, while the last two layers are NN layers with 256 and 1 neuron, respectively. In addition, dropout operation is performed between these layers as in the image. By doing this, randomly selected nodes are left at each learning stage. Thus, the mesh is prevented from being over-fit.

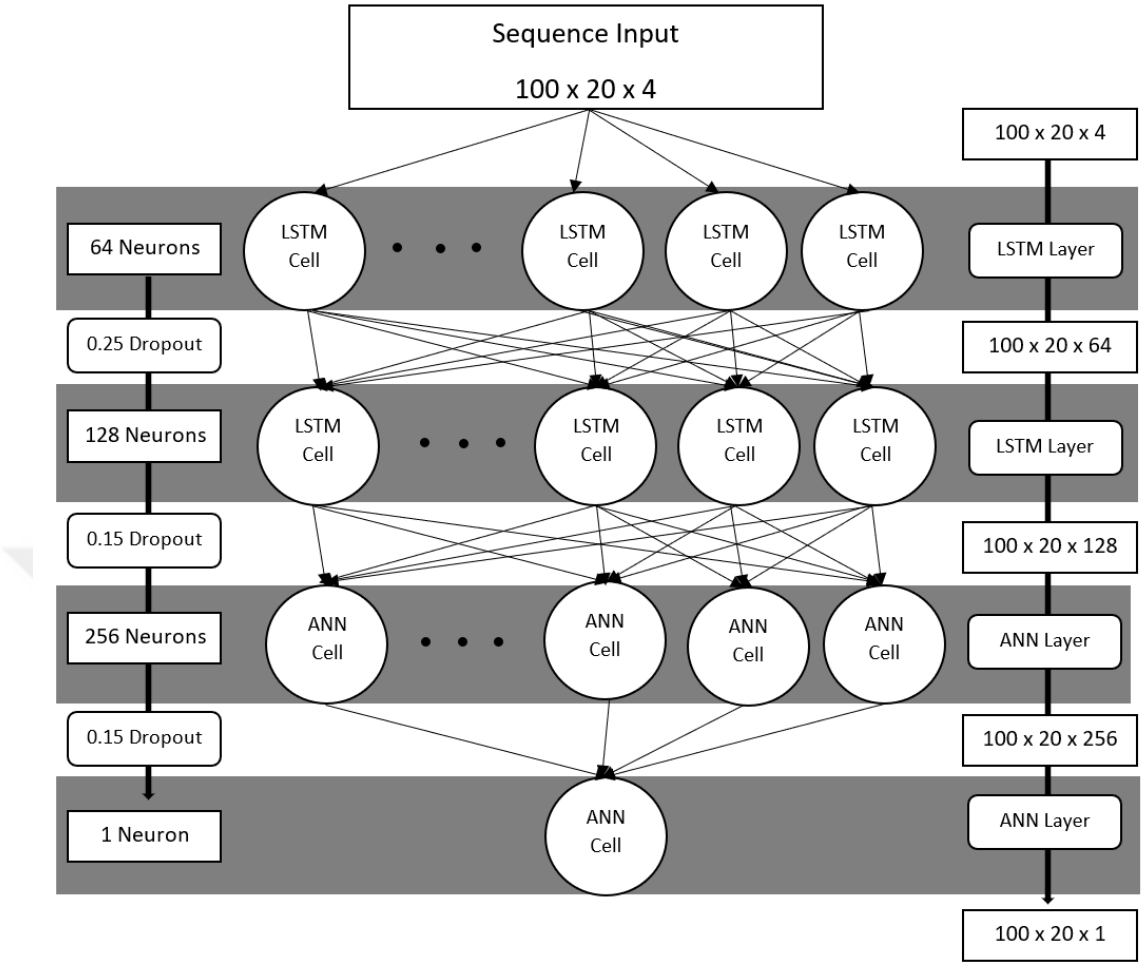

Figure 5. 5. LSTM Architecture

Variable learning rate is used while network training is being done. The rate, which was initially given as 0.01, has been gradually reduced to 0.001 during training. Adam optimizer was used and MSE was chosen for the loss function. The dataset is divided into 169 batches with 100 data each. A total of 20,280 iterations, that is, 120 epochs of training, were carried out until the targeted 0.015 MSE was reached. The MSE that changes in each iteration is given in Figure 5. 6.

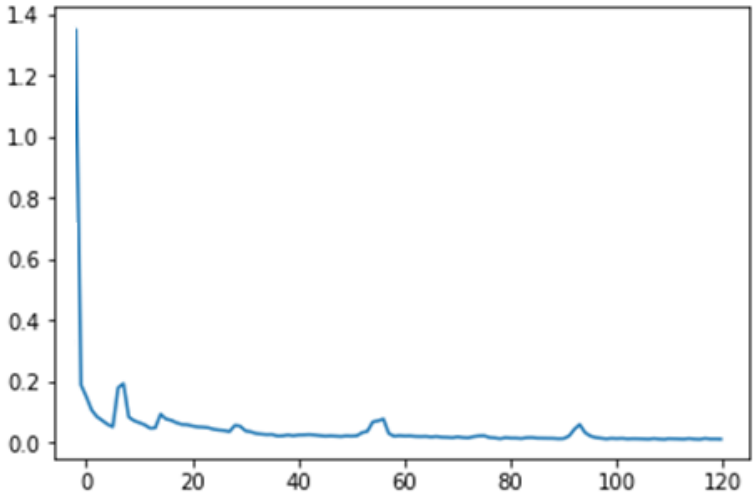

Figure 5. 6. Mean Squared Error

5.2.2. ANN

Unlike the CNN-based system, there is no time dependent variable in the HM-based positioning system. Therefore, ANN with a simpler structure was used instead of LSTM. The dataset is two-dimensional since it does not have any time intervals. With the previously mentioned ratios, it is divided into training, testing and validation as in Table 5. 3.

Table 5. 3. Dataset for ANN Training

	Train Data	Test Data	Validation Data
Input	192,000x4	96,000x4	32,000x4
Output	192,000x1	96,000x1	32,000x1

The built ANN architecture is as shown in Figure 5. 7. Adam optimizer was used with a learning rate of 0.01. MSE was used as the loss function. For the targeted 0.01 MSE, a total of 224000 iterations were trained with 224 epochs.

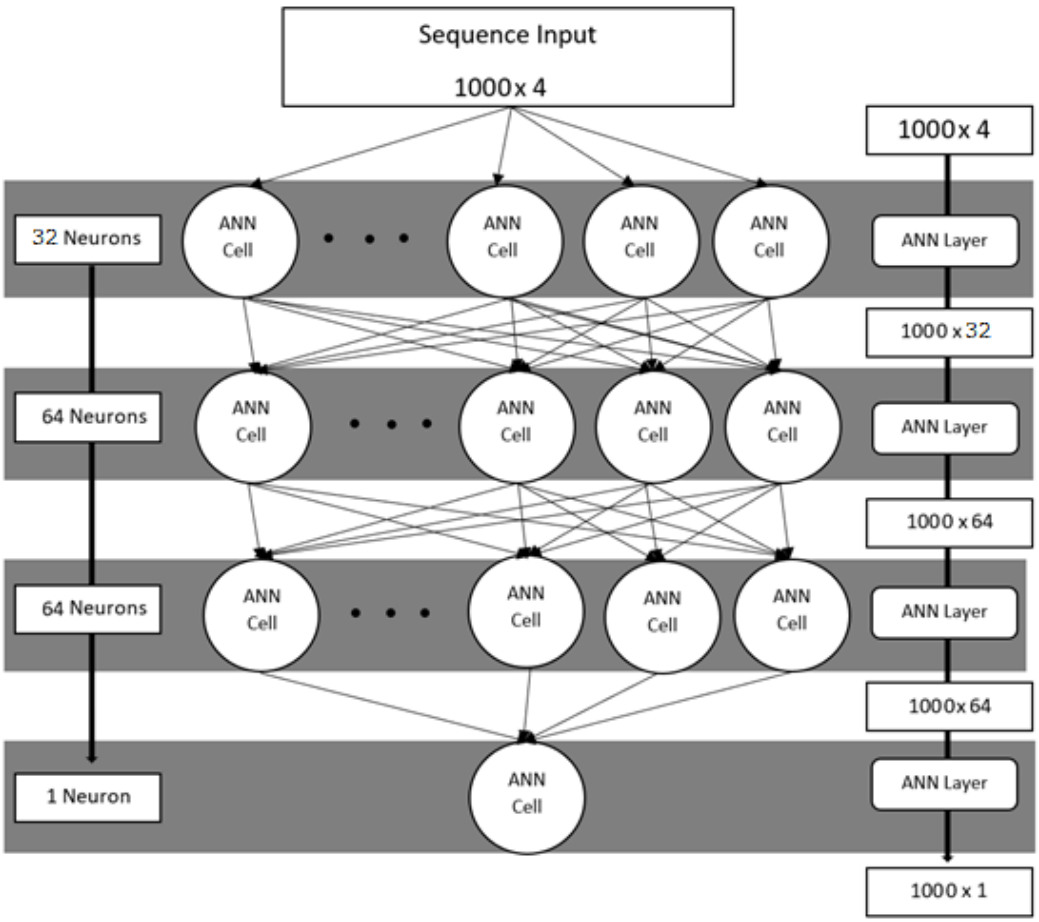

Figure 5. 7. ANN Architecture

6. TEST RESULTS AND COMPARISON

An example of nine steps with and without improving methods in CNN-based localization is shown. As the number of steps increases, the probability of deviation of the system that does not use the development method increases. This is because in the multi-pattern parts previously mentioned the positioning estimate becomes inefficient and the new estimates are stuck in the same region.

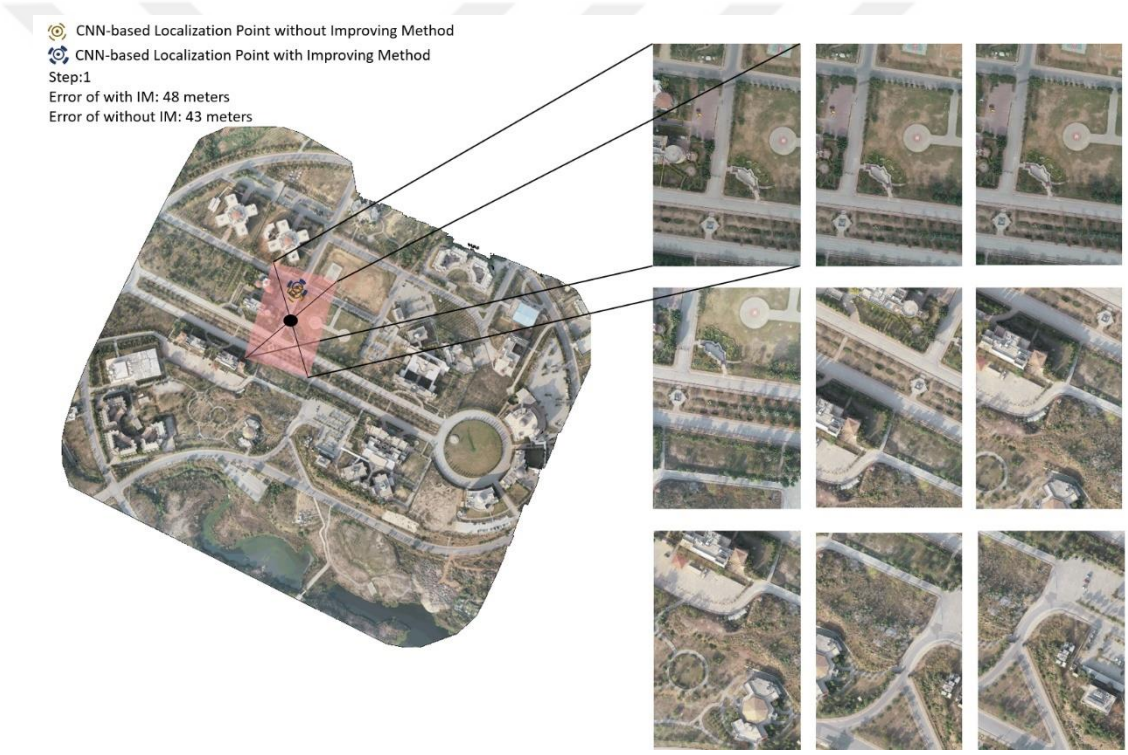

Figure 6. 1. CNN-Based Localization with and without IM Step:1

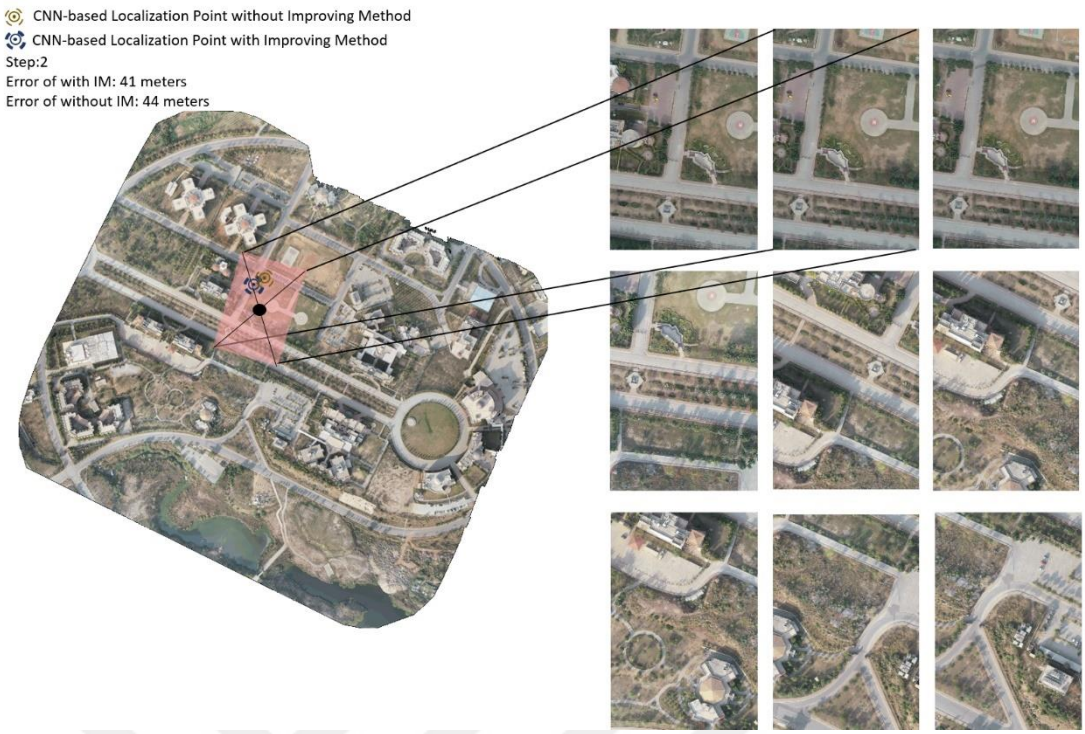

Figure 6. 2. CNN-Based Localization with and without IM Step:1

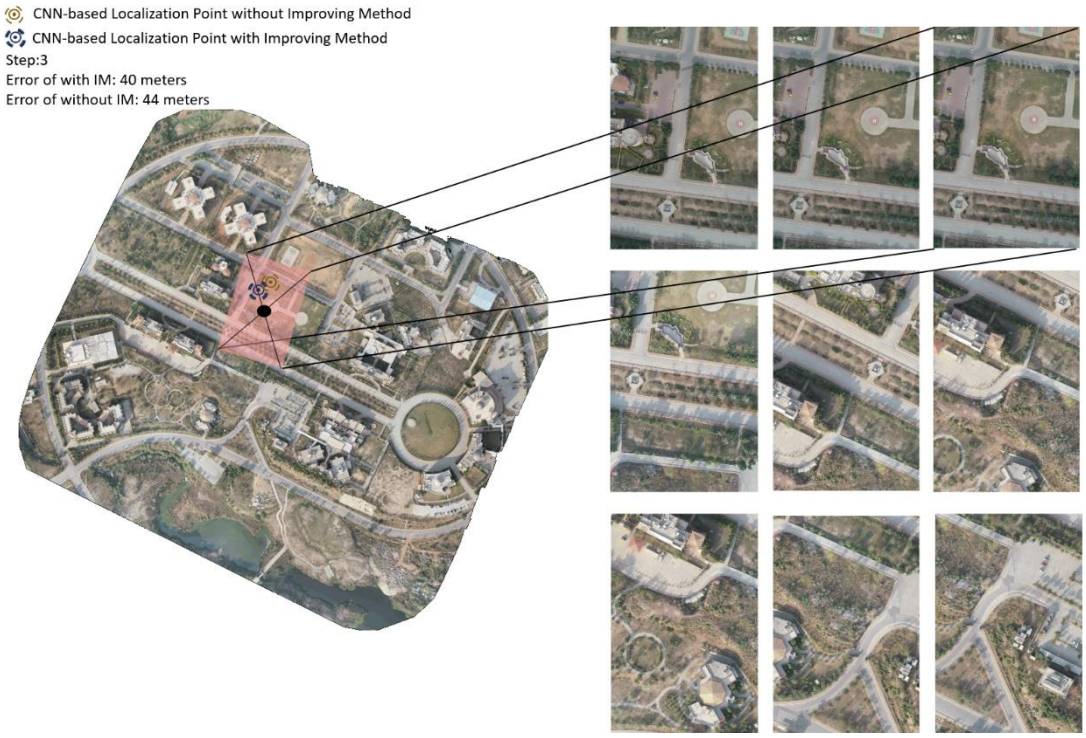

Figure 6. 3. CNN-Based Localization with and without IM Step:3

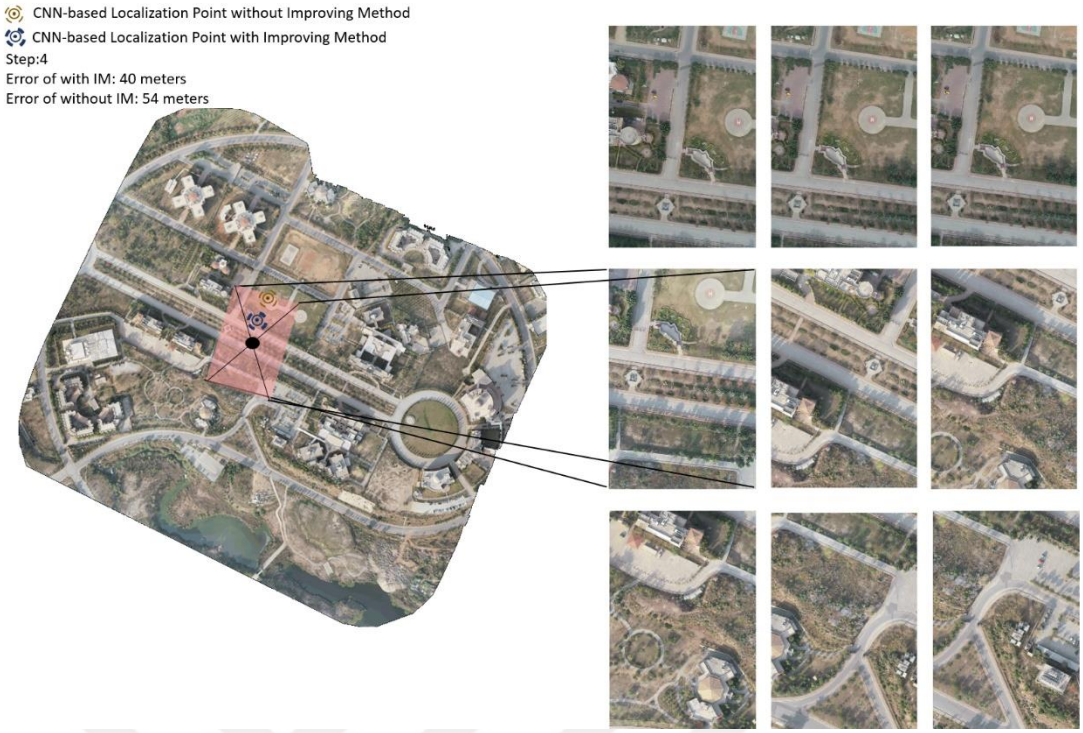

Figure 6. 4. CNN-Based Localization with and without IM Step:4

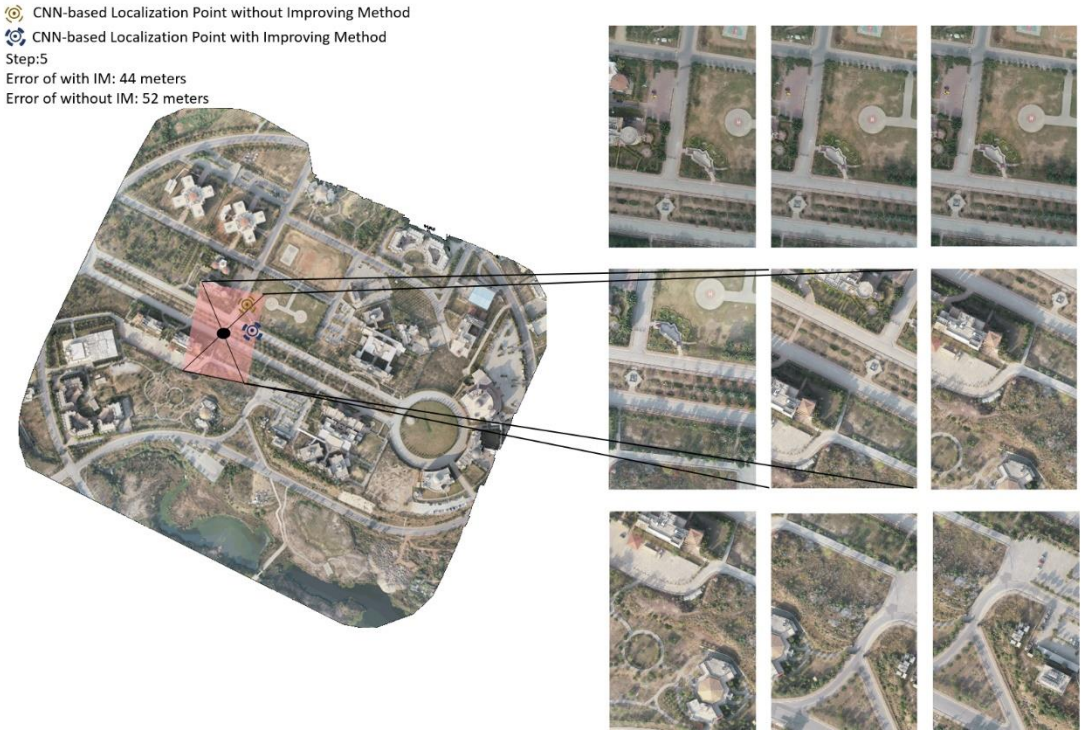

Figure 6. 5. CNN-Based Localization with and without IM Step:5

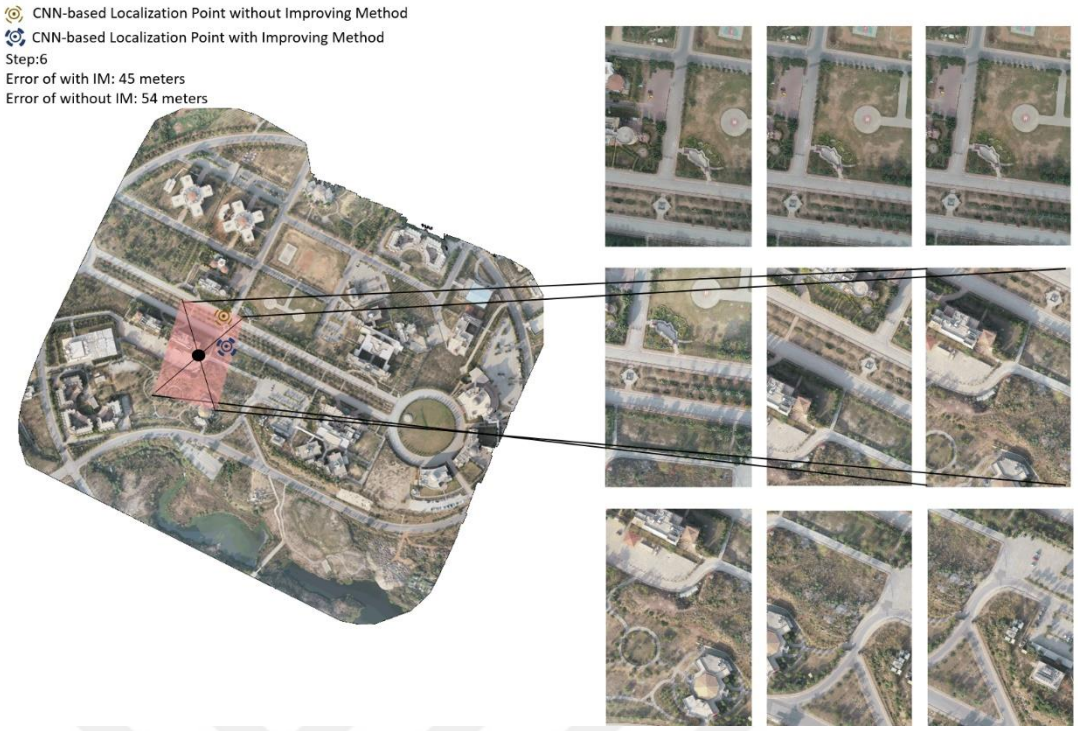

Figure 6. 6. CNN-Based Localization with and without IM Step:6

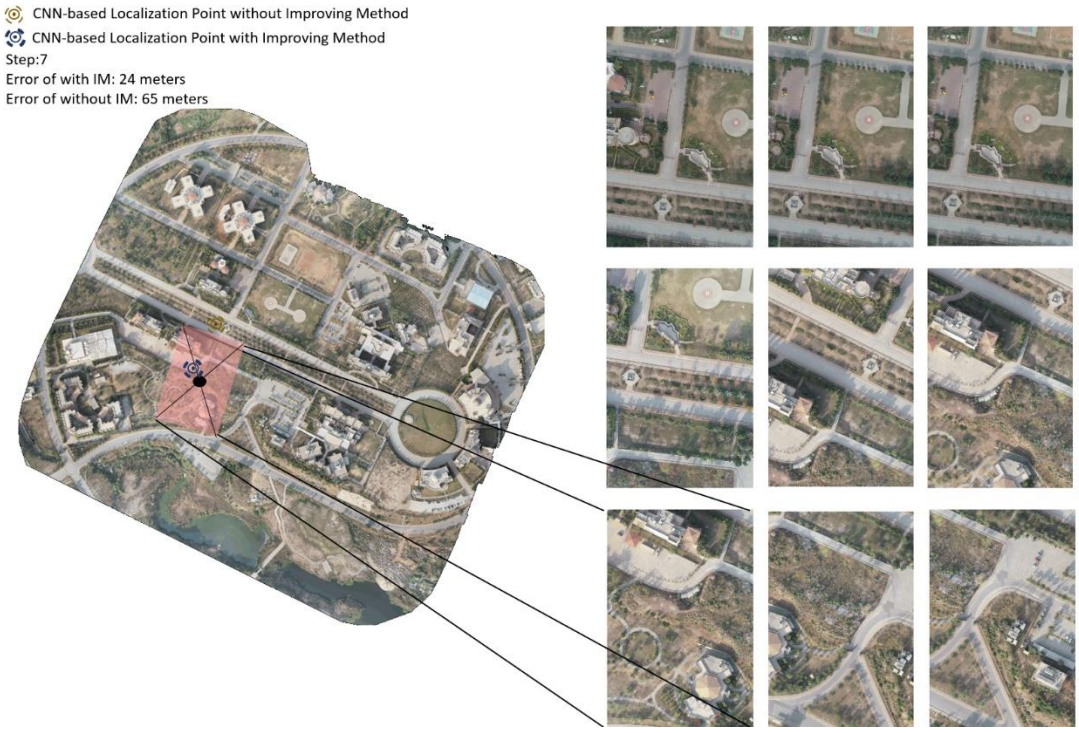

Figure 6. 7. CNN-Based Localization with and without IM Step:7

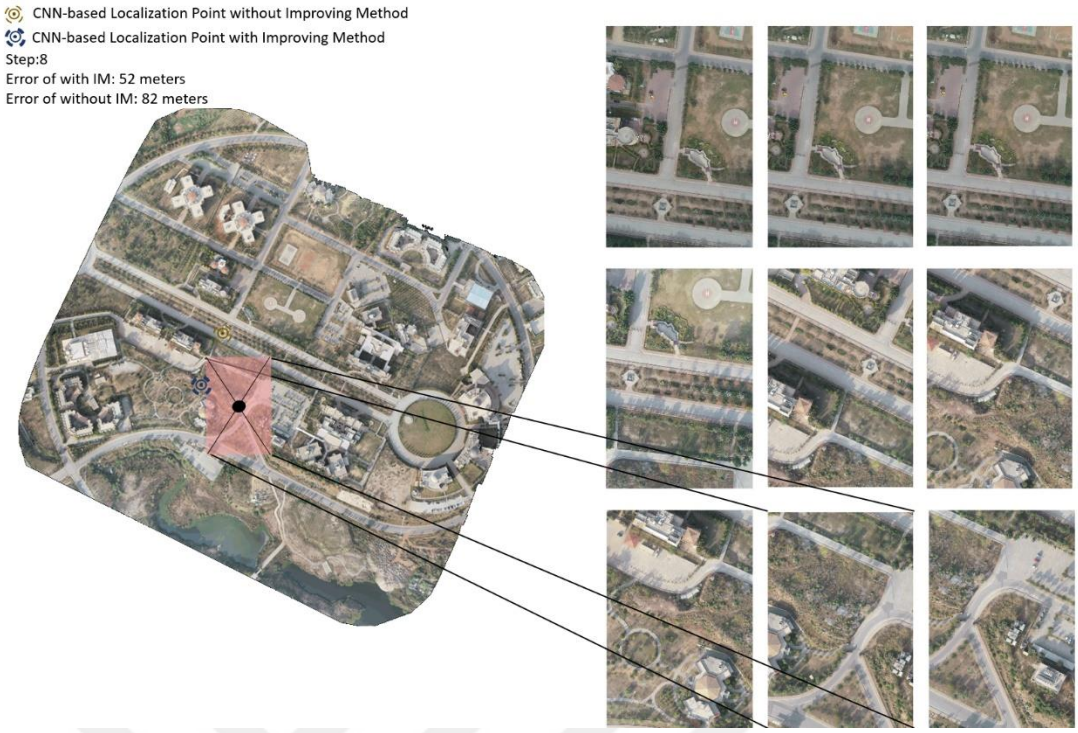

Figure 6. 8. CNN-Based Localization with and without IM Step:8

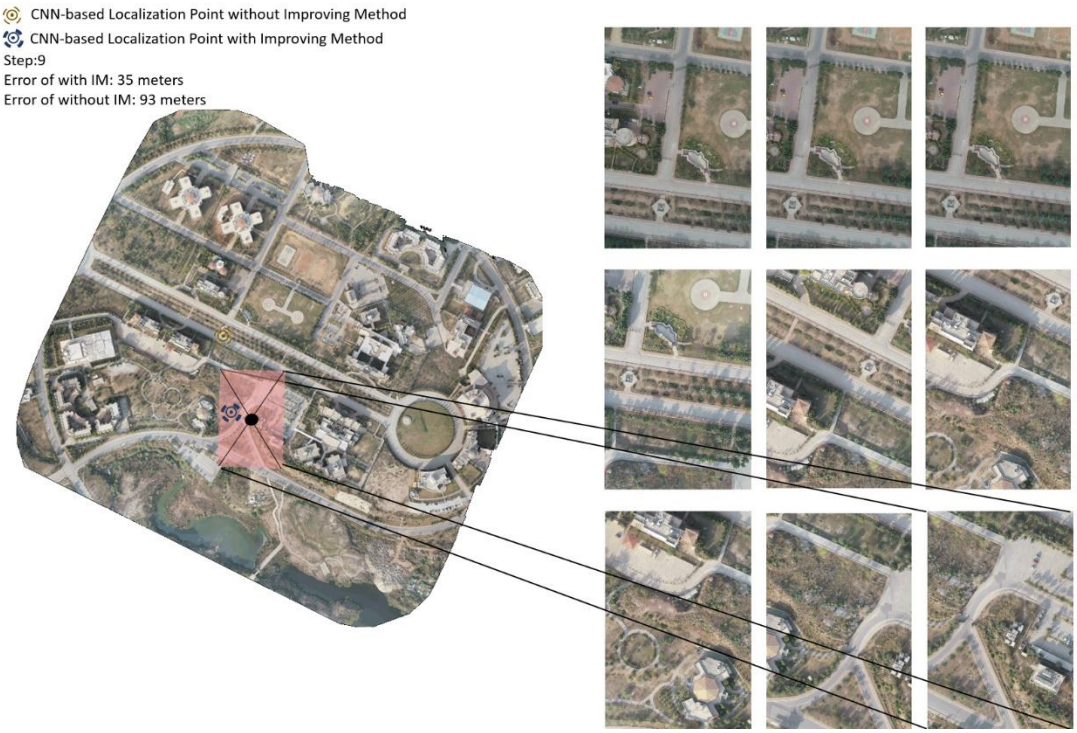

Figure 6. 9. CNN-Based Localization with and without IM Step:9

Figure 6. 10. and Figure 6. 11. shows the time dependent error rates of the systems. The systems are CNN-Based, HM-based, blended equally and adaptively blended. Blended equally data is obtained by blending CNN-Based and HM-Based systems with constant and equal weights. Although the adaptive blended system does not always converge to the dominant system in terms of accuracy, it generally gives better results than the equally weighted system. This result shows us a trade-off. When the platform is required to have back-up in terms of positioning source, the weighting problem can be solved by adaptive method. But a blended system will almost never yield better results than when the most accurate system available is used as the sole source. The main reason why blended systems are preferred is to prevent loss of control in case of not getting data from any source. What makes the adaptive blended system usable compared to other fixed weight blending systems is the deviations that may occur in the positioning sources. Deviations are detected by the trained network, and under conditions where deviations are predicted to occur, the weighting is in favour of the system that does not deviate or deviates less. This situation is shown in Figure 6. 12. and Figure 6. 13. In 7.2 percent of the tests, one or more sources were found to deviate more than 100 meters. Out of this 7.2 percent, 82 percent have deviations from a single source. The adaptive blended system further weighted the undeviated source in 74 percent of these single source deviations, largely avoiding the bias of the final positioning data. In addition, the performance of the systems is shown in Figure 6. 14. - Figure 6. 21. only by considering the changing weather conditions. As the performance criterion, the performance of the CNN-Based system in low fog, snowless environment and sufficient light was accepted as one and other systems were rated.

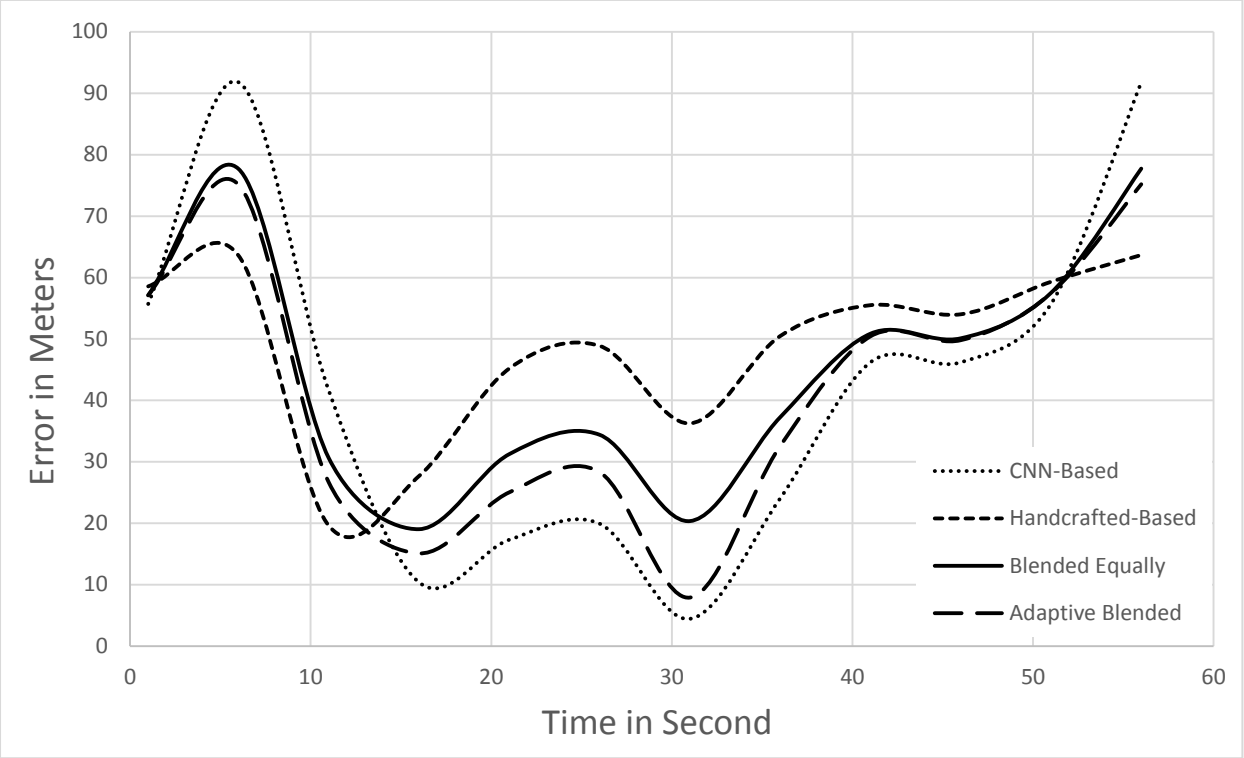

Figure 6. 10. Systems Error Comparison by Time

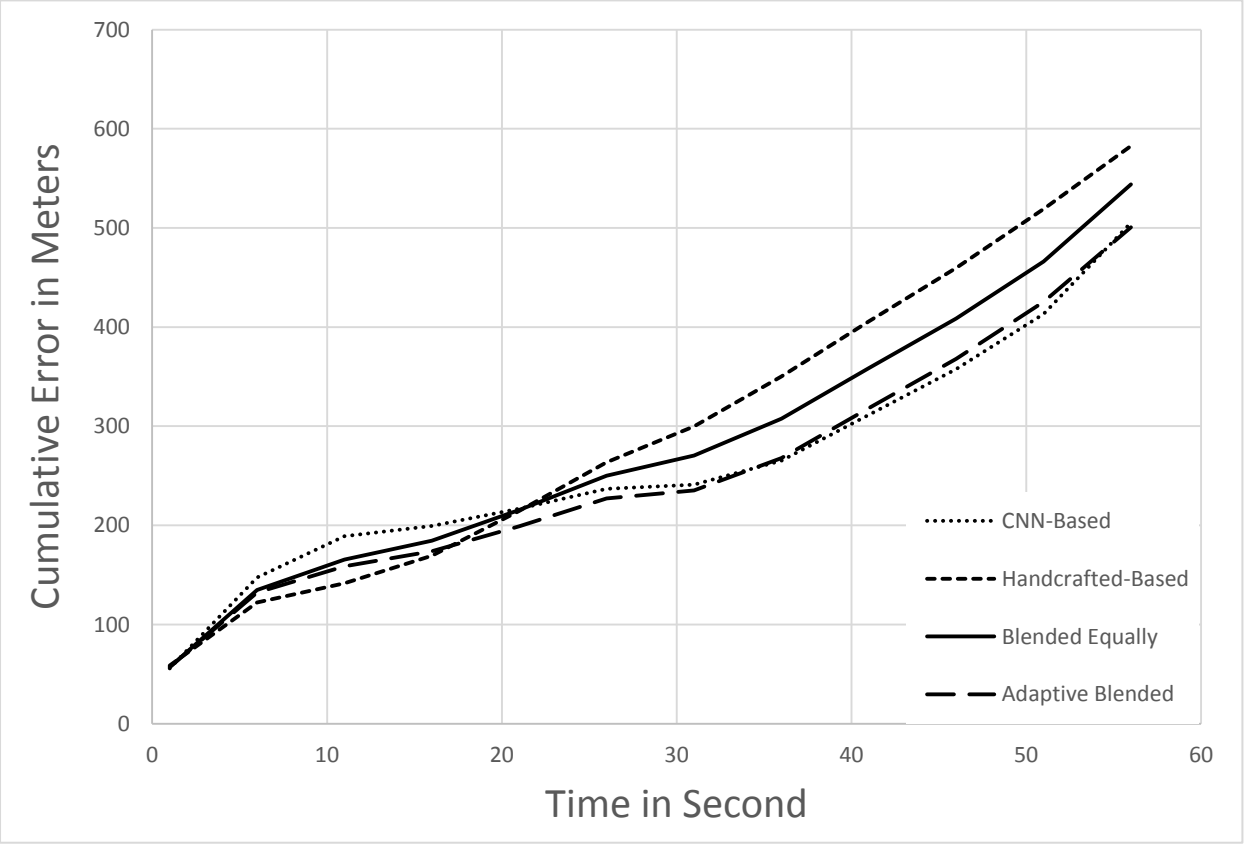

Figure 6. 11. Systems Cumulative Error Comparison by Time

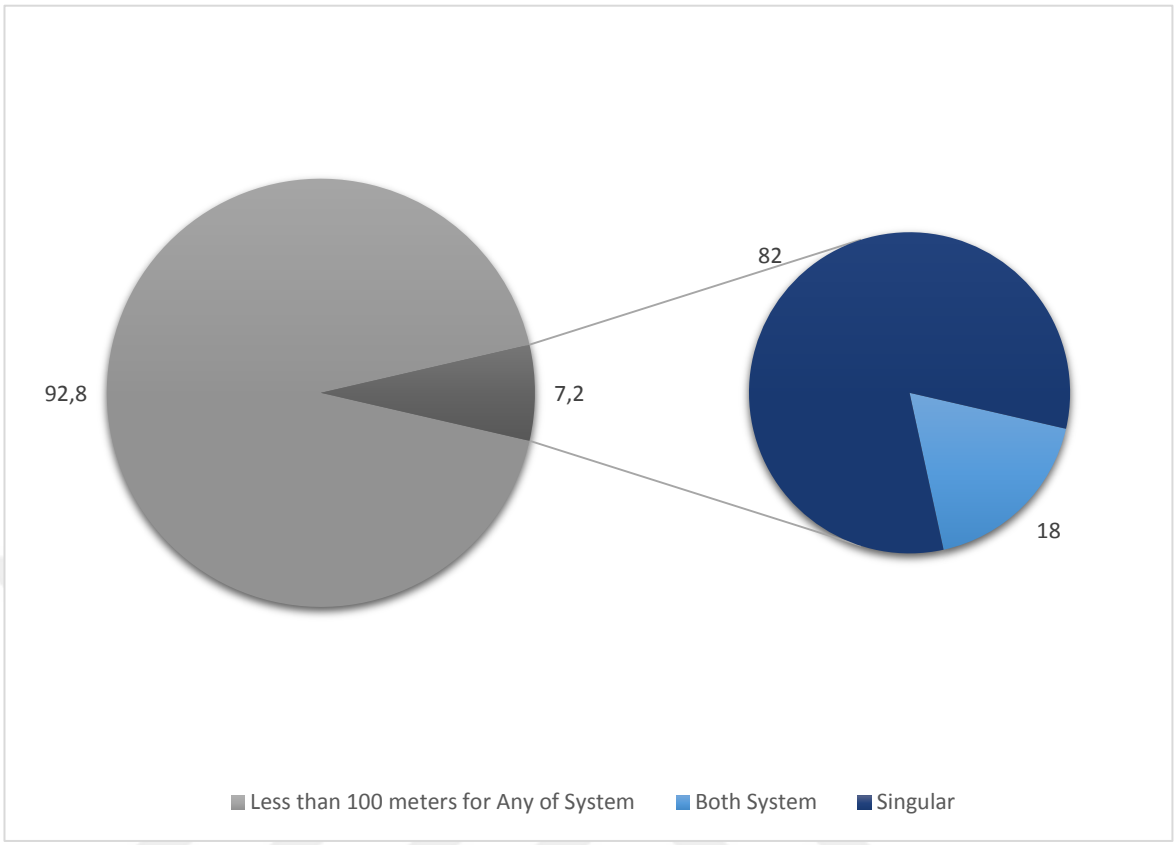

Figure 6. 12. Deviation Rates of Systems and Singularity of Deviation States

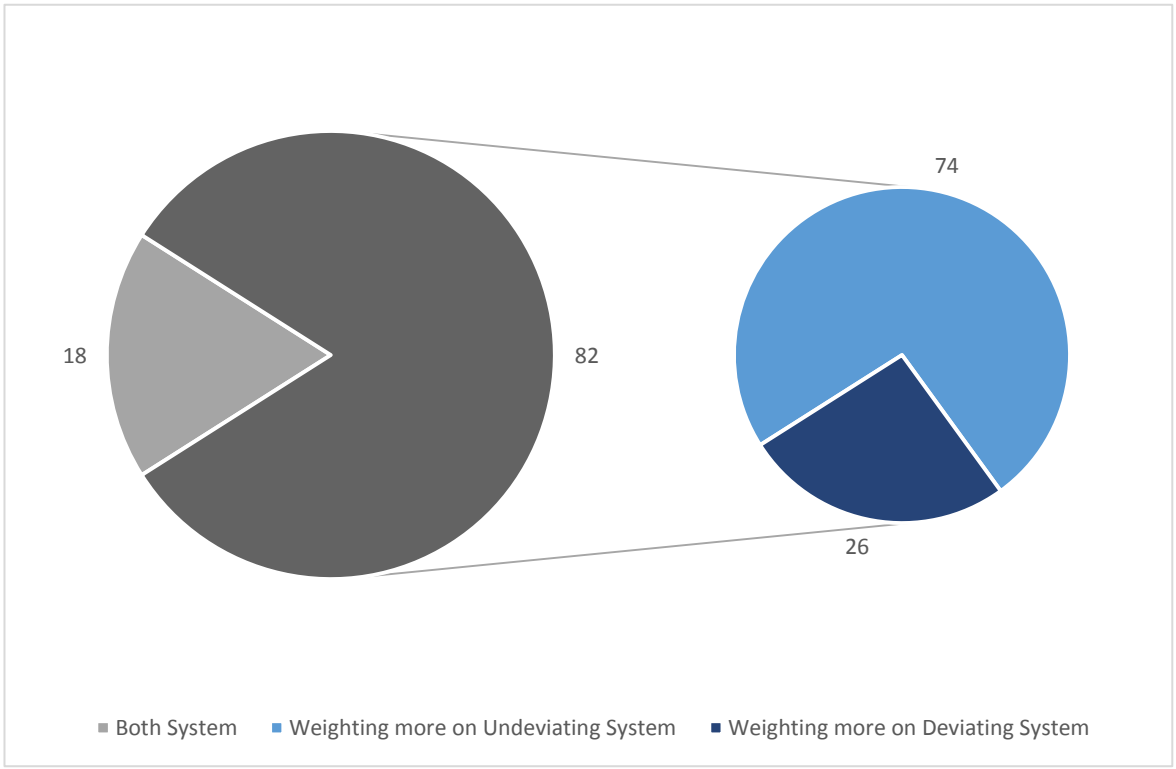

Figure 6. 13. Behaviour of Adaptive Blended System in Cases of Singular Aberrations

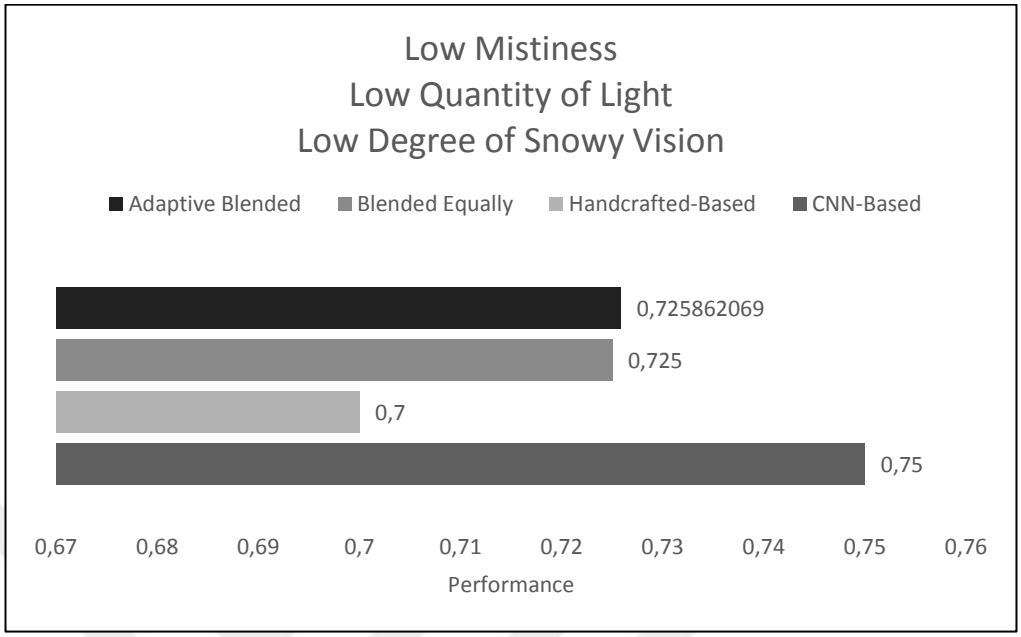

Figure 6. 14. Systems Performance Comparison in Scenario 1

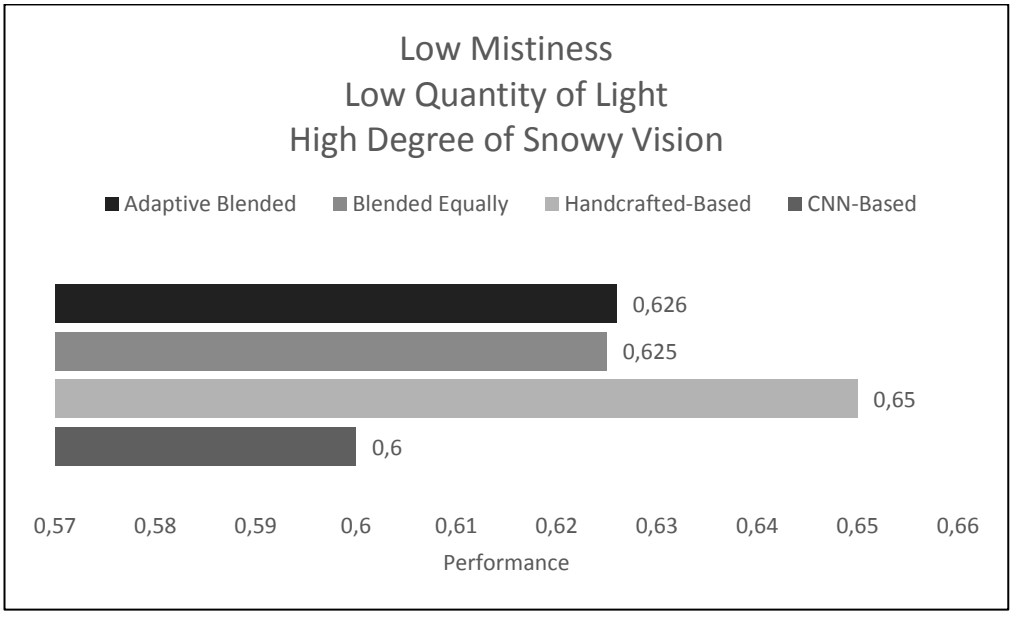

Figure 6. 15. Systems Performance Comparison in Scenario 2

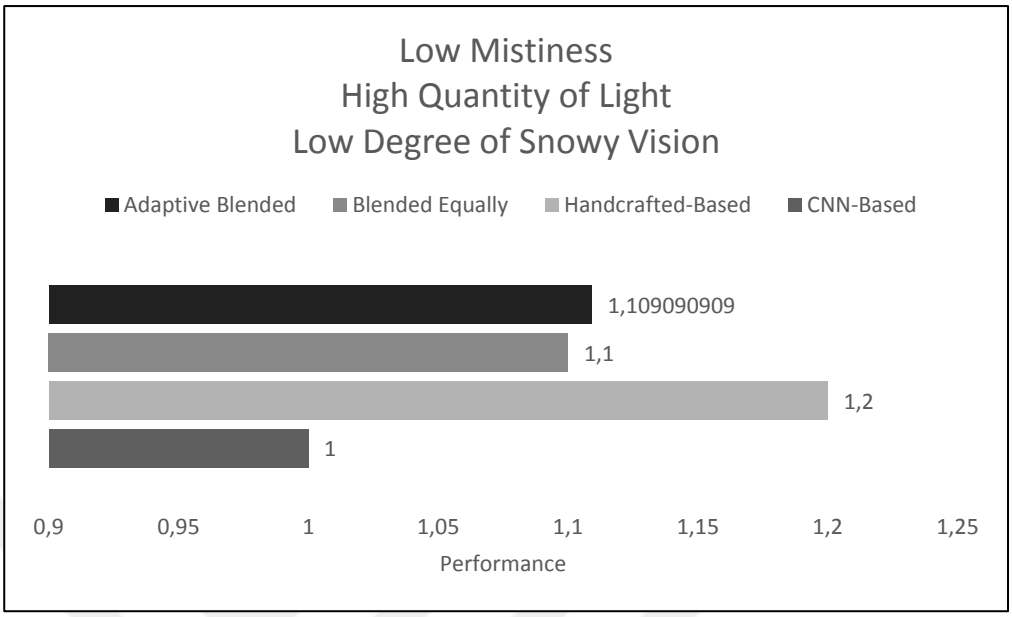

Figure 6. 16. Systems Performance Comparison in Scenario 3

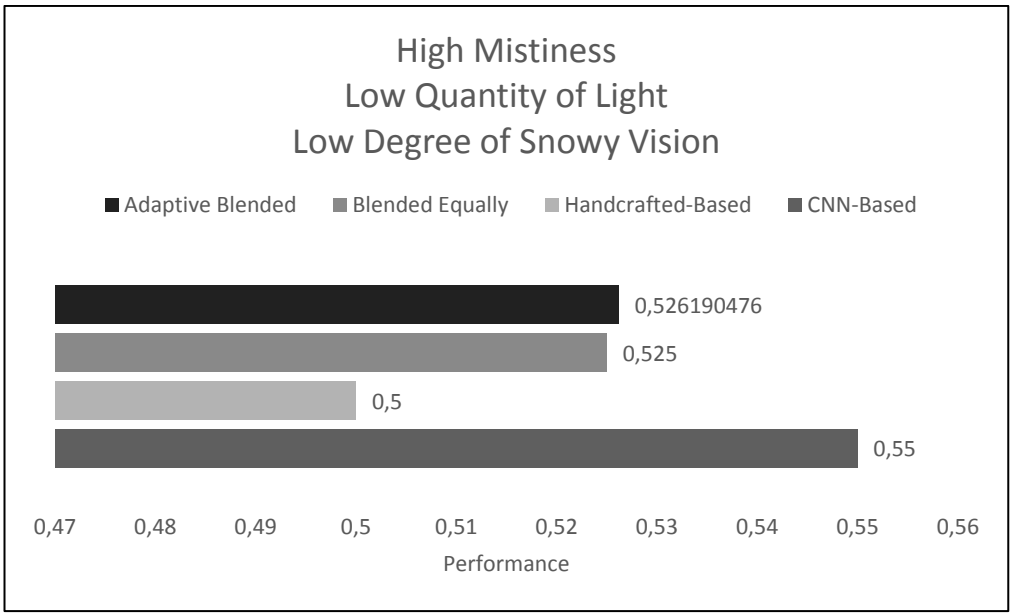

Figure 6. 17. Systems Performance Comparison in Scenario 4

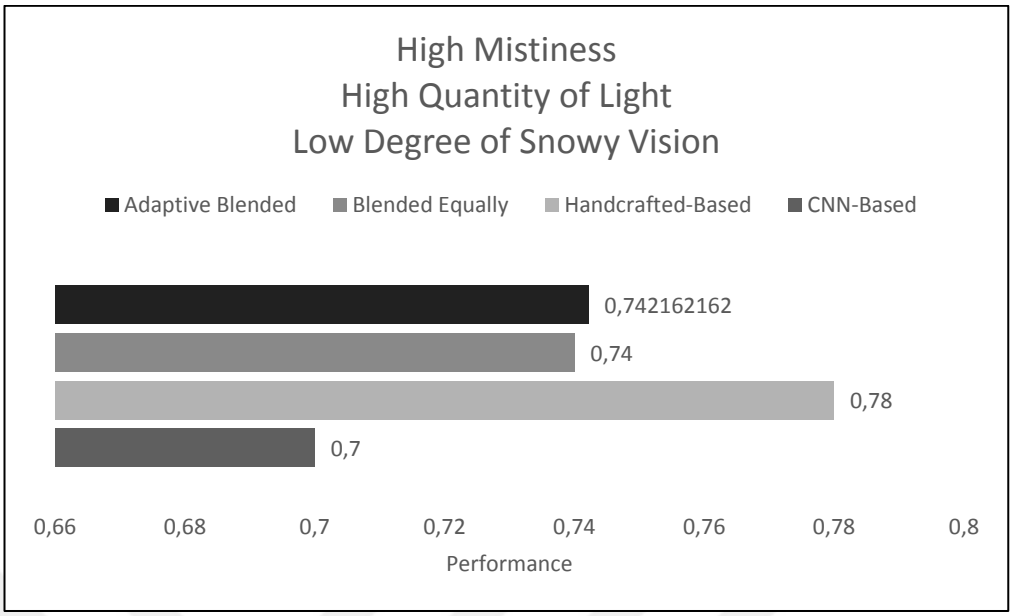

Figure 6. 18. Systems Performance Comparison in Scenario 5

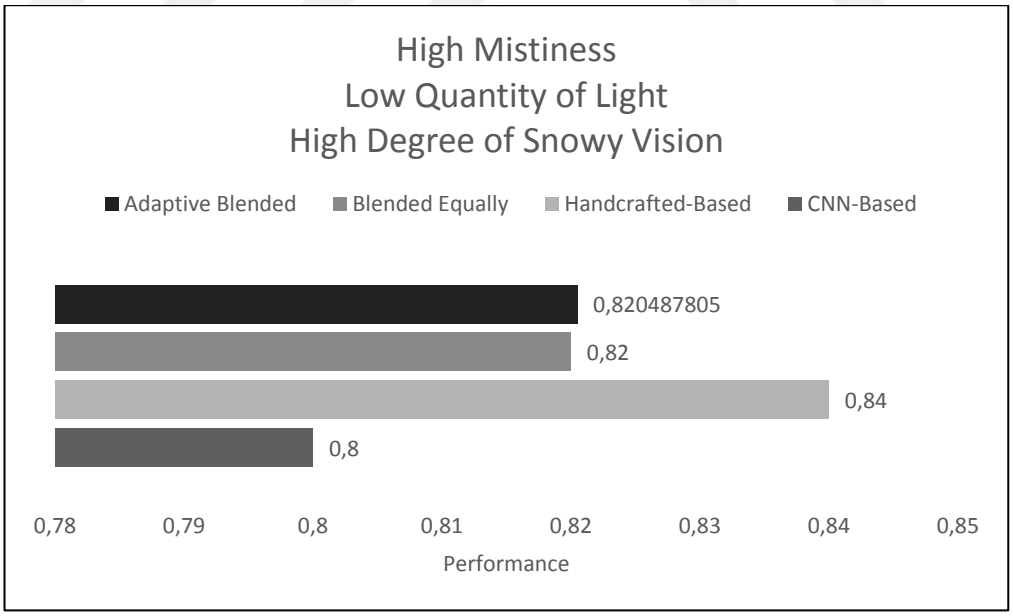

Figure 6. 19. Systems Performance Comparison in Scenario 6

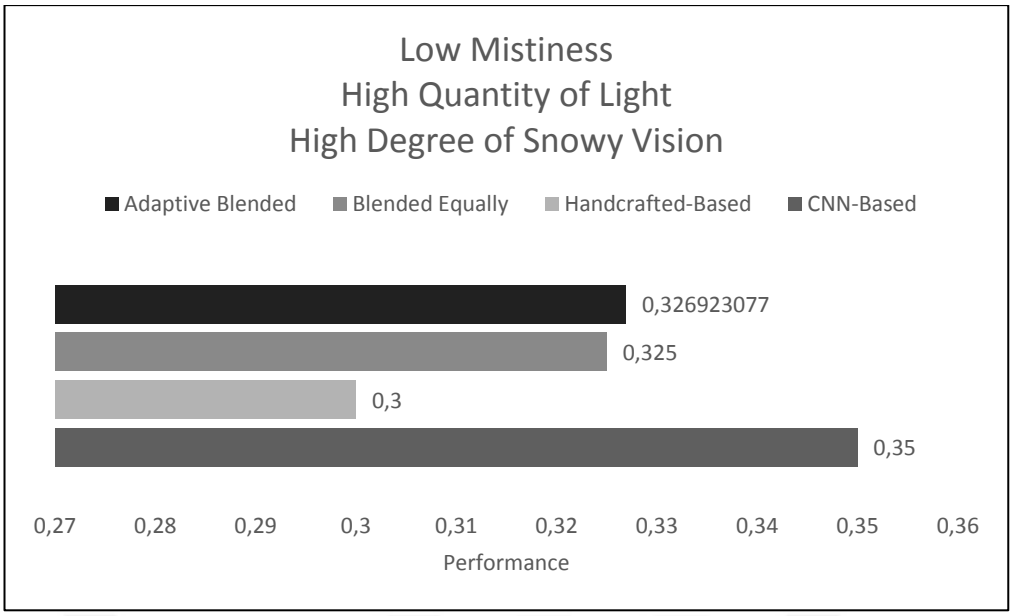

Figure 6. 20. Systems Performance Comparison in Scenario 7

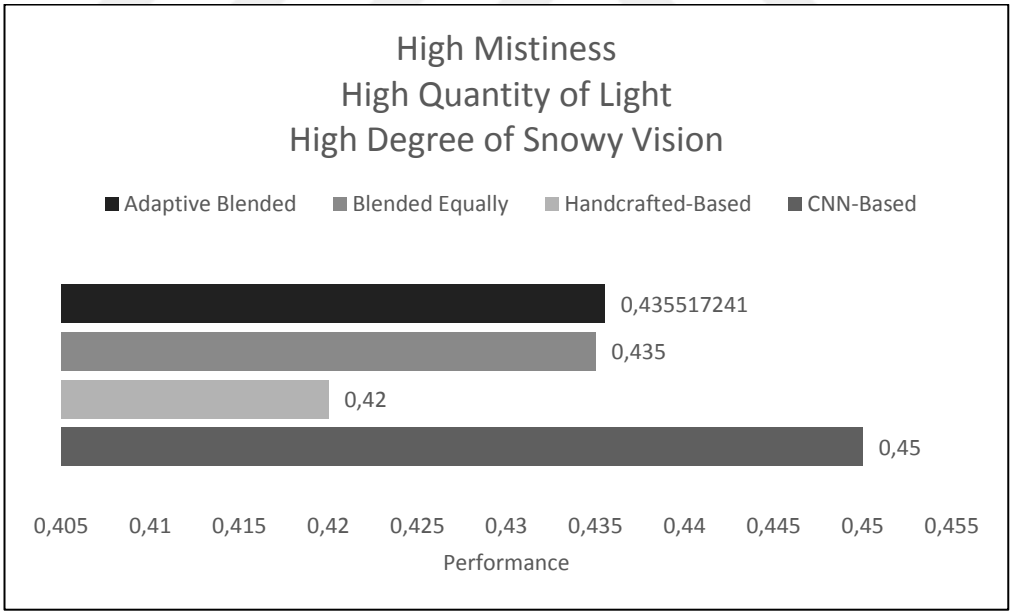

Figure 6. 21. Systems Performance Comparison in Scenario 8

7. DISCUSSION

Some of the issues that were mentioned briefly throughout the study and that was stated to be detailed later are explained in this section.

The first of these issues is adaptive scaling, which is mentioned in the Scaling section. The image data set used in the study includes terrain images taken from a fixed height. Therefore, the methods used throughout the study do not include elevation changes. Therefore, the images were initially scaled and then matched with this measure. In future studies, if a data set with variable height is obtained, improvements can be made in this regard.

Another issue is the reflection of the flight characteristics mentioned in the maneuver method to the position estimation process. In this study, instead of obtaining flight data during the flight, the maximum and minimum flight limits of the aircraft were determined at the beginning and the flight characteristics were reflected in the process in this context. Since the use of control surfaces and flight commands as input to the location estimation made by instant filtering, may be change the concept of this study, it can be mentioned in future studies.

The last issue is the determination of the variables in which the positioning systems are tested while the simulation is being prepared. At this stage, attributes such as control surface movements were not used. The reason for this is that simulating the flight commands to be given while performing the determined flight routes is challenging within the scope of this study. As mentioned before, it can be evaluated in future projects as a work in itself.

8. CONCLUSION

In tests using the trained network, it has been observed that positioning scenarios with adaptive blending have an average of 12 percent less errors than equally weighted blending. In addition, there is parallelism in the positioning errors of the two systems, this is due to the use of common resources. Although two different source processing methods are used, there is a similarity between the maximum and minimum points of error of visual positioning systems fed from the same source.

In the tests performed in different scenarios, it is seen that the adaptive blended system is between one and two percent more efficient than the fixed weighted system in cases where there is no deviation and only changing weather conditions are taken as criteria. The main reason for this situation is that the responses of the systems to variables such as light level, amount of snow and amount of fog show parallelism. In cases where the behaviour of the systems is not differentiated, the importance of the weights used in blending decreases. On the other hand, adaptive blending gives much more noticeable results when deviations and terrain features are also included in the evaluation.

REFERENCES

[1] A. Nassar, K. A. (2018). A Deep CNN-Based Framework For Enhanced Aerial Imagery Registration with Applications to UAV Geolocalization. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, (pp. 1594-159410). doi:10.1109/CVPRW.2018.00201

[2] Alan D., F. J. (1999). Nautical Almanac Office Sesquicentennial Symposium: U.S. Naval Observatory.

[3] Asche, G. (1972). The Omega System Of Global Navigation. *10th International Hydrographic Conference.*

[4] Burns, V. J. (1963). Aircraft Navigation: Design Theory for A Self-Organizing, High Accuracy Navigation System. *Tenth Annual East Coast Conference on Aerospace and Navigational Electronics*, (pp. 3.4.5-1-3.4.5-6). doi:10.1109/TANE.1963.4502273

[5] Chen, J.-X. &. (2005). Practical matching confidence analysis technique based on neural networks for normalized edge magnitude cross correlation. 722-725.

[6] Costea, D. &. (2016). Aerial image geolocalization from recognition and matching of roads and intersections. *ArXiv*. doi:abs/1605.08323

[7] Dufournaud, Y. &. (2020). Image Matching with Scale Adjustment.

[8] F. A. Gers, J. S. (1999). Learning to forget: continual prediction with LSTM. *Ninth International Conference on Artificial Neural Networks ICANN 99*, (pp. 850-855). doi:10.1049/cp:19991218

[9] F. Pappalardi, S. J. (2001). Alternatives to GPS. *An Ocean Odyssey* (pp. 1452-1459). MTS/IEEE Oceans 2001. doi:10.1109/OCEANS.2001.968047

[10] H. Zhu, L. J. (2019). A Novel Neural Network for Remote Sensing Image Matching. *IEEE Transactions on Neural Networks and Learning Systems*, 2853-2865. doi:10.1109/TNNLS.2018.2888757

[11] Hochreiter, S. &. (1997). Long Short-term Memory. *Neural computation*. doi:10.1162/neco.1997.9.8.1735

[12] I. Rocco, M. C. (2018). Neighbourhood consensus networks. *Adv. Neural Inf. Process. Syst*, 151-1662.

[13] J. Kim, S. W. (2017). Lidar-guided autonomous landing of an aerial vehicle on a ground vehicle. *2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence*, (pp. 228-231). doi:10.1109/URAI.2017.7992719

[14] J. R. G. Braga, H. F. (2016). An image matching system for autonomous UAV navigation based on neural network. *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, (pp. 1-6). doi:10.1109/ICARCV.2016.7838775.

[15] Liebe, C. C. (2002). Accuracy performance of star trackers - a tutorial. *IEEE Transactions on Aerospace and Electronic Systems*, 587-599. doi:10.1109/TAES.2002.1008988

[16] M. H. Mughal, M. J. (2021). Assisting UAV Localization Via Deep Contextual Image Matching. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2445-2457. doi:10.1109/JSTARS.2021.3054832

[17] Ma, J. J. (2021). Image Matching from Handcrafted to Deep Features: A Survey. *Int J Comput Vis 129*, 23–79. doi:10.1007/s11263-020-01359-2

[18] Mohanta, N. (2021, June 7). *How many satellites are orbiting the Earth in 2021?* Retrieved from https://www.geospatialworld.net/blogs/how-many-satellites-are-orbiting-the-earth-in-2021/

[19] Morelle, R. (2017, October 24). *Astrolabe: Shipwreck find 'earliest navigation tool'*. Retrieved from BBC: https://www.bbc.com/news/science-environment-41724022

[20] National Geographic Society. (1994). *Exploring Your World: The Adventure of Geography.* (D. J. Crump, Ed.) Washington: National Geographic Society.

[21] Needham, J. (1986). *The Shorter Science and Civilisation in China.* Cambridge: Cambridge University Press.

[22] Norman, B. (2012). A Brief History of Global Navigation Satellite Systems. *Journal of Navigation*, 65. doi:10.1017/S0373463311000506

[23] Pomerleau, D. (1993). Knowledge-based training of artificial neural networks for autonomous robot driving. *Robot Learning*, 19-43.

[24] R. S. Ornedo, K. A. (1998). GPS and radar aided inertial navigation system for missile system applications. *IEEE 1998 Position Location and Navigation Symposium*, (pp. 614-621). doi:10.1109/PLANS.1998.670222

[25] Raquet, A. C. (2017). Airborne Magnetic Anomaly Navigation. *IEEE Transactions on Aerospace and Electronic Systems*, 67-80. doi:10.1109/TAES.2017.2649238

[26] S. F. Lunsaeter, Y. I. (2020). Terrain Classification from an Aerial Perspective. *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, (pp. 173-177). doi:10.1109/SMC42975.2020.9283333

[27] T. Wang, C. S. (2018). Aerial-DEM Geolocalization for GPS-Denied UAS Navigation. *2018 Ninth International Conference on Intelligent Control and Information Processing (ICICIP)*, (pp. 289-294). doi:10.1109/ICICIP.2018.8606681

[28] Tamre, R. H. (2009). Aerial imagery terrain classification for long-range autonomous navigation. *2009 International Symposium on Optomechatronic Technologies*, 88-91. doi:10.1109/ISOT.2009.5326104

[29] The Franklin Continuous Radar Plot Technique. (2001). In *Radar Navigation and Maneuvering Board manual.* National Imagery and Mapping Agency.

[30] Thrun, S. (1998). Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 21-71. doi:10.1016/S0004-3702(97)00078-7

[31] W. Ma, J. Z. (2019). A Novel Two-Step Registration Method for Remote Sensing Images Based on Deep and Local Features. *IEEE Transactions on Geoscience and Remote Sensing*, 4834-4843. doi:10.1109/TGRS.2019.2893310

[32] W. Yang, Y. G. (2020). An Overview of Inertial Navigation Error Compensation and Alignment Calibration Methods. *IEEE 20th International Conference on Communication Technology (ICCT)*, 661-666. doi:10.1109/ICCT50939.2020.9295651

[33] Wang, Y. Z. (2019). A Lightweight Neural Network Framework for Cross-Domain Road Matching. *2019 Chinese Automation Congress (CAC)*, 2973-2978. doi:10.1109/CAC48633.2019.8996270

[34] Yilmaz, O. E. (2013). A novel fast and accurate algorithm for Terrain Referenced UAV localization. *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, 660-667. doi:10.1109/ICUAS.2013.6564746

[35] Z. Yang, T. D. (2018). Multi-Temporal Remote Sensing Image Registration Using Deep Convolutional Features. *IEEE Access*, 38544-38555. doi:10.1109/ACCESS.2018.2853100

AUTOBIOGRAPHY

EDUCATION:

- **Bachelor's Degree:** 2018, University of Turkish Aeronautical Association, Aeronautical Engineering

PROFESSIONAL EXPERIENCE:

- 2 year, Turkish Airlines Technic, Aircraft System Engineer
- 1 year, Turkish Aerospace İndustry, Unmanned Air Vehicle System Engineer

PATENT:

- Vertical Take-off Landing Unmanned Air Vehicle, TR201723297A1, December, 2017.