GPS-Free Navigation Using Vision-Based Convolutional Neural Networks

Ayman Hamdy KASSEM*,[1], Hamdy Ayman HAMDY[2]

*Corresponding author
*,[1]Aerospace Engineering Department,
Cairo City, Giza, Egypt,
akassem@cu.edu.eg
[2]Faculty of Urban and Regional Planning,
Cairo University, Giza, Egypt,
Ayman_222031@stud.furp.cu.edu.eg

DOI: 10.13111/2066-8201.2025.17.1.2

Received: 04 February 2025/ Accepted: 20 February 2025/ Published: March 2025

Abstract: *This paper presents a novel approach to address the challenge of self-localization of flying vehicles. It utilizes visual cues provided by the map imagery fed to a map-recognition convolution neural-network (CNN). This approach is invaluable during the navigation of flying vehicles in scenarios where the Global Positioning System (GPS) signal is unavailable. The proposed approach leverages the power of convolutional neural networks (CNNs) to imitate the visual perception and navigation abilities of homing pigeons, enabling the vehicle to navigate using solely real- time visual data with limited or no GPS information. Two pre-trained CNN's (SqueezeNet and GoogLeNet) are selected and re-trained with Google Maps imagery, enabling them to efficiently learn and generalize from the diverse visual attributes present in the map. Extensive experimentation and evaluation have demonstrated the efficacy and resilience of the vision-based GPS-free navigation system. The resulting system predicts position accurately achieving an accuracy of 89.9% and 96.4% for SqueezeNet and GoogLeNet, respectively, for images with a resolution of (one km* x *one km) and reaching an accuracy of 94.7 for GoogLeNet for images with a resolution of (374 m* x *374 m). Results underscore the potential of this approach for overcoming the challenge of GPS unavailability in aerial navigation.*

Key Words: *Convolutional Neural Network, CNN, GPS-free Navigation, Visual Perception, Image-based Navigation*

1. INTRODUCTION

Navigation systems play a pivotal role in various applications, including autonomous vehicles and unmanned aerial systems. However, in certain scenarios, such as indoor environments or areas with GPS signal obstruction, relying solely on GPS for navigation becomes impractical. To overcome this limitation, this work proposes a vision-based GPS-free navigation system that employs a convolutional neural network (CNN) to enable a flying vehicle to navigate autonomously using visual information. The proposed system is inspired by homing pigeons which have an extraordinary ability to navigate and find their way back to their home lofts from distant locations. While the exact mechanisms used by homing pigeons, are not fully understood, researchers have identified several factors that contribute to their navigation

abilities. Visual Landmarks are one of these factors. Pigeons are adept at recognizing and memorizing visual landmarks along their route. They use prominent landmarks such as rivers, mountains, or distinctive buildings to orient themselves and stay on track [1]. The proposed system leverages CNNs to mimic this capability by extracting high-level features from imagery.

The idea of using a camera in localization has gained publicity in research. Syed-Yaser et al. (2023) [2] have combined a monochrome camera and IMU and used Kalman filter to support autonomous landing of UAV. This paper focuses on the autonomous landing of a UAV which necessitates reliable information for near-ground maneuvers. Karkar et al. (2021) [3] presented an indoor navigation mobile-based system based on image processing (matching) techniques. It uses multiscale local binary pattern (MSLBP) features to recognize places. The system has been compared with existing systems that use rotated robust independent elementary features (ORB) and scale invariant feature transform (SIFT) features, but it suffers from the inability to differentiate between different locations that look similar. Chang et. al. (2023) [4] surveyed different techniques used for UAV autonomous navigation in GPS-denied environments. They used CNN-based Distance Estimation in reality-like indoor environments. Elaraby et al. [5] used image processing techniques for object detection and distance estimation for improving rover navigation and SLAM (Simultaneous Localization and Mapping). A method is described to return the aircraft from the GPS-degraded region to its launch point by means of visual navigation techniques. Lewis et. al. [6] used a framework for visual return-to-home capability in GPS-denied environments. They record a sequence of overlapping key frames, on the outbound flight. On the return flight, they use holography between frames to bring the aircraft back. Abozied et al. [7] proposed a cascaded neural networks approach for improving navigation system efficiency during GPS outage. The networks are used to better estimate velocity and position errors to improve navigation accuracy.

The main function of the proposed approach is visual localization (i.e. finding the Longitude and Latitude of a given area using an image of this area). The test case of this system is a rectangular area of 450 square km around Cairo University as shown in Figure 1.

Figure 1. Test area Cairo City [31.09130859375 ,30.10711788709] [31.30004882812, 29.89780561015] (Google maps)

2. METHODOLOGY

Convolutional Neural Network Architecture

A Convolutional Neural Network (CNN) is a specialized deep learning architecture designed to process structured grid data, such as images, and is widely used in tasks like image classification, object detection, and segmentation [8]. CNNs automatically extract hierarchical features from raw input data, reducing the need for manual feature engineering. The architecture typically consists of convolutional layers that apply filters to the input to detect patterns, pooling layers that reduce spatial dimensions while retaining essential features, and fully connected layers that aggregate the extracted features for final predictions. Convolutional layers preserve spatial relationships and learn local dependencies, while activation functions like ReLU introduce non-linearity to model complex data patterns. Pooling layers, such as max pooling, help reduce computational costs and control overfitting by summarizing feature maps. CNNs often include normalization layers, such as batch normalization, to stabilize training, and dropout layers to prevent overfitting. The output layer, tailored to the task, uses activation functions like SoftMax for classification or linear for regression. CNNs excel at feature extraction and hierarchical learning, capturing both low-level and high-level patterns in images. Their advantages include parameter sharing, translation invariance, and robust performance across varying visual data. Applications range from image classification and medical imaging to autonomous vehicles and object detection, making CNNs a cornerstone of modern computer vision. Two CNN`s were used in this work: GoogLeNet and SqueezeNet.

SqueezeNet

Introduced by Iandola et al. (2016) [9], SqueezeNet typically consists of 18 layers, including convolutional layers, pooling layers, and fully connected layers. These layers are organized into fire modules, which are the building blocks of the architecture. The number of layers in each fire module can vary, but the overall network depth remains shallow compared to other deep neural network architectures. The compactness of SqueezeNet is achieved by reducing the number of parameters while maintaining competitive performance on image classification tasks. It is a compact and lightweight deep neural network architecture designed for efficient inference on resource-constrained devices, such as mobile phones and embedded systems. It was developed to address the need for smaller and faster models without compromising on accuracy. The Matlab implementation of SqueezeNet is given in Figure 2.

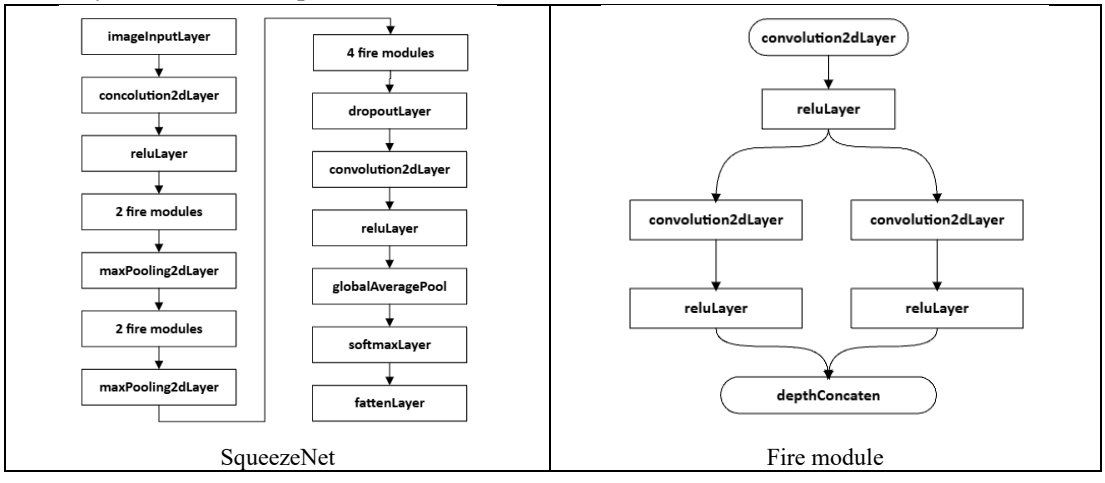

Figure 2. SqueezeNet graphical representation

The basic elements of SqueezeNet are explained in Table 1.

Table 1. Explanation of SqueezeNet elements

Element	**Description**
imageInputLayer	Specifies the dimensions of the images fed into the network (e.g., 256×256 ×3 for out test case).
convolution2dLayer	Applies a set of convolutional filters to extract low-level features
reLULayer	Rectified Linear Unit (ReLU) activation function
Fire Module	A building block of SqueezeNet
maxPooling2dLayer	Down samples feature maps by retaining the most important features, reducing spatial dimensions and computational complexity.
dropoutLayer	Introduces regularization by randomly deactivating neurons during training, helping to prevent overfitting.
globalAveragePoolingLayer	Replaces fully connected layers by computing the average of each feature map, reducing model size and preventing overfitting.
softmaxLayer	Converts the feature vector into class probabilities by applying the softmax function.
classificationLayer	Assigns a label to the input image based on the output probabilities from the softmax layer.

GoogLeNet

Introduced by C. Szegedy et al. (2015) [8], the GoogLeNet architecture consists of 22 layers, including convolutional layers, pooling layers, and fully connected layers. One of the key innovations introduced in GoogLeNet is the concept of the Inception module. The Inception module employs multiple parallel convolutional operations of different filter sizes within a single layer, allowing the network to capture features at various scales and abstraction levels. This parallelization helps the network learn a diverse set of features, enhancing its ability to recognize complex patterns in images.

The Matlab implementation of GoogLeNet is given in Figure 3.

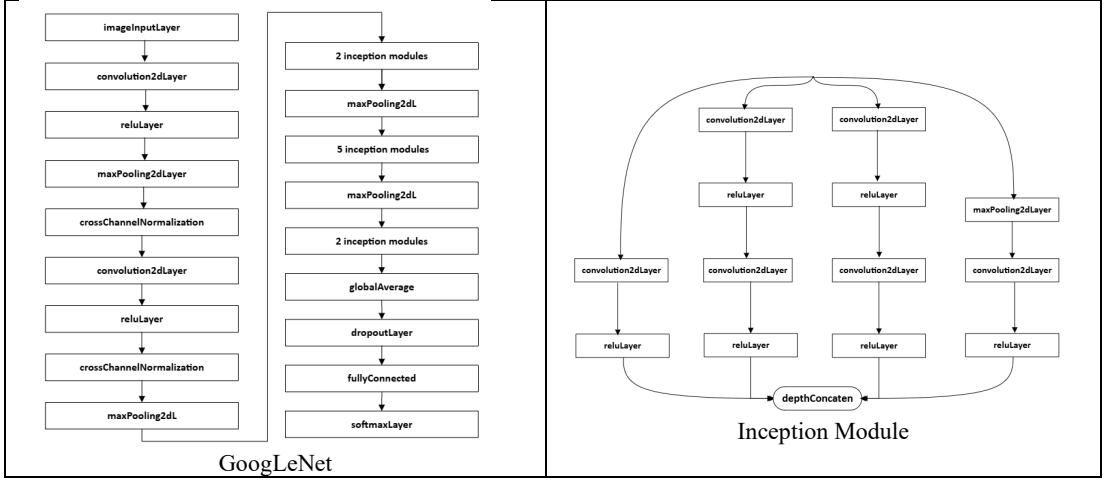

Figure 3. GoogLeNet graphical representation

The basic elements of GoogLeNet are explained in in Table 2

Table 2. Explanation of GoogLeNet elements

Element	Description
imageInputLayer	Specifies the dimensions of the images fed into the network (e.g., 256×256 ×3 for out test case).
convolution2dLayer	Applies convolutional filters to extract low-level features
reluLayer	The Rectified Linear Unit (ReLU) activation function.
maxPooling2dLayer	Down samples feature maps by selecting the maximum value in each region, reducing spatial dimensions and computational cost while retaining key features.
crossChannelNormalizationLayer	Normalizes feature maps across channels to stabilize training and improve convergence, often used after ReLU activations.
Inception Module	A core innovation of GoogLeNet, enabling multi-scale feature
- **1x1 Convolutions**	Reduce feature map dimensions, acting as bottleneck layers to limit computational cost.
- **3x3 and 5x5 Convolutions**	Capture features at different spatial scales, enabling the network to learn diverse representations.
- **maxPooling2dLayer (within module)**	Extracts dominant features at a broader scale and feeds them into the depth concatenation layer.
- **depthConcatenationLayer**	Combines outputs from parallel convolutions and pooling into a unified feature map for the next layer.
globalAveragePoolingLayer	Replaces fully connected layers by computing the average of each feature map, reducing model size and preventing overfitting.
dropoutLayer	Introduces regularization by randomly deactivating a fraction of neurons during training to improve generalization.
fullyConnectedLayer	A dense layer that aggregates all extracted features into a single vector for classification.
softmaxLayer	Converts the feature vector into class probabilities by applying the softmax function.
classificationLayer	Assigns a label to the input image based on the output probabilities from the softmax layer.

Data Preparation

The images for the specified area around Cairo University were extracted from Google Maps with two spatial resolutions. Each image was tagged by its longitude and latitude to represent a separated class.

For spatial resolution 1 (case one), we have 22x19 image/ area which approximately equivalent to images with spatial resolution of (1 km × 1 km) per image.

For special resolution 2 (case two) we have 86 × 73 image/ area which approximately equivalent to images with spatial resolution of (374 m × 374 m) per image.

For each case three different image resolutions were collected (512x512 pixels, 768x768 pixels, and 1024x1204 pixels). For the first spatial resolution, each image represents a class which we would like our network to identify.
This sums us to 22x19 = 418 classes. To prepare enough images for training and testing, the collected images (512x512 pixels, 768x768 pixels, and 1024x1204 pixels) are scaled to 256x256 and rotated at different angles and some noise was introduced.
This sums up to 216 images per location for each of the 418 locations (classes) making the total images used for training 90,288 images. An additional 108 images for each location (class), produced with different rotation angles than the one used for training, were used for the testing of the final network.
Samples of training images are shown in Figure 4.

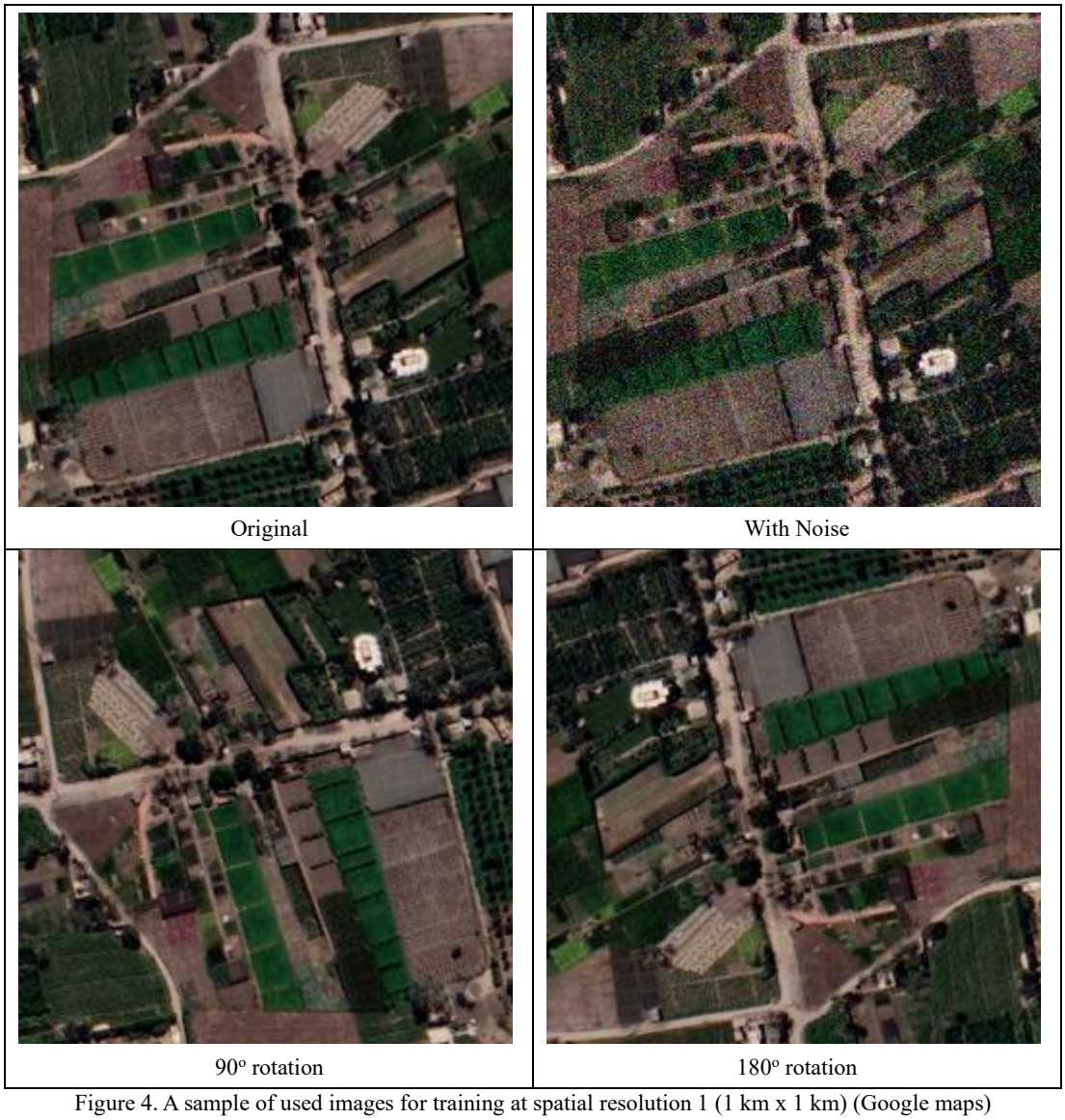

Figure 4. A sample of used images for training at spatial resolution 1 (1 km x 1 km) (Google maps)

For the second spatial resolution, each image represents a class which we would like our network to identify.

This sums us to 86×73 = 6278 classes. The same procedure was used to produce enough images for training (216 images per class which sums up to a total of 1356048 training images) and for testing (108 images per class which sums up to a total of 678024 testing images).

Original | 90° rotation

Figure 5. A sample of used images for training at spatial resolution 2 (374 m x 374 m) (Google maps)

Training Process

The training process involves feeding the two CNNs (SqueezeNet and GoogLeNet) with labeled images and running the training engine in Matlab deep neural network designer. The computer used is i5 generation 12 with 8 cores, two 12 Gig.
Nvidia 3060 GPUs and 40 Gig. DDR4 ram. Case 1 was run first, and the resulting networks parameters were used as initial guess for training case 2.
Extensive experiments, using different number of images per class for training and validation, were conducted to assess the performance of the system in terms of accuracy, robustness, and speed.

3. RESULTS

For case 1, the neural network can identify the longitude and latitude for given images with high accuracy.
SqueezeNet Training accuracy is 98.4 %, Validation accuracy is 98.4 and the test accuracy is 89.9 %. GoogLeNet training accuracy is100 %, Validation accuracy is 99.7 and the test accuracy is 96.4 %.
The training and validation processes are shown for both networks in Figure 6 and Figure 7, respectively.

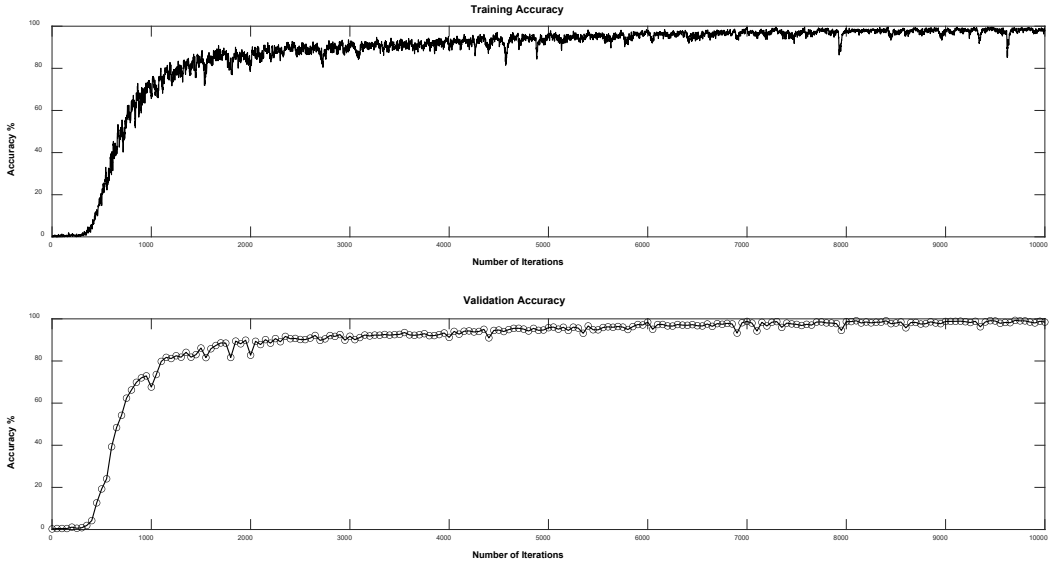

Figure 6. Training and validation accuracy data for SqueezeNet (Case 1)

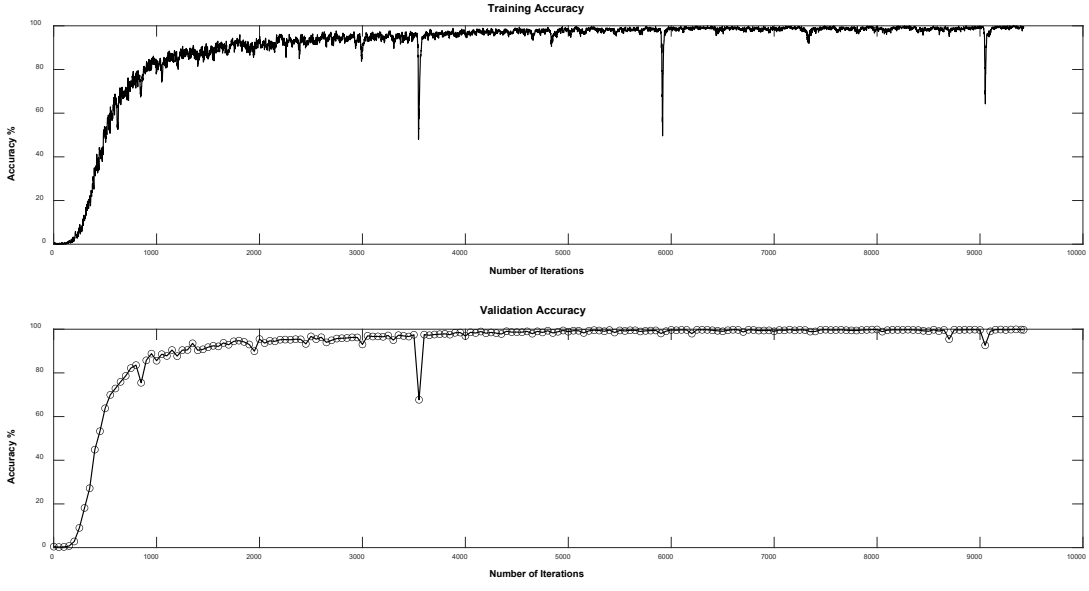

Figure 7. Training and validation accuracy data for GoogLeNet (Case 1)

GoogLeNet gives better accuracy for testing compared to SqueezeNet (96.4 % compared to 89.9) which means better generalization. GoogLeNet is used for training the second spatial resolution images (Case 2). The number of classes is modified from 418 to 6278 classes. With some experiments with the number of training images per class, I was found that the network requires only 48 images per class and the training iterations drops from nearly 10000 iterations in case one to about 1100 iteration. The resulting training accuracy is 98 % and Validation accuracy is 97.4 and the test accuracy is 94.7 %. The training and validation processes are shown in Figure 8.

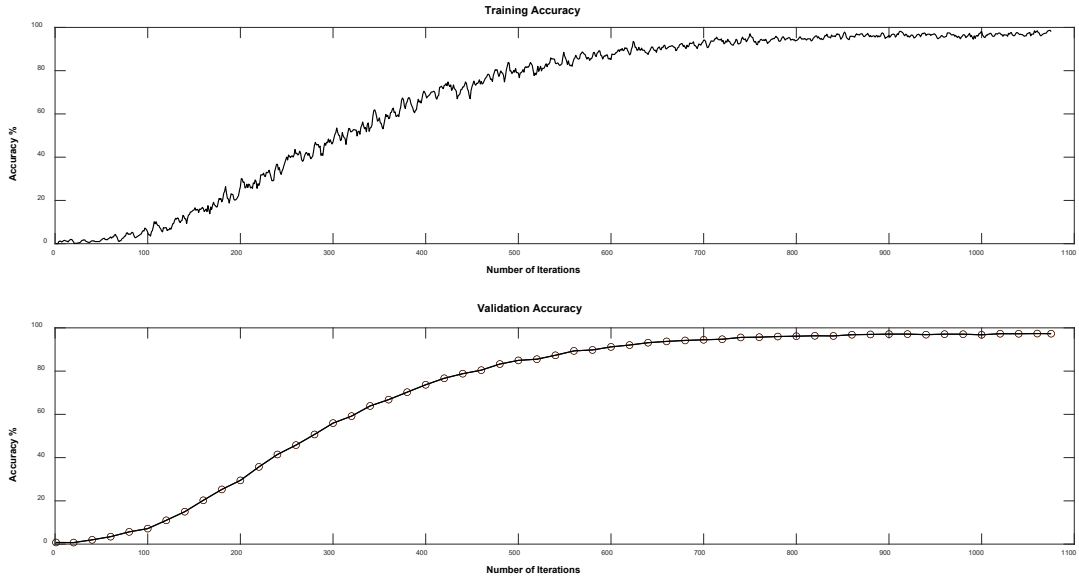

Figure 8. Training and validation accuracy data for GoogLeNet (Case 2)

The results demonstrated the effectiveness and accuracy of the vision-based navigation system in successfully guiding the flying vehicle in GPS-denied environments. It also shows that we can build on trained neural networks to improve spatial accuracy without requiring excessive computational power due to increasing the number of classes. In case 2, although the number of classes increase from 418 to 6278, we could reduce the number of images per class from 216 to 48 and the training iterations from nearly 10000 to about 1100 iterations.

4. CONCLUSIONS

This study demonstrates the potential of a Convolutional Neural Network (CNN) trained on Google Maps imagery to enable accurate image-based, GPS-free navigation. The proposed approach effectively learns and interprets diverse visual cues from map images, allowing it to estimate location with a high degree of accuracy. Specifically, using a GoogLeNet architecture, the system achieved location prediction accuracies of 96.4% and 94.7% for case 1 and case 2, respectively. This translates to a spatial accuracy of approximately 1 kilometer for case 1 and more precise 374 meters for case 2. Importantly, the research suggests that further improvements in accuracy, particularly in reducing the location error, can be achieved by training the CNN with a dataset of more zoomed-in or higher-resolution map images. This refined training process would likely allow the network to detect finer visual details, leading to more precise location predictions. Overall, the findings presented in this paper strongly support the viability of the CNN-based system for reliable GPS-free navigation using only map images. This technology holds significant promise for applications in environments where GPS signals are unreliable or completely absent, offering a crucial alternative for navigation. Furthermore, the practical implications of this research are substantial, as the trained neural network allows for seamless integration onto a flight computer and coupling with an imaging system. This integration paves the way for the development of a robust and effective support/standalone navigation system for various platforms, including unmanned aerial vehicles (UAVs) or other applications where GPS-denied environments are a challenge.

REFERENCES

[1] A. Gagliardo, E. Pollonara, M. Wikelski, Pigeons remember visual landmarks after one release and rely upon them more if they are anosmic, *Anim Behav*, **166**, pp. 85-94, 2020)

[2] Seyed-Yaser Nabavi-Chashmi, Davood Asadi, Karim Ahmadi, Image-based UAV position and velocity estimation using a monocular camera, *Control Engineering Practice*, Volume **134**, 105460, ISSN 0967-0661, https://doi.org/10.1016/j.conengprac.2023.105460, 2023.

[3] Karkar, Abdelghani et al. CamNav: a computer-vision indoor navigation system, *The Journal of Supercomputing*, **7**: 7737-7756, 2021.

[4] Y. Chang & Y. Cheng & U. Manzoor & J. Murray, A review of UAV autonomous navigation in GPS-denied environments, *Robotics and Autonomous Systems*, **170**, 104533. 10.1016/j.robot.2023.104533, 2023.

[5] A. F. Elaraby, A. Hamdy and M. Rehan, A Kinect-Based 3D Object Detection and Recognition System with Enhanced Depth Estimation Algorithm, *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, BC, Canada, pp. 247-252, doi: 10.1109/IEMCON.2018.8615020, 2018.

[6] B. Lewis, & R. Beard, A framework for visual return-to-home capability in GPS-denied environments,. *International Conference on Unmanned Aircraft Systems (ICUAS)*, 633-642. 10.1109/ICUAS.2016.7502608, 2016.

[7] M. Abozied & A. Maher & A. Kamel & M. Sabbagh, Promoting navigation system efficiency during GPS outage via cascaded neural networks: A novel AI based approach, *Mechatronics,* **94**, 10.1016/j.mechatronics.2023.103026, 2023.

[8] Y. LeCun, L. Bottou, Y. Bengio & P. Haffner, Gradient-Based Learning Applied to Document Recognition, *Proceedings of the IEEE*, **86**(11), 2278-2324, 1998.

[9] C. Szegedy et al., Going deeper with convolutions, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, pp. 1-9, doi: 10.1109/CVPR.2015.7298594, 2015.

[10] F. Iandola & M. Moskewicz & K. Ashraf & S. Han & W. Dally & K. Keutzer, *SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and textless1MB model size*, 2016.