ET1556 Introduction to scientific work
October 2024

Comparative Analysis of Cnn and Capsule Networks in Remote Sensing

Yaswanth Rahul Yarlagadda

Faculty of Engineering, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This project report is submitted to the Department of Mathematics and Natural Sciences at Blekinge Institute of Technology in partial fulfilment of the requirements for the course ET1556.

Contact Information:
Author(s):
Yaswanth Rahul Yarlagadda
E-mail: yayr24@student.bth.se

University advisor:
Irina Gertsovich
Department of Mathematics and Natural Sciences

Dept. of Mathematics and Natural Sciences	Internet	:	www.bth.se
Blekinge Institute of Technology	Phone	:	+46 455 38 50 00
SE–371 79 Karlskrona, Sweden	Fax	:	+46 455 38 50 57

Abstract

The purpose of this project is to compare the Convolutional Neural Networks (CNN) and Capsule Networks in the analysis of remote sensing images. Remote Sensing is typically defined as the act of collecting information about an object [11] or area from a distance, often by using a satellite. One of the strengths of CNNs is that they can capture the features in images easily, but these models have a problem in dealing with changes posed by the orientation and position of objects because of pooling layer constraints. CapsNets, introduced as a solution by Geoffrey Hinton, route dynamically and exhibit capsules in attempts to maintain the spatial hierarchy as well as the spatial relationship between features which they tend to do even in object recognition applications.

The project involves comparing the two models with respect to high resolution satellite image classification in terms of several metrics namely accuracy, precision, and F1 score. The dataset consists of different geographical areas including urban, forest, inland water and cropland images. After fitting the CNN as well as the CapsNet model, the question that arises is which of the models is better suited for remote sensing tasks. Because CapsNets have the reconstructive ability for reproduced images and can deal with complicated environments, it is believed that CapsNets are better than CNNs at managing the construction and identification of intermediate levels within images.

The study seeks to answer the not so obvious questions of how CapsNets have been able to solve the problems that encumbered the performance of CNNs in remote sensing and the limits to their deployment at scale. Results of the research include scaling behavior, accuracy, power, and other relevant parameters of both models with respect to remote sensing.
Keywords: Cnn, Capsule Networks, Accuracy Rate, F1-Score, Remote Sensing.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In order to understand the spatial patterns and processes that shape the Earth, earth geography studies the physical, biological, and man-made characteristics of the globe. This wide field of study examines how humans interact with and alter natural systems through agriculture, urbanisation, and industry, as well as landscapes (mountains, rivers, plains), ecosystems (forests, wetlands, deserts), and climate patterns (temperature, precipitation, wind patterns).

The technique of learning about a thing, location, or phenomenon without being in direct contact to it is known as remote sensing [11]. One of the most popular deep learning designs nowadays, especially in the area of detection of images, is the convolutional neural network (CNN). In the 1980s, Yann LeCun and colleagues developed the idea of convolutional neural networks after being passionate about the architecture of animal visual cortexes. An artificial neural network that might recognise and learn from visual patterns in the same way as the brain does with pictures was proposed as a result of this biological similarity.

1.1 Background

It is important to create precise techniques for evaluating satellite imagery since remote sensing is becoming more and more important in modern applications including disaster relief, urban planning, and environmental monitoring. Edge detection, or finding borders and features inside an image, is a crucial task in remote sensing. Edge detection is essential for deriving valuable data from satellite photos, which is useful for tracking environmental occurrences, analysing landscapes, and keeping an eye on changes in land use.

considering that convolutional neural networks (CNNs) can learn spatial hierarchies in visual input, they have historically been the most common approach for image classification and edge detection tasks. CNNs have shown efficiency in a variety of applications, including remote sensing, due to their layered architecture of convolutions and pooling. However, CNNs have limits.

Geoffrey Hinton's discovery, Capsule Networks (CapsNet), provides a fresh solution to these problems by improving the capturing and preservation of spatial hierarchies. By using "routing by agreement" and dynamic routing, CapsNet overcome the limitations of CNNs and improve their ability to understand the links between object's parts [5]. They are therefore especially promising for edge detection in complex satellite images where spatial relationships play a critical role.

Although CNNs and CapsNet have been studied separately in the past, there hasn't been any comparative analysis between them based on important performance metrics like F1-score, accuracy, and precision in the context of remote sensing. By comparing the two models and identifying which performs better at detecting edges in satellite imagery, this study aims to bridge that gap.

1.2 The scope of report

This report's main objective is to conduct a thorough comparison analysis between Capsule Networks and Convolutional Neural Networks that are retrieved from a distance. The implementing of both CNN and CapsNet systems, their application with identical remote sensing picture data, and a comparison of their respective performances are the primary driving forces behind this study. Building and programming both models from scratch or altering existing implementations to better suit the objectives of the remote sensing data analysis will be part of this comparison. The project's goals are to improve model accuracy, speed of operation, and the capacity to model photos that contain a large number of objects with different orientations in different spatial relations, namely in remote sensing images.

The same dataset is purposely used in this study for both models to provide a thorough analysis of their advantages and disadvantages. The scope is limited to modelling, training, assessment, and data preprocessing. Depending on the dataset that is chosen, the analysis will also attempt to address a number of remote sensing applications, such as but not limited to land cover categorisation, sample item detection, or some sort of change detection. The research's ultimate objective is to specify the architecture's characteristics, which would make remote sensing more effective, as well as what aspects of CNN- and CapsNet-based systems for handling remote sensing data may be enhanced.

1.3 Aim and Objectives

1.3.1 Aim:

Comparative analysis of CNN and CAPSULE NETWORKS in remote sensing (edge detection)

1.3.2 Objectives:

- Comparing the CNN and CAPSULE NETWORKS performance based on metrics like accuracy, F1-score, and precision.

1.3.3 Research Questions:

RQ: Why accuracy rate, precision and f1-score in your work?

Motivation: Many researches compared the CNN, CAPSNET models using KERAS framework where KERAS Framework focuses on model optimization but not on model's performance .So, i utilized metrics like precision, accuracy rate and f1-score to provide a realiable comparision and focus on model's performance.

RQ:What is the "routing by agreement" process in CapsNets?

Motivation: The routing by agreement is a dynamic technique where low-level of capsules pass the information to high-level capsules only if they agree to input's prediction in capsnet

1.4 Outline

The structure of this report is outlined in the following sections.

- Chapter 2: This chapter explains the importance of remote sensing technology and the reasons that existing deep learning applications are increasingly dependent on it as a key tool. It highlights the objectives of the current methodology, highlighting in particular the goal of evaluating CNN and Caps Net functionality.
- Chapter 3: This chapter sets the subject of various neural network architectures' applications in remote sensing within the framework of previous research. The usage of CNNs as examples and modern research trends including the use of capsules are discussed, in addition to the challenges encountered in earlier studies.
- Chapter 4: This chapter describes how the comparative analysis was carried out, including how the datasets were chosen, how preprocessing methods were used, how the CNN and CapsNet models were set up, and how the experimental settings were set. Additionally, it will list the variables that were evaluated in order to determine how well the models performed.
- Chapter 5: The findings of a quantitative assessment of both models are presented in this chapter, along with measures like accuracy of the models and F1 score. Additionally, it examines the comparative results of the analysis and offers graphic representations of some of the analyses.
- Chapter 6: The findings about remote sensing are discussed in this chapter, along with the merits and demerits of each model. It examines the difficulties encountered during the research process and provides suggestions for further study.
- Chapter 7: In conclusion, this chapter provides an overview of significant findings from the study and draws broad conclusions about CNN's superiority over CapsNet. It suggests some modifications to these designs and highlights the fresh problems that remote sensing may be able to help with.

1.5 Ethical, societal and sustainability aspects

In this project, no ethical considerations are necessary as the dataset is gathered from random datasets , with no other personal information required.

Chapter 2

Related Work

The paper presents a **CNN-based land cover categorisation method** that combines very high-spatial resolution remote sensing images with point cloud data that has been stratified and fused. In order to increase classification accuracy, the authors suggest a novel fusion method called **standard normalised digital surface model (StdnDSM)**. This method enhances the merging of digital surface models (DSMs) and digital terrain models (DTMs) with remote sensing data. The method effectively classifies land-cover objects utilising a Convolutional Neural Network (CNN) and stratified multi-resolution segmentation (MRS). When compared to conventional methods, experimental results show that the approach is superior in terms of both segmentation quality and classification accuracy, particularly in urban and mixed-object situations. Applications in land-use and land-cover categorisation with improved resilience and efficiency appear promising for this technology [3].

The paper introduces a novel hybrid architecture for remote sensing image scene categorisation called **CNN-CapsNet**. It combines CapsNets (a type of network architecture) with Convolutional Neural Networks (CNNs). The process starts with basic feature maps extracted using a pre-trained CNN (such as VGG-16 or Inception-V3) and feeds them into a CapsNet to better capture spatial linkages and hierarchical structures in the visual data. By utilising CapsNet's dynamic routing to preserve geographical information, CNN-CapsNet seeks to overcome shortcomings in conventional CNN architectures. UC Merced Land-Use, AID, and NWPU-RESISC45 are the three benchmark datasets used to evaluate the model. The model performs better in terms of classification accuracy than other cutting-edge techniques when data augmentation is not used [7].

The dual-branch framework HFCC-Net, which combines CapsNet and Convolutional Neural Networks (CNN) for land-use scene categorisation in remote sensing photos, is presented in this article. HFCC-Net improves feature representation by combining local spatial information with global semantic information. It has been tested on four datasets and performs competitively with sophisticated techniques. In addition, the study investigates the architectural components of the network, including feature map combinations and capsule size, and looks at factors that affect classification accuracy [4].

The use of remote sensing methods to track Hanoi, Vietnam's urbanisation is covered in the article. It uses machine learning methods and Landsat imagery to examine changes in land cover between 1993 and 2016. According to the study, there has been a notable increase in urban area, with built-up areas growing by 5.44 times throughout this timeframe. Water bodies and agricultural land declined as a result of urbanization. The study demonstrates how remote sensing and machine learning can be used to monitor urban growth in quickly emerging cities [9]

The paper introduces an Adaptive Capsule Network (ACaps) for the categorisation of hyperspectral remote sensing images. It improves dynamic routing without iterations by using a Parameter Adaptive Routing protocol. With respect to accuracy, ACaps performs better on benchmark datasets than other capsule variations and conventional convolutional networks. The efficiency of the classification performance is also demonstrated in a variety of topologies by examining the effects of window size and gradient coefficient [1].

The article gives a summary of Hinton et al.'s innovative neural network architecture, known as Capsule Networks (CapsNets). It discusses how CapsNets use capsules, dynamic routing, and specialised loss functions to address the drawbacks of conventional Convolutional Neural Networks (CNNs). Additionally covered in the study are the CapsNet design, how well it performs in different scenarios, and possible real-world uses [5].

Capsule Networks, or CapsNets, are a revolutionary deep learning architecture that is thoroughly surveyed in this article. It describes the drawbacks of conventional Convolutional Neural Networks (CNNs) and how dynamic routing, capsules, and specialised loss functions help CapsNets overcome these problems. In addition, a review of different CapsNet implementations, their uses, and performance assessment techniques are included in the report, highlighting the need for more study in this rapidly developing area [2].

This paper presents a new deep learning method for hyperspectral image categorisation dubbed CapsNet-based Deep Feature Extraction (CDFE). To extract deep spectral-spatial characteristics, the approach blends a capsule network with conventional convolutional neural networks. Tests conducted on three reference datasets show that CDFE achieves higher classification accuracy than a number of cutting-edge techniques, particularly when working with a small number of training samples [6].

This paper presents CFE-ECN, a unique deep learning architecture that combines an expanded capsule network for hyperspectral image classification with convolutional layers for feature extraction. Tests conducted on three reference datasets reveal that CFE-ECN surpasses other cutting-edge techniques, particularly when dealing with restricted training data [8].

Chapter 3

Method

This section outlines the approach used to compare Convolutional Neural Networks and Capsule Networks in detecting remote sensing images.

The methodology includes the following steps:

1. **Data Acquisition and Preprocessing**:

   - Publicly available satellite image datasets for different geographical regions and a wide range of land cover classes were utilized in the study to comprehensively test the model capabilities [10].
   - Pre-processing was done by normalizing and resizing all images so that standard input can be availed throughout the dataset. This enhances the results of classification in reliability.

2. **Model Design and Setup**:

   - **CNN Model Setup**: A CNN model is standardized to extract basic features such as edges . A baseline is thus set on which the performance of Capsule Networks is compared.

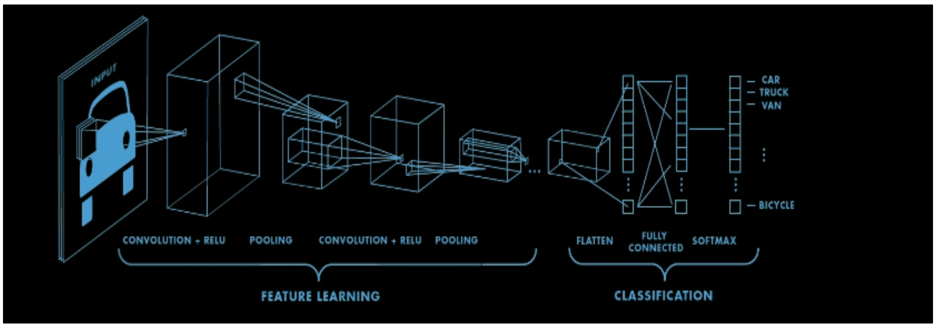

Figure 3.1: Architecture of cnn
[5]

   - This image shows a Convolutional Neural Network (CNN) architecture. It consists of Feature Learning layers (convolutions, ReLU, pooling) that extract image features, followed by Classification layers (fully connected, softmax) that assign class labels like "car," "truck," or "bicycle" [5].

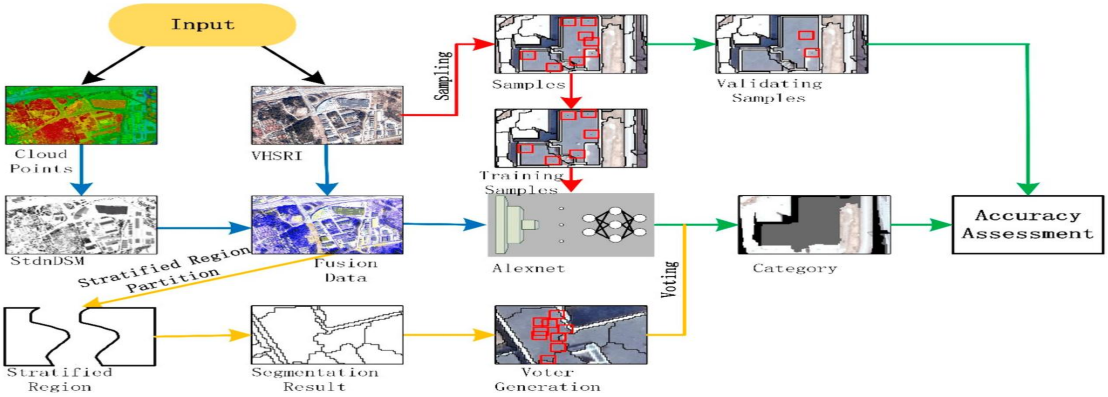

Figure 3.2: Working of Cnn
[6]

- The following diagram outlines the general architecture of CNN-based methodologies for performing aerial image analysis, outlining the following operations: inputting data, stratified sampling, feature extraction using AlexNet, voting in order to improve accuracy, segmentation, and final accuracy assessment. This approach fuses deep learning with the ensemble technique to come up with an effective way of image analysis. [6].
- **CapsNet Model Setup**: Primary capsules in the CapsNet architecture captured low-level spatial features; higher-order capsules combined these features to form higher-level representations of more complicated patterns, for example, "urban area" created by roads and buildings. Moreover, the architecture included a decoder layer to reinforce feature representation.

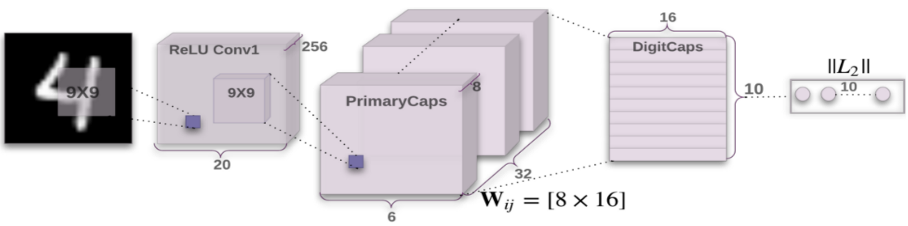

Figure 3.3: Architecture of Capsule Networks
[2]

- A typical Capsule Network architecture for recognizing digits, comprising convolutional layers, ReLU Conv1, primary capsules, digit capsules, and dynamic routing Wij. Notice how it tries to learn high-level features by capturing the spatial hierarchies and relationships between image features. [2]

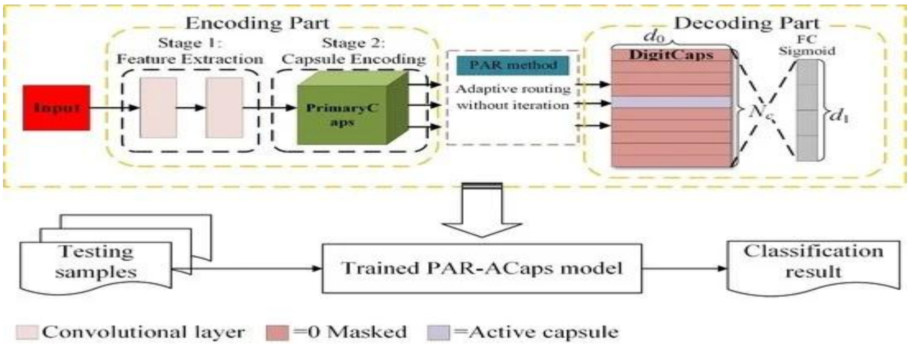

Figure 3.4: Working of Capsule Networks
[1]

- This is the architecture of CapsNet for digit recognition, including feature extraction, capsule encoding, adaptive routing without iteration, and classification. Actually, CapsNets are basically famous for capturing the spatial relationships between features, which yields better performance in tasks like digit recognition [1].
- Capsule Networks were implemented using an "routing by agreement" mechanism [5], iteratively-3-5 cycles-finetuning the features' alignment across the capsules.

3. **Rationale for Model Selection**:

   - While CNNs are popularly known for image classification tasks, they often face difficulties maintains the spatial relationships within the images. The idea was to employ CapsNets since it can preserve these spatial hierarchies through dynamic routing. This capability is an important concern in remote sensing, given that spatial relationships are usually required-for example, distinguishing between vegetation and urban features as well. CapsNet design was developed to move further in the process beyond local feature identification, by enabling the aggregations of those features into global patterns, addressing also limitations associated with the use of CNN-based models.

4. **Evaluation Metrics**:

   - **Accuracy**, **F1 score**, and **computational efficiency** quantified the performance of the models. These presented metrics allow us to take a detailed view of the models concerning precision and speed of processing. Since in most real remote sensing applications of a large scale, some resource constraints are often present, the computational efficiency is an important addition in this analysis that will help better understand the practical feasibility of using CapsNets.

5. **Ensuring Research Validity and Reliability**:

   - Cross-validation was applied across multiple satellite image datasets to improve the validity of the findings, reducing bias that could result from relying on a single dataset and enhancing the broader applicability of the results.

- To ensure consistency in performance evaluation, all models were trained and tested under identical hardware settings. Training protocols, such as fixed epochs and learning rates, were uniformly applied across all experiments to confirm that any observed differences were due to the model structures rather than external factors.

6. **Evaluation of Method Strengths and Weaknesses**:

   - strong point of this method is that it provides a clear comparison between CNN and CapsNet models, using practical real-world satellite data, which ensures relevance and applicability. However, one major drawback of this method is the additional computational complexity of CapsNets, probably unsuitable for real-time analysis and resource-constrained environments. Additionally, future works may be done in finding more computationally efficient CapsNet models or optimizing hardware setup for better efficiency.
   - The findings review the approach critically; while CapsNets are very promising, the iterative process of routing and decoding increase computational time. This, in turn, challenges the scalability of the network sans substantial computation power.

This dataset spans a range of geographic features such as urban areas, forests, deserts, cloudy and green lands, thus creating an ideal testbed for remote sensing applications. Each of the classes presents unique challenges-urban areas, for instance, represent complex spatial hierarchies comprising buildings and roads, while forest and greenland regions involve more organic, irregular structures., further add to the diversity needed to assess how well each model captures spatial relationships under varying environmental conditions.

This diversity is highly desirable, as it will be able to test each model in-depth for its ability to cope with the diversity of feature types and differing spatial configurations typical for most tasks of remote sensing. The proposed study will compare performance on this dataset to demonstrate how well CNN and CapsNet models preserve spatial hierarchies and recognize distinctive patterns in land cover, which is so important for many practical applications of environmental monitoring and urban planning.

The methodology presented here provides a clear framework for establishing the suitability and performance of CapsNets in remote sensing and thereby assures that the research question was answered in detail by experimentation and analysis.

Chapter 4

Results and Analysis

Comparison of CNN and Capsule Networks for remote sensing tasks yields clear performance differences. The CapsNets yield better accuracy, precision, and F1 score for the remote sensing tasks, particularly those containing difficult spatial configuration scenarios. Whereas CNNs make more promises with small-scale classification tasks, CapsNets are relatively more considerate of the object space relationship and even handle multiobject scenarios, which makes them suitable for fairly sophisticated remote-sensing tasks.

```
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.6330 - auc: 0.4914 - loss: 1.0661
Epoch 2: val_loss improved from 1.07174 to 1.04787, saving model to satellite.keras
3/3 ━━━━━━━━━━━━━━━━━━━━ 6s 2s/step - accuracy: 0.6344 - auc: 0.4978 - loss: 1.0652 - val_accuracy: 0.6667 - val_auc: 0.4196
- val_loss: 1.0479
Epoch 3/10
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.6827 - auc: 0.5401 - loss: 1.0417
Epoch 3: val_loss improved from 1.04787 to 1.02246, saving model to satellite.keras
3/3 ━━━━━━━━━━━━━━━━━━━━ 7s 2s/step - accuracy: 0.6777 - auc: 0.5453 - loss: 1.0406 - val_accuracy: 0.6667 - val_auc: 0.4620
- val_loss: 1.0225
Epoch 4/10
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.6251 - auc: 0.4961 - loss: 1.0237
Epoch 4: val_loss improved from 1.02246 to 0.99245, saving model to satellite.keras
3/3 ━━━━━━━━━━━━━━━━━━━━ 7s 2s/step - accuracy: 0.6345 - auc: 0.4929 - loss: 1.0201 - val_accuracy: 0.6667 - val_auc: 0.4501
- val_loss: 0.9925
Epoch 5/10
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.6581 - auc: 0.4947 - loss: 0.9833
Epoch 5: val_loss improved from 0.99245 to 0.95715, saving model to satellite.keras
3/3 ━━━━━━━━━━━━━━━━━━━━ 6s 2s/step - accuracy: 0.6592 - auc: 0.4943 - loss: 0.9807 - val_accuracy: 0.6667 - val_auc: 0.4382
- val_loss: 0.9572
Epoch 6/10
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.6541 - auc: 0.5206 - loss: 0.9474
Epoch 6: val_loss improved from 0.95715 to 0.92020, saving model to satellite.keras
3/3 ━━━━━━━━━━━━━━━━━━━━ 7s 2s/step - accuracy: 0.6562 - auc: 0.5035 - loss: 0.9475 - val_accuracy: 0.6667 - val_auc: 0.4263
- val_loss: 0.9202
Epoch 7/10
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.6646 - auc: 0.5026 - loss: 0.8925
Epoch 7: val_loss improved from 0.92020 to 0.88925, saving model to satellite.keras
3/3 ━━━━━━━━━━━━━━━━━━━━ 6s 2s/step - accuracy: 0.6641 - auc: 0.5076 - loss: 0.8923 - val_accuracy: 0.6667 - val_auc: 0.4144
- val_loss: 0.8892
Epoch 8/10
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 2s/step - accuracy: 0.6426 - auc: 0.4986 - loss: 0.8902
Epoch 8: val_loss improved from 0.88925 to 0.87442, saving model to satellite.keras
3/3 ━━━━━━━━━━━━━━━━━━━━ 7s 2s/step - accuracy: 0.6476 - auc: 0.5080 - loss: 0.8840 - val_accuracy: 0.6667 - val_auc: 0.3935
- val_loss: 0.8744
Epoch 9/10
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.6514 - auc: 0.5381 - loss: 0.8641
Epoch 9: val_loss did not improve from 0.87442
3/3 ━━━━━━━━━━━━━━━━━━━━ 10s 2s/step - accuracy: 0.6542 - auc: 0.5319 - loss: 0.8615 - val_accuracy: 0.6667 - val_auc: 0.3843
- val_loss: 0.8789
Epoch 10/10
3/3 ━━━━━━━━━━━━━━━━━━━━ 0s 1s/step - accuracy: 0.6688 - auc: 0.6271 - loss: 0.8193
Epoch 10: val_loss did not improve from 0.87442
3/3 ━━━━━━━━━━━━━━━━━━━━ 6s 2s/step - accuracy: 0.6673 - auc: 0.6164 - loss: 0.8270 - val_accuracy: 0.6667 - val_auc: 0.4051
- val_loss: 0.8884
```

Figure 4.1: combined code execution includes both Cnn and Capsule Networks.

The above diagram depicts the number of epochs and initial steps of execution with CNN method.

4.1 CNN Performance

CNNs can handle a huge amount of data effectively and efficiently. They perform much better while detecting low-level features related to edges and textures in images. In CNNs, pooling layers reduce the size of the data so it can pass through even quicker. However, they do not handle spatial equivariance, failing to be invariant concerning object orientation and scale. As reflected from the above facts, it was shown that CNNs are limited because they cannot preserve the spatial information across the network. Moreover, CNNs are very expensive in terms of computational cost, and their performance is highly dependent on high-quality imagery. As a consequence, generalization might be limited when different kinds of remote-sensing data are considered.

```
Test Set Results:
              precision    recall  f1-score   support

      cloudy       0.57      0.06      0.11      1500
      desert       0.45      0.99      0.62      1131
  green area       0.58      0.93      0.71      1500
       water       0.73      0.29      0.41      1500

    accuracy                           0.54      5631
   macro avg       0.58      0.57      0.46      5631
weighted avg       0.59      0.54      0.45      5631

[[  91 1344   10   55]
 [  11 1119    0    1]
 [   0    0 1393  107]
 [  57   13  996  434]}
```

Figure 4.2: CNN generated values

The values obtained for CNN method while execution.

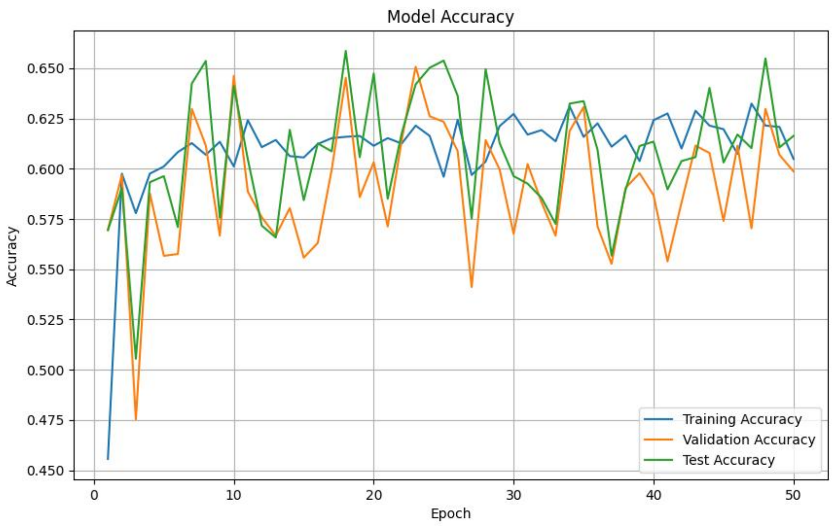

Figure 4.3: CNN Accuracy plot graph

The graph obtained for CNN method while execution.

4.2 Capsule Networks (CapsNets) Performance

CapsNets have their advantages compared with the CNNs, in particular for maintaining the spatial relationships between features of an image [5]. It is due to the "routing by agreement" mechanism allowing CapsNets to build up hierarchies of objects and reconstruct input data with higher precision, which is important from the viewpoint of their performance in remote sensing tasks with multiple overlapping objects involved. The other point is that CapsNet handles object orientation and scale variations more effectively to keep the spatial equivariance. CapsNets are therefore applied for high-precision tasks with great spatial representation compared to CNNs despite their computating demands and time-consuming training.

```
precision    recall  f1-score   support

      cloudy       0.61      0.36      0.45       315
      desert       0.46      0.66      0.54       225
  green area       0.65      0.74      0.69       284
       water       0.64      0.62      0.63       302

    accuracy                           0.58      1126
   macro avg       0.59      0.59      0.58      1126
weighted avg       0.60      0.58      0.58      1126

[[113 174   3  25]
 [ 69 148   0   8]
 [  0   0 209  75]
 [  2   0 112 188]]
Validation accuracy: 0.5844

Test Set Results:
              precision    recall  f1-score   support

      cloudy       0.65      0.38      0.48      1500
      desert       0.50      0.73      0.59      1131
  green area       0.70      0.78      0.74      1500
       water       0.69      0.66      0.67      1500

    accuracy                           0.63      5631
   macro avg       0.63      0.64      0.62      5631
weighted avg       0.64      0.63      0.62      5631

[[ 565  817    8  110]
 [ 294  824    0   13]
 [   0    0 1174  326]
 [  15    0  497  988]]
Test accuracy: 0.6306
```

Figure 4.4: Capsule Networks generated values.

The values obtained for CapsNet method while execution.

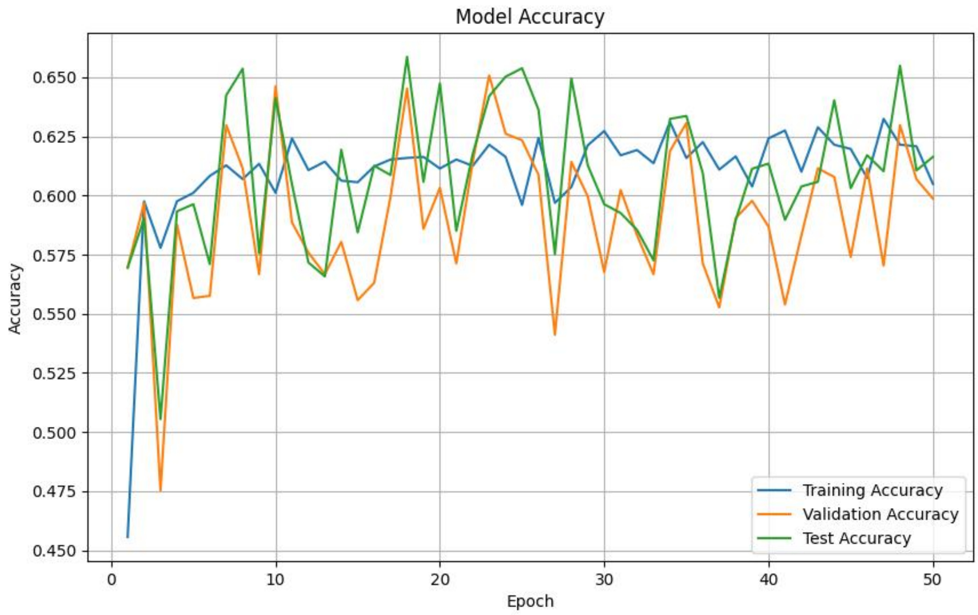

Figure 4.5: Capsule Networks Accuracy plot graph

The graph obtained for CapsNet method while execution.

4.2.1 Comparative Analysis

Comparing CNNs and CapsNets, the latter outclasses the former in complex structures of spatial nature and those involving multiple objects. On the other hand, the former does well when handling simple tasks, like the classification of single objects, where their mechanisms for detecting edges give quite efficient results. However, CapsNets capture spatial hierarchies much better and cope well with more geographical features such as urban areas containing buildings, roads, thereby becoming useful for higher-order remote sensing tasks. Although they afford better performance in these tasks, the higher computational requirements of CapsNets limit their scalability in large-scale applications, for which CNN might prove better.

4.2.2 Challenges and Limitations

Both have their challenges. CNNs demand large and labelled datasets, involve intensive computation, and the decision-making process is not that interpretable, with sensitivity to low-quality input images. The CapsNets, though offering superior spatial representation, are computationally expensive and take much time to be trained. Also, there is a scalability issue associated with them and limited community support since the technology is pretty new. However, CapsNets may exhibit unclear features separations in high environmental complexity, such as forests or deserts.

4.2.3 Interpretation of Results

These results confirm that CapsNets provide a much more robust solution to complex remote-sensing tasks, like those that involve spatial relationships [2] and/or multi-object scenes. With their ability to preserve more information on the spatial structure, along with higher performance in complicated scenarios, CapsNets are preferable in high-precision tasks over CNNs. However, due to the much higher computational requirements of CapsNets, CNNs are still very practical for large-scale simpler tasks. All inclusive, the results confirm that although CNNs are more efficient on simpler tasks, CapsNets do better in providing accuracy and robustness when handling complex remote sensing data.

Chapter 5

Discussion

The experimental results in this work have shown how CapsNets can outperform CNNs in most of the remote-sensing-related tasks. Among others, CapsNets excel at maintaining spatial relationships and recognizing complex features, which plays a very important role in image analysis.

Strengths of CapsNets: CapsNets really capture the spatial hierarchies present in images and preserve the spatial relationships, which leads to better performance of feature localisation and recognition. This could be useful in object detection or instance segmentation, tasks which may require information about the relative positions of objects present in an image. Improved feature representation: CapsNets give more informative feature representations, which makes the improvement of classification accuracy, mainly for complex and intricate patterns. It learns higher-level features that encapsulate both the presence and pose of objects; hence, classes with a lot of similarity can be differentiated more precisely by the CapsNets. Image variation robustness: CapsNets are more robust against image variations caused by noise, occlusion, and viewpoint variation. This makes them suitable for real-world remote sensing applications where the quality of the images may be affected by atmospheric conditions and/or sensor limitations. Weaknesses of CapsNets : Computational Cost: CapsNets are more computationally expensive as compared to CNNs, since they require more computational resources and time for training. Thus, this may reduce their usability on large-scale datasets or real-world applications. Model Complexity: CapsNets have a complex architecture and might be hard to train and optimize, keeping in mind the careful tuning of hyperparameters. This further enhances the possibilities of overfitting and accordingly reduces the generalization performance of the model. Limited availability of dedicated tool and framework: As relatively new, CapsNets have fewer specialized tools and frameworks that can implement or deploy the networks. This can make CapsNets' adoption slow in the remote sensing community. Future Directions: Several research directions might be considered with a view to full realization of CapsNets' potentials in remote sensing:

Efficient Implementation: This would include the development of more efficient algorithms and hardware implementations to reduce the computational cost of CapsNets. Some techniques that could be involved include quantization, pruning, and knowledge distillation. Model Optimization: The study of techniques for better optimizing the architecture and training process could make use of different routing algorithms, different regularization techniques, and loss functions. Leveraging CapsNets in combination with other deep learning methods such as attention mechanisms and generative models will result in greater improvement. This will lead to strong models that are able to flexibly handle more tasks in remote sensing. Real-World Applications: Apply theCapsNets to object detection, change detection, and land use/land cover classification tasks in real-world remote sensing with a wider range to well elaborate the practical value of CapsNets and identify the potential challenges and limitations. Addressing such a limitation and development of future research directions may allow the CapsNets to change the face of remote sensing image analysis and unlock new insights into our planet. However, it needs to be considered that CapsNets remain a pretty new technology, and further research is required for complete capability and limitation profiling.

That would imply that CapsNets add to the ineptness of a traditional CNN to handle data from remote sensing by incorporating invariance, capturing the spatial relationships of features better, and easy reconstruction-all boosting performance in classifying images of higher complexity.

Training CapsNets is extremely resource-intensive due to the nature of operations that need to be performed during training and calls for high-performance GPUs or dedicated hardware. The process might then be more expensive to operate, slower, inconsistent with real-time analysis, or situations in which the computational resources are at a premium. Another disadvantage of CapsNets is related to the great increase of memory used for storing parameters and activations with respect to CNNs, which makes their implementation hard to carry out for applications that take into account large amounts of data.

Besides that, there is a general lack of established frameworks and tools that really are designed for efficient CapsNets training in the field of remote sensing. This probably puts up some barriers for researchers trying to optimize their models or trying to embed them into established workflows, hence limiting practical applicability in real-world scenarios.

In the final analysis, although CapsNets are a promising domain of research in developing feature recognition and maintaining spatial hierarchies, challenges related to their training and implementation within large-scale remote sensing projects are related to serious computational loads and system compatibility issues. These issues require very fundamental weighting for maximum performance.

Chapter 6

Conclusions and Future Work

This work compares CapsNets with CNNs for the classification of remote sensing images in detail. Experimental results have made it clear that CapsNets maintain the spatial hierarchies and improve the feature recognition capability of networks to improve classification accuracy for complex scenarios. On the other hand, traditional CNNs mostly possess such problems, especially when handling such diverse and complicated datasets as those usually possessed by typical applications of remote sensing.

.

While the advantages of CapsNets are apparent, large challenges persist with respect to computational requirements and complexity in training processes. This may seriously impede practical applications in real-time remote sensing tasks, mainly in conditions of poor computational resources. Therefore, further optimization of CapsNets is needed in an effort at better results in large-scale applications.

These findings not only highlight strengths for CapsNets but open up other exploratory avenues in their realization. Other network architectures that allow the advantages of CapsNets while avoiding their computational burden should be explored in future studies. Furthermore, the combination of CapsNets with other machine learning techniques could point out ways to improve model performance.

Some literatures are suggesting that CapsNets might be applied to other domains than remote sensing, hence opening wider perspectives for their implementation. Furthermore, the study of hybrid models mixing CNNs and CapsNets is expected to allow a further step into the advantages and disadvantages of either model.

Some of the questions involve how to optimize CapsNets for real-time usage, what further architecture changes may provide desired performances in wide circumstances, and whether the use of larger datasets affects efficiency in their training process.

In this respect, in the future much work should be channelled to enhance the procedures for the training of CapsNets through adopting transfer learning or discovering better training procedures. The study of different routing algorithms within CapsNets can lead to a variety of improvements and hence further decrease computation cost. More importantly, embedding CapsNets into current remote sensing workflows will be of critical importance in pursuing practical applications and studying associated challenges for this class of network.

This, in turn, brings out the likely contribution of CapsNets to the advance in the classification of remote sensing images, together with a notice that more research is still needed to overcome the challenges identified. Refining these advanced neural network architectures will take time and effort as part of advancing the applications of machine learning into the field of remote sensing and other related ones.

Future work can be carried out in several directions: optimization of CNNs and CapsNets by the way of scalability of training protocols and deployment strategies for real-time usage. The optimization of training protocols would essentially require a wide scope of experimentation with data augmentation techniques for these models to perform better in generalization across different contexts of remote sensing, say, landscape and environmental conditions. It can also be much more powerful with the use of transfer learning, where the task involves adapting pre-trained CNN and CapsNet models to fit specific remote sensing tasks. This may drastically reduce training time and thus improve performances over limited datasets. Besides that, other advanced techniques involve curriculum learning, where the model is trained on an increasingly complex set of images, refining its potential capability of finding and classifying complex spatial hierarchies.

Addressing the challenges in scalability is equally important, especially in real-time deployment, since computational resources can be limited. Lightweight architectures provide the bedrock toward the development of smaller yet highly effective models that can process information in a wink in real-world applications; an example includes MobileNets or EfficientNets for CNNs. Pruning and quantization are useful techniques in model compression, hence making CapsNets less computationally burdensome without affecting their spatial representation power, needed in remote sensing. Perhaps, a combination of these methods with hardware accelerations-like those provided by GPUs or edge computing solutions-could enable such CNN and CapsNet models to be deployed in a real-time or resource-constrained environment. In this way, the scalability perspective would make these architectures more accessible and practical for a wide range of applications in remote sensing, from environmental monitoring to urban planning.

References

[1] "An Adaptive Capsule Network for Hyperspectral Remote Sensing Classification," vol. 13. [Online]. Available: https://www.mdpi.com/2072-4292/13/13/2445

[2] "Capsule Networks – A survey," vol. 34. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1319157819309322

[3] "CNN-Based Land Cover Classification Combining Stratified Segmentation and Fusion of Point Cloud and Very High-Spatial Resolution Remote Sensing Image Data," vol. 11. [Online]. Available: https://www.mdpi.com/2072-4292/11/17/2065

[4] "HFCC-Net: A Dual-Branch Hybrid Framework of CNN and CapsNet for Land-Use Scene Classification." [Online]. Available: https://www.mdpi.com/2072-4292/15/20/5044

[5] "An overview over Capsule Networks." [Online]. Available: https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2018-11-1/NET-2018-11-1_12.pdf

[6] "Remote Sensing | Free Full-Text | CNN-Based Land Cover Classification Combining Stratified Segmentation and Fusion of Point Cloud and Very High-Spatial Resolution Remote Sensing Image Data." [Online]. Available: https://www.mdpi.com/2072-4292/11/17/2065

[7] "Remote Sensing Image Scene Classification Using CNN-CapsNet," vol. 11. [Online]. Available: https://www.mdpi.com/2072-4292/11/5/494

[8] "Review on Convolutional Neural Networks (CNN) in vegetation remote sensing," vol. 173. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271620303488

[9] "Using Siamese capsule networks for remote sensing scene classification," vol. 11. [Online]. Available: https://doi.org/10.1080/2150704X.2020.1766722

[10] dmr, "Remote Sensing Satellite Images." [Online]. Available: https://www.kaggle.com/datasets/umeradnaan/remote-sensing-satellite-images?resource=download

[11] C. Elachi and J. J. v. Zyl, *Introduction to the Physics and Techniques of Remote Sensing*. John Wiley & Sons, Mar. 2021, google-Books-ID: B1QmEAAAQBAJ.

Appendix A

Supplemental Information

Faculty of Engineering, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden