Remote Sensing Image Classification Using CNN-CapsNet

Carby Wu, Jason Rong, Thomas McBeth, Christian Lewandowski
University of Michigan
{ginwu, jasonryj, tmcbeth, lewandoc}@umich.edu

1. Introduction

The development of earth observation technology has resulted in a rapid increase in the available remote sensing data. The availability of big data sets and the development of novel deep learning architectures for computer vision make remote sensing imagery highly amenable to the application of deep learning techniques. Remote sensing image classification aims to use deep learning techniques to apply semantic labels to images according to their content [2]. This task can be applied urban planning, land resource management, traffic control, and disaster monitoring.

We implemented a CNN-CapsNet architecture to classify remote sensing images based on the work of Zhang, et al. [4]. The paper is structured as follows: we first review related works, outlining the development of remote sensing image classification techniques. We then detail the dataset and the CNN-CapsNet architecture. Finally, we present results and conclude.

2. Related Work

2.1. Baseline CNN Models

Baseline CNNs, such as AlexNet, have shown reliable performance in remote sensing image classification by extracting hierarchical features from raw data, achieving accuracies near 90% on standard datasets.

2.2. Advanced CNN Models

Han et al. introduced AlexNet-SPP-SS, which integrates spatial pyramid pooling (SPP) and side supervision (SS) to enhance classification by leveraging multi-scale spatial information and mitigating overfitting [1]. This approach achieved 96.67% accuracy on the UC Merced dataset, surpassing conventional methods.

2.3. Deep Residual Networks (ResNet)

ResNet, with its skip connections, resolves vanishing gradient issues and enables effective training of deep networks. Zhao et al. applied ResNet-18 to multispectral data, achieving 94.68% accuracy by capturing robust hierarchical features [5].

2.4. Hybrid CNN-SVM Models

Sun et al. proposed a hybrid CNN-SVM model where an SVM classifier replaced the CNN's output layer [3]. This combination improved generalization and classification accuracy, achieving 96.42% on the UC Merced dataset.

3. Method

3.1. VGG-16 Feature Extraction

VGG-16 is employed first to extract spatial features from input images. Fully connected layers are replaced with an identity function to preserve spatial details. Input images of size 256×256 are processed, producing feature maps of $16 \times 16 \times 512$. These maps are used as inputs for the capsule layers, enabling the retention of hierarchical spatial information critical for classification.

3.2. Capsule Layers

The capsule layers consist of two components: the PrimaryCaps layer and the DigitCaps layer. These layers encode features into capsules that represent spatial and hierarchical relationships.

3.2.1 PrimaryCaps

The PrimaryCaps layer processes the feature maps from VGG-16 and converts them into capsules. The input feature maps of size $16 \times 16 \times 512$ are transformed into capsule vectors of size 32×8. This transformation is achieved using an 8×8 convolutional kernel with a stride of 2. A squashing function normalizes the capsule lengths to ensure they lie between 0 and 1, as given by:

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \frac{s_j}{\|s_j\|}, \tag{1}$$

where s_j is the input vector, v_j is the squashed output vector, and $\|s_j\|$ is the norm of s_j.

3.2.2 DigitCaps

The DigitCaps layer aggregates the capsules from the PrimaryCaps layer to form higher-level capsules representing

each class. The input capsules of size 32×8 are transformed into 16-dimensional capsules for each class. The dynamic routing mechanism is used to refine the connections between capsules. The coupling coefficients are calculated as:

$$c_{ij} = \frac{\exp(b_{ij})}{\sum_k \exp(b_{ik})}, \tag{2}$$

where b_{ij} are logits initialized to zero. The predicted vector for a higher-level capsule is computed as:

$$\hat{u}_{j|i} = W_{ij} u_i, \tag{3}$$

where W_{ij} are learnable weights, and u_i is the output from a lower-level capsule. The final capsule output is calculated using the squashing function defined in (1).

3.3. Training Strategy

The model is trained using the Margin Loss, defined as:

$$l_k = T_k \max(0, m^+ - \|v_k\|)^2 + \lambda(1 - T_k) \max(0, \|v_k\| - m^-)^2, \tag{4}$$

where T_k is the ground truth for capsule k, $\|v_k\|$ is the capsule length, m^+ and m^- are predefined thresholds, and λ is a scaling factor for negative losses.

3.4. Implementation Details

The model is trained with a batch size of 32 using the Adam optimizer with a learning rate of 1×10^{-4}. A 50% dropout is applied after the PrimaryCaps layer to prevent overfitting.

4. Experiments

4.1. Dataset

The CNN-CapsNet model is evaluated on the UC Merced Land Use dataset, a benchmark in remote sensing image classification [4]. The dataset consists of 2,100 images across 21 land-use categories, each with 100 images of size 256×256. A 50-50 split is used for training and validation.

4.2. Data Preprocessing and Augmentation

Images are resized to 256×256, normalized using ImageNet statistics, followed by transformations such as random resized cropping, 360-degree rotation, horizontal and vertical flips to enhance generalization. Validation images are resized and normalized without augmentation for consistent evaluation.

4.3. Hyperparameter Selection

Hyperparameters are set following [4]. The routing iteration number is set to 2, and capsule dimensions for PrimaryCaps and DigitCaps are 8 and 16, respectively. The model is trained for 50 epochs with a batch size of 32 using the Adam optimizer at a learning rate of 1×10^{-4}. This training limit balances computational constraints and model performance.

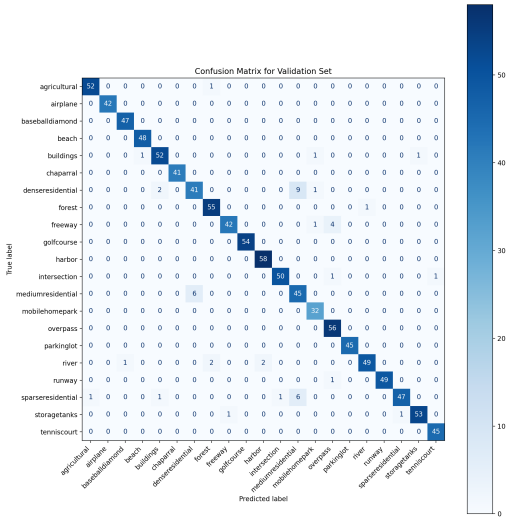

Figure 1. Confusion Matrix for the Validation Set. Each row corresponds to the true class, and each column represents the predicted class. The high diagonal values indicate strong classification performance.

4.4. Results and Discussion

The CNN-CapsNet achieves 95.52% accuracy, outperforming the baseline CNN's typical 90% accuracy. While slightly below state-of-the-art methods like AlexNet-SPP-SS (96.67%), the model demonstrates strong performance and robustness to spatial transformations. Figure 1 includes the confusion matrix. The main misclassification occurs in distinguishing between sparse, medium, and dense residential areas, likely due to their similar visual characteristics.

5. Conclusions

Our CNN-CapsNet model achieves a competitive accuracy of 95.52% on the UC Merced dataset, demonstrating its effectiveness in preserving spatial relationships through the combination of VGG-16 and capsule networks. However, its computational cost presents a limitation for scaling to larger datasets. Future work will explore optimizing computational efficiency and applying the model to more extensive datasets while maintaining its robust classification performance.

References

[1] X. Han et al. Pre-trained alexnet architecture with pyramid pooling and supervision for high spatial resolution remote sensing image scene classification. *Remote Sensing*, 9(8):848, 2017.

[2] H. Song et al. A survey of remote sensing image classification based on cnns. *Big Earth Data*, 2019.

[3] X. Sun et al. Classification for remote sensing data with improved cnn-svm method. *IEEE Access*, 7:164507–164516, 2019.

[4] W. Zhang et al. Remote sensing image scene classification using cnn-capsnet. *Remote Sensing*, 2019.

[5] Y. Zhao et al. Deep learning classification by resnet-18 based on the real spectral dataset from multispectral remote sensing images. *Remote Sensing*, 14(19):4883, 2022.