International Journal of Scientific & Engineering Research, Volume 11, Issue 3, March-2020
ISSN 2229-5518

Classification of Hyperspectral Imagery Data using Capsule Cap Network

Snehal Sarode, Sarika Jadhao, Bhavna Shinde, Rajashree Gadhave

Abstract— **Hyperspectral image (HSI) classification is a function of dividing the class label across the pixels of the captured image using visual sensors. HSI collects and processes information from an electromagnetic microscope. The purpose is to find the capacity of each pixel in the image of the acquisition site, to identify materials. We propose a new CNN based concept on the spectral-spatial capsule networks to achieve accurate HSI classification while significantly reducing the complexity of the network design. Hyperspectral image classification has become one of the most active research fields within the remote sensing community, as it is able to provide relevant data for a variety of land monitoring applications such as natural sciences, precision agriculture and land surveillance services. We are introducing a new deep learning framework which is extension of traditional CNN for separating HSI data cube. This help in making the features revealed by the network more informative, which ultimately helps to reduce the complexity and lead to more accuracy.**

Index Terms— Capsule networks (CapsNets), convolutional neural networks (CNNs), hyperspectral image (HSI), principle component analysis (PCA), pixels, spectral-spatial, data cube**.**

1 Introduction

Hyperspectral imaging (HSI) is widely used in the remote sensing community for the opportunity to create hundreds of visual channels in one place. However, HSI requires strong techniques and accurate segmentation to extract features from the image. HSI classification has been considered a challenging problem mainly due to the complex nature of the image space (i.e., large amounts of data, aggregated pixels and limited training samples), and as a result many efforts have been made to address this issue over the past few decades. In the first phase of HSI classification, spectral domain planners, such as vector support machines (SVMs), random forest (RF) and multinomial (MLR) transcripts were made great improvement in understanding image scenes. Recent technological advances provide promising ways to deal with HSI classification. For example, morphological profiles (MPs), Markov fields (MRFs), and methods that use visual imagery (e.g. sparse models) image scenes using spatial location and content. These approaches aim to differentiate HSI through knowledge and spatial information. For example, the integrated sparse model combines data from several neighbouring pixels of a test pixel, which has been proven as an effective way to improve group performance.

- *Author Snehal Sarode is currently pursuing bachelor's degree program in computer engineering in Mumbai University, India, PH-08355913454. E-mail: snehalsarode77@gmail.com*
- *Co-Author Sarika Jadhao is currently pursuing bachelor's degree program in computer engineering in Mumbai University, India, PH-08082049990. E-mail: sarika8932@gmail.com*
- *Co-Author Bhavna Shinde is currently pursuing bachelor's degree program in computer engineering in Mumbai University, India, PH-08149869520. E-mail: shinde.bhavna11@gmail.com*
- *Co-Author Rajashree Gadhave is faculty of Computer Engineering Department at PHCET, Mumbai University, India, PH-09869829358. E-mail: rgadhave@mes.ac.in*

Recently, deep learning is interesting for researchers in the field of computer visualization. In particular, neural networks (CNNs) have attracted much attention because of their high performance in many domains, such as facial recognition, object detection and video segmentation.

Depending on the feature release, CNNs can read feature presentations with enough blocks to determine. In opposition to the traditional methods of using traditional domain-based methods, CNNs can learn features automatically from original images. In addition, CNNs can be created as an end-to-end framework that can generate precision classification maps. So are most CNN models included in the HSI phase. For example, Chen, et al. used several permutable layers to extract nonlinear and non-invasive elements in raw HSI, and the authors also examined a number of strategies to avoid overlaps in the model. In, a deep CNN was integrated with the dimension reduction method to extract the spectral features of the HSI configuration, and the obtained discriminant features resulted in the highest use of benchmark data. In the CNN structure was exploited to create deep-level automation features to drive the HSI classification function. The same work was reproduced by in which a specific CNN deep layer was introduced to study both visual and spatial features. In various attribute profiles were extracted and added to the raw HSI data as included in the CNN model, which captured the geometric and individual properties of the HSI optimum. CNN's previously used types of HSI scenarios for categories have focused on the automatic removal of visual features and locations. Zhao, et al. investigated how to integrate deep learning features extracted from multiple spatial scales, which improved the performance of HSI classification to some extent. In a partial viewing strategy was used to compose multiple visual input of a specific CNN structure for land use classification. There are some notable examples in which CNNs have been used, such as the discovery of an oil tank, incident isolation and road network removal.

International Journal of Scientific & Engineering Research, Volume 11, Issue 3, March-2020
ISSN 2229-5518

As can be seen from the examples above, many CNN channels view the HSI classification as a function of extracting the high-level dynamic features.

2 EXISTING SYSTEM

In recent years K-means integration algorithm, Gaussian learning method, partial analysis, random forest processing, deep learning techniques and the latest method validation network are used. For the Rand forest study method, it aims to achieve convergent classifier convergence in hyperspectral image analysis but limitations are RF-BHC methods should be applied to several databases with different characteristics to better evaluate their overall performance. The Gaussian procedure also restores the Bayesian framework to be separated. The program explores two ways to analyse Laplace and ways to maximize readiness. The system implements these two methods by using covariance functions such as exponential squid and neural network covariance. The disadvantage is that the required EP dosing time is not extended. This is because the matrix incision is generated during the learning process. In depth-based Hyperspectral Data auto encoder partitioning helps to increase SVM accuracy and heart rate Spectral segmentation always results in sound-free propagation points in the image. Time-lapse features may not affect specific sub-regions. Another disadvantage of SAE-LR is its high-test time. In K-means the dimension reduction method is used to remove the compression but the standard PCA + M- SVM process takes longer to execute. And the accuracy level is low compared to the proposed method. In advanced machine learning techniques, it can compress big data and be categorized in advance. The separation and suppression results obtained in real time are therefore the best advantage for good exploitation.

The method can generate error reconstruction/debugging, a natural error that is not one of the fastest ELM processes can achieve acceptable error without loss of data in a short period of time but the Compression method of the proposed method is smaller than the widely used PCA. In the proposed 3D Convolutional Neural Network, the model is simpler, less likely to overload, and easier to train. The use of visual-spatial information can significantly improve the performance of the hyperspectral image (HSI) category. 3D-CNN HIS classification techniques still need to exploit anonymous samples, as the anonymous samples are much easier to access than the labelled samples. Various existing systems used to date have problems such as: 1. multiple data sets are needed for training purposes to be able to predict accurate results. 2. Due to the complex nature of the time required algorithm integration is high. 3. Spectral segmentation sometimes results in noisy image separation. 4. Characteristics of periodically dominated areas are mistakenly classified into other smaller areas. 5. It takes more time to get killed. Even after that level of accuracy is low. 6. To reduce the size of the images one has to use a compression phase that takes a lot of compaction time and then reduces performance. 7. The convolutional neural network has layers of binding where it loosens the spatial detail and specificity of the object in the image and therefore image segmentation and object detection are negative. Therefore, to overcome all the issues described above we create a model that will work with a sample of data and provide a better classification result. This model stores information about the location of the object and the movement of the object in relation to the location within the image. This model resolves the consistency of the image (wrap image). It is also designed in such a way that it will reduce the complexity of the network problem, shorten compilation time and produce more accurate results.

3 PROPOSED SYSTEM

We proposed two frameworks based on spectral information and spatial information, respectively. The spectral information of each pixel is regarded as an input sample. This architecture is named as 1- D CapsNet, which consists of two convolutional layers and one fully connected layer. The image feature is first obtained through an ordinary convolutional layer, and then the vector-output capsules are constructed by a capsule convolutional layer and are sent to the fully connected layer. However, 1-D CapsNet fails to capture the spatial contextual information. To further improve the classification performance of HSI, it is necessary to consider the spatial information to this end, a 2- D CapsNet is proposed for HSI classification by using both the spatial and spectral information.

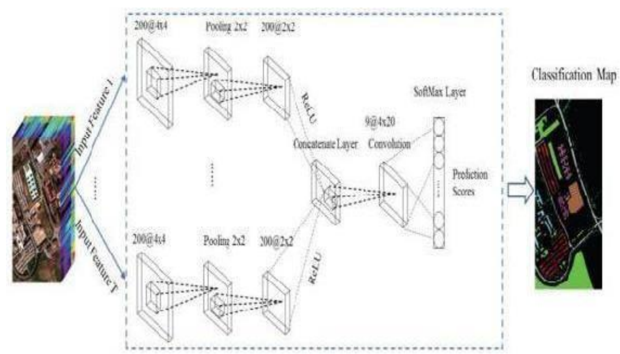

Fig. 1. Block diagram of capsule network.

Figure 1 shows the outline of the proposed framework. The first step of this framework is the release of multiple HSI features followed by several CNN blocks. Given a set of features, each CNN block will read a representative feature map, and all feature maps will be merged with the corresponding layer. The weight and selection of each block is optimized for this network with back-propagation. Network pixel output per pixel is an allowance for the preservation of class memberships and C units, corresponding to the C classes specified in the hyperspectral data set. The key principles of the proposed

International Journal of Scientific & Engineering Research, Volume 11, Issue 3, March-2020
ISSN 2229-5518

framework are described in detail in the following sections. The characterization of spatial data aggregated by morphological profiles (MPs) may represent differences in image properties. However, features produced by a particular MP cannot be modelled as other geometric features. In order to model different geometric features at the same time of feature extraction in the HSI system, the use of attribute profiles (APs) was first introduced in the work of APs have shown interesting properties in HSI performance, which can be used to generate extended AP (EAP). APs are a standard MP method, which can be found in an image using a criterion. The formation of APs is dependent on the morphological filter (AFs), and can be detected by following the sequence of AFs in the critical image. AFs are defined as connected agents that process the image by combining its linked elements instead of pixels. After the operators use the circuits, the attribute results are compared with the previously defined reference value. The region is determined to be saved or removed from the image depending on whether or not the reference is met (e.g., attribute results are stored if the value is greater than the predefined reference value). Values in the deleted region will be set as the closest grayscale value for the nearest region. If the merged region is at a low (large) level, then a decrease (thickness) is added. Subsequently, the AP can be constructed directly using the decreasing sequence and intensity of AF used in the image with the set of methods given.

Convolutional Neural Networks

CNNs aim to extract representative features of different data paths through the construction of multiple offline structures. The features learned by CNN are often more reliable and effective than the rules- based features. In this paper, we consider the separation of HSI and the so-called directed acyclic graphs (DAG) in which the layers are not confined to the consecutive contact. With HSI classification, the neural network is able to detect the HSI pixel resistance function of input to pixel emulation labels. Another important type of layers is a compilation that works as a sample function below. The most common types of address-calling are max-pooling and mean-pooling. The pool function divides the input feature map into a square cluster and extracts the max value / mean for each region. Therefore, complex cohesion can be reduced. Usually, the softmax function is performed in the top layer so that the potential distribution as output is obtained by each unit representing the probability of class membership. According to the above principle, in this paper, various aspects of the immature image are transmitted to each CNN compatible block, and the network is optimized for back propagation.

4 Implementation

4.1 Routing Algorithm

Step 1: Procedure ROUTING (,*p*,*k*)

Step 2: For all capsule *m* in layer *k* and capsule *n* in layer (*k+1*):*bmn*←0

Step 3: For *p* iterations do

Step 4: For all capsule m in layer k:cm←softmax(bm)

Step 5: For all capsule *n* in layer (*k+1*):s ←∑

Step 6: For all capsule *n* in layer (*k+1):*xj←squash(sn)

Step 7: For all capsule *m* in layer *k* and capsule *n* in layer (*k+1*):*bmn*←*bmn* + . n

Return xn.

Apply convolutional layer to neural network. That will output array of bunch of feature maps. We need to reshape this array to get set of vectors for each location. To preserve vector's orientation, we use squash (u) function. It also ensures length of vector between 0 to 1. It is because no vector is larger than 1 since vector length represents probability. After ap-ply-ing squash Function we get primary capsules. Those primary capsule layers predict output of next layer. In this layer, matrix multiplication of transformation matrix and own activation function matrix is computed. From this model will learn all the co-relationships. In the next step dot product of predicted output vector and actual output vector takes place which is then added to scalar product. Good predictions will have higher weight. When there is strong agreement u j—I is large and if it is wrong then its value is small. bij is scalar product that will have higher weights for accurate predictions. To allow multiple classes we need to minimize margin loss. Dynamic routing algorithm will overcome the ambiguity caused by crowdy images. If similar objects occupy nearest location in an image it will generate ambiguity for object identification. If image contains overlapping objects then also difficulties arise while identification and classification.so the dynamic routing is the best solution to overcome such problems. Involvement of large no of inner loops causes model to compute the calculations slowly but it will provide more accurate result with limited amount of data sample.

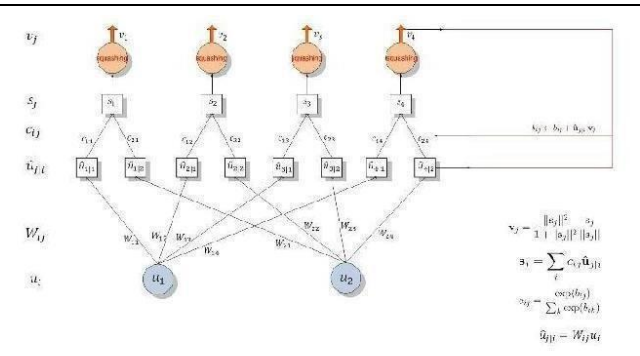

Fig. 2. Dynamic Routing.

We used CNNs that received input text in the form of sequences of integer representations of stemmed unigrams. Our character processing included the conversion of emoticons into word representations, and the removal of non-Latin characters. We also removed frequently occurring url

International Journal of Scientific & Engineering Research, Volume 11, Issue 3, March-2020
ISSN 2229-5518

components (e.g., names of popular websites), metadata encoded in the main body-text (e.g., 'RT: '), and a variety of social media platform- specific features. Hashtags and @-mentions were reduced to binary features. The text was then lower-cased and tokenized using NLTK's TweetTokenizer3 . The tokenized text was next encoded using a dictionary of integers, with the original ordering of the tokens preserved. The encoded text was converted into dense vectors of fixed size. This one-dimensional embedding was fed into a single-layer CNN with 200 embedding dimensions, 150 output dimensions, and 200 convolution kernels. The kernels were optimized using Tensorflow's 'adagrad' optimizer (lr=0.001) using categorical cross-entropy as the loss function.

The 150 output dimensions were flattened using a sigmoid function into two output nodes whose values are floats between 0 and 1, with 1 representing bullying and 0 representing non-bullying. To test the performance of our model, we took 70% of the dataset as training set, and 30% of it for testing. As suggested by previous research, we also added textual features (total used: 93) from LIWC 2015 4 to build another model for comparison. We set the threshold which got the best result (here we used highest F-measure to represent the performance). Comparison can be seen in Table 1. We put ZeroR and SVM (Support Vector Machine) models as baselines for comparison. Because the ZeroR model puts everything in the majority class, labelling all of the positive instances as negative ones, both the F-measure for the positive class and TruePositive score are 0. Meanwhile, the AUC values of the ZeroR model is 0.5 and the accuracy measure depends on the distribution of positives and negatives in the dataset. It is obvious that the performance of CNN model is better than that of the SVM model in terms of F-measure, AUC, and True Positive rate. It is also expected that adding LIWC features could help to improve the F-measure and accuracy. However, for other important parameters (i.e., True Positives and AUC value), our original model with NLTK- tokenized features got a better index. Note that all the thresholds are taken as 'optimal' because they lead to the highest F-measure, which is not only influenced by True Positive rate but also by the True Negative rate.

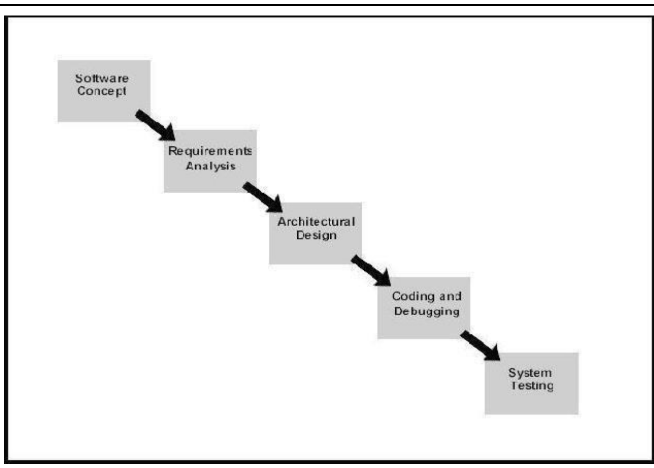

Fig. 3. Steps towards implementation.

The System Development Life Cycle is the process of developing information systems through investigation, analysis, design, implementation, and maintenance. The System Development Life Cycle (SDLC) is also known as Information Systems Development or Application Development. Below are the steps involved in the System Development Life Cycle. Each phase within the overall cycle may be made up of several steps.

Step 1: Software Concept

The first step is to identify a need for the new system. This will include determining whether a business problem or opportunity exists, conducting a feasibility study to determine if the proposed solution is cost effective, and developing a project plan. This process may involve end users who come up with an idea for improving their work. Ideally, the process occurs in tandem with a review of the organization's strategic plan to ensure that IT is being used to help the organization achieve its strategic objectives. Management may need to approve concept ideas before any money is budgeted for its development.

Step 2: Requirements Analysis

Requirements analysis is the process of analysing the information needs of the end users, the organizational environment, and any system presently being used, developing the functional requirements of a system that can meet the needs of the users. Also, the requirements should be recorded in a document, email, user interface storyboard, executable prototype, or some other form. The requirements documentation should be referred to throughout the rest of the system development process to ensure the developing project aligns with user needs and requirements. Professionals must involve end users in this process to ensure that the new system will function adequately and meets their needs and expectations.

Step 3: Architectural Design

After the requirements have been determined, the necessary specifications for the hardware, software, people, and data resources, and the information products that will satisfy the functional requirements of the proposed system can be determined. The design will serve as a blueprint for the system and helps detect problems before these errors or problems are built into the final system. Professionals create the system design, but must review their work with the users to ensure the design meets users' needs.

Step 4: Coding and Debugging

Coding and debugging are the act of creating the final system. This step is done by software developer.

International Journal of Scientific & Engineering Research, Volume 11, Issue 3, March-2020
ISSN 2229-5518

Step 5: System Testing

The system must be tested to evaluate its actual functionality in relation to expected or intended functionality. Some other issues to consider during this stage would be converting old data into the new system and training employees to use the new system. End users will be key in determining whether the developed system meets the intended requirements, and the extent to which the system is actually used.

Step 6: Maintenance

Inevitably the system will need maintenance. Software will definitely undergo change once it is delivered to the customer. There are many reasons for the change. Change could happen because of some unexpected input values into the system. In addition, the changes in the system could directly affect the software operations. The software should be developed to accommodate changes that could happen during the post implementation period.

5 Conclusion

To prove the CNNs' potential for HSI classification, present a framework containing a novel CNN model. The framework is designed to contain several CNN blocks with complete features such as installation. In addition to reading efficiency as well as providing location information and HSI content, maps for each block release feature are compiled and presented in sequential sections to obtain pixel label images. By using fine-tuning, the built-in network is not shallow but efficient, and can similarly exploit the interaction of different data visualizations and layer by layer using concatenating layer. When comparing a learning method based on a single CNN feature, the classification results were greatly improved by the multiple features involved. In addition, unlike traditional law-based actors, a CNN-based program can deliver more in-depth features more efficiently and more effectively.

We proposed the idea of classification of Hyperspectral images which can be used further to suite any application. It is done by selection and extraction of spatial features. Using capsule cap network architecture complexity can be reduced. Also, it gives more accuracy of classification. The model is able to extract a more relevant and complete information about HSI data cubes. This approach can work with limited amount of training data samples.

Here we present an idea which states that it is more robust to changes in the orientation and size of input. It would need much less data and internal representation; thus, it is more efficient to classify the data correctly. This means Capsnet can identify new unseen variations of the class without ever being trained on them.

References

[1] 1D-Convolutional Capsule Network for Hyperspectral Image Classification
Haitao Zhang, Lingguo Meng, Xian Wei, Member, IEEE, Xiaoliang Tang, Member, IEEE, Xuan Tang, Member, IEEE, Xingping Wang, Bo Jin, Member, IEEE, and Wei Yao, Member, IEEE.

[2] Dynamic Routing Between Capsules
Sara Sabour Nicholas Frosst Geoffrey E. Hinton Google Brain Toronto.

[3] Hyperspectral Image Classification: A k-means Clustering Based Approach
Sameer Ranjan, Deepak Ranjan Nayak, Kallepalli Satish Kumar, Ratnakar Dash, Banshidhar Majhi Pattern Recognition Lab, Department of Computer Science and Engineering

[4] Deep Learning-Based Classification of Hyperspectral Data
Yushi Chen, Member, IEEE, Zhouhan Lin, Xing Zhao, Student Member, IEEE, Gang Wang, Member, IEEE, and Yanfeng Gu, Member, IEEE.

[5] Investigation of the Random Forest Framework for Classification of Hyperspectral Data
JiSoo Ham, Yangchi Chen, Melba M. Crawford, Senior Member, IEEE, and Joydeep Ghosh, Senior Member, IEEE.

[6] Capsule Networks – A survey
Mensah Kwabena Patrick a, Adebayo Felix Adekoya a, Ayidzoe Abra Mighty b, Baagyire Y. Edward c.

[7] A new deep convolutional neural network for fast hyperspectral image classification
M.E. Paoletti, J.M. Haut, J. Plaza, A. Plaza Hyperspectral Computing Laboratory (HyperComp), Department of Computer Technology and Communications, Escuela Politecnica de Caceres, University of Extremadura, Avenida de la Universidad s/n, E-10002 Caceres, Spain.

[8] Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines.
IEEE Trans. Geosci. Remote Sens. 2004, 42, 1778–1790. [CrossRef].

[9] Pal, M.; Foody, G.M. Feature selection for classification of hyperspectral data by svm.
IEEE Trans. Geosci. Remote Sens. 2010, 48, 2297–2307. [CrossRef].

[10] Investigation of the random forest framework for classification of hyperspectral data.
Ham, J.; Chen, Y.; Crawford, M.M.; Ghosh, IEEE Trans. Geosci. Remote Sens. 2005, 43, 492–501. [CrossRef].

[11] Semisupervised hyperspectral image classification using soft sparse multinomial logistic regression.
Li, J.; Bioucas-Dias, J.M.; Plaza, A. IEEE Geosci. Remote Sens. Lett. 2013, 10, 318–322.

[12] Classification of hyperspectral data from urban areas based on extended morphological profiles.
Benediktsson, J.A.; Palmason, J.A.; Sveinsson, J.R.IEEE Trans. Geosci. Remote Sens. 2005, 43, 480–491. [CrossRef].

[13] Spectral and spatial classification of hyperspectral data using svms and morphological profiles.

International Journal of Scientific & Engineering Research, Volume 11, Issue 3, March-2020
ISSN 2229-5518

Fauvel, M.; Benediktsson, J.A.; Chanussot, J.; Sveinsson, J.R IEEE Trans. Geosci. Remote Sens. 2008, 46, 3804– 3814. [CrossRef].

[14] Spectral–spatial hyperspectral image segmentation using subspace multinomial logistic regression and markov random fields.
Li, J.; Bioucas-Dias, J.M.; Plaza, A. IEEE Trans. Geosci. Remote Sens. 2012, 50, 809–823. [CrossRef].