Abstract—A Hyperspectral Image (HSI) contains much more number of channels as compared to a Red, Green, Blue (RGB) image, hence containing more information about entities within the image. The Convolutional Neural Network (CNN) and the Multi-Layer Perceptron (MLP) have been proven to be an effective method of image classification. However, they suffer from the issues of long training time and requirement of large amounts of the labeled data, to achieve the expected outcome. These issues become more complex while dealing with hyperspectral images. To decrease the training time and reduce the dependence on large labeled dataset, method of transfer learning is proposed in this paper. The hyperspectral dataset is preprocessed to a lower dimension using Principal Component Analysis (PCA), then Deep Learning (DL) models are applied to it for the purpose of classification. The features learned by this model are then used by the transfer learning model to solve a new classification problem on an unseen dataset. A detailed comparison of CNN and multiple MLP architectural models is performed, to determine an optimum architecture that suits best the objective. The results show that the scaling of layers not always leads to increase in accuracy but often leads to overfitting, and also an increase in the training time. The training time is reduced to a greater extent by applying the transfer learning approach rather than just approaching the problem by directly training a new model on large datasets, without much affecting the accuracy.

Index Terms—Hyperspectral Image Classification, CNN, MLP, Transfer Learning

I. INTRODUCTION

Hyperspectral Image (HSI) is an image cube made up of hundreds of spatial images. A normal image has 3 channels of pixel values, namely, Red, Green, and Blue (RGB). In contrast, HSI record tens to hundreds of narrow colour channels (i.e. wavelengths). To process different qualities of HSI, it is necessary to combine all the narrow colour channels, which form the basis of the classification of HSI. The high dimensionality and lack of data samples make these images highly complex to classify.

The Convolutional Neural Network (CNN) is a class of Deep Learning (DL) neural networks. CNNs represent a huge breakthrough in image recognition. The use of CNN for HSI classification is also visible in recent works [1][2].

While CNNs have become a standard model for the image classification, the Multi-Layer Perceptron (MLP) models are also extensively used for the HSI classification, especially when the computing resources are limited [3]. MLP classification is recommended for the embedded platforms, as it showed the shorter run-time as compared to CNN and comparable accuracy.

However, in the case of HSIs, training time is long for the DL models, and the demand for the marked data is large. To overcome this issue, we try transfer learning for the problem. DL models are trained on some other dataset and using the learned parameters, the image in the given dataset is classified. This would reduce the training time as well as help to overcome the obstruction of having fairly small dataset given the high dimensionality. With the advent of modern remote sensing technology it has become possible to obtain much higher resolution and much better quality satellite images than ever before. One category of satellite images is hyperspectral images, which work similar to ordinary images.

HSI is an image captured in such a way that each pixel includes complete spectrum. Thus the HSI forms a three dimensional data (i.e. is in the form of a data cube). In the Hyperspectral Dataset, we have an image along with its correct labels, also known as ground truth values. Since the availability of the labeled Hyperspectral data is low, so to avoid overfitting and also to reduce the training time, the transfer learning approach is explored. In their survey, Shao et al., have very well explained the application of transfer learning in the context of visual categorization [4]. Using related data from a different domain, we expand the knowledge for the problem domain by extracting important information from a related domain and transferring it to be used in the target domain. As future data will almost certainly over-fit, training data in the same feature space or with the same distribution can be used here.

The DL models will be trained for the classification of the Indian Pines dataset. These models will then be used to transfer the knowledge extracted from the Indian Pines dataset, i.e. they will act as a pre-trained model for the classification task of the University of Pavia dataset.

This paper aims to compare the different DL models coupled with transfer learning, to perform classification of the Hyperspectral Images (the University of Pavia image in our case).

The organization of the manuscript is as follows: Section 2 states the literature Review part, section 3 is about the proposed methodology with pre-processing part, CNN and MLP Model training and Transfer learning followed by section 4 that explains results with dataset. Finally section 5 are about conclusion of the manuscript.
Recent developments in the field of machine learning have identified DL to be a powerful tool to achieve feature extraction and address nonlinear problems, furthermore DL models are already heavily used in several image processing applications. Considering these factors, and especially considering the achievements of DL techniques in other, similar applications, DL has also been introduced to classify HSIs. In the paper, Ying Li et al., authors have systematically reviewed several pixel-wise and scene-wise Remote Sensing image classification approaches which use DL methods [5]. The authors performed comparative analysis on CNNs, Stacked Autoencoders and Deep Belief Networks(DBN) and involved analysis of Spectral, Spatial, and Spectral-Spatial feature classification. Their results on the Pavia University dataset show that CNN works most accurately with spectral-spatial feature classification. Additionally, they found dual channel CNNs to provide the highest accuracy on the Indian Pines dataset. A detailed comparative study of the various DL models used for hyperspectral classification including DBN, Generative Adversarial Networks(GAN’s), Recurrent Neural Networks(RNNs) and CNNs and it’s different variants like the Gabor-CNN, S-CNN and RESNET, has been presented by Shutao Li et al. [6]. It was observed that the CNN based models(S-CNN,Gabor-CNN) were particularly good for the classification task, with most of them having state of the art accuracies (>99%). It was also shown that the DL models outperform the traditional machine learning models like SVM, EMP which were also used for the purpose of classifying hyperspectral images.

The Spectral-spatial features are extracted from a target pixel and its neighbours. The convolution on these spectral-spatial features leads to a number of 1D feature maps. The feature maps are stacked into a 2D matrix and can be fed to the convolutional neural net. This type of model can be referred as a HSI-CNN model [7]. For using classifiers like KNN and SVMs one requires a more in-depth knowledge of HSI, on the other hand DL techniques like neural nets allow us to extract complicated features from hidden layers without too much pre-processing of the data. HSI-CNN is a good trade-off between number of training samples and complexity of the network, and overcomes over-fitting. XGBoost can be considered as a substitution of the softmax layer of HSI-CNN in order to prevent over-fitting.

The Multi-Layer Perceptron (MLP) have also been extensively used for classification of the multispectral as well as hyperspectral images. They are relatively fast to train and produce comparable results to the CNNs. A better performance is observed when they are coupled with the Principal Component Analysis PCA [8]. A study by Beatriz et al. has focused on the time optimization for the classification of HSIs by MLP. They proposed 5 different architectures for MLP and compared them with the state of the art SVM model. They also proposed a parallel programming scheme for training the MLP [9]. A variation can be using SVM in the last layer of the CNN model instead of the traditional softmax layer (SVM-CNN model). The resultant model would now require a lesser training time in contrast to that with the softmax layer, which happens to be more computationally expensive due to the softmax function. The model with the softmax layer has been showed to slightly outperform the SVM-CNN in terms of accuracy, but the SVM-CNN is slightly faster to train [10].

The paper by Roy et al. shows the advantages of using Hybrid Spectral Net (Hybrid SN), a 3D-2D CNN instead of using pure 2D or pure 3D CNN [1]. The pure 2D CNN on its own is not able to extract features regarding the spectral dimensions while pure 3D CNN would become computationally expensive to use and it performs worse for classes having similar textures over many spectral dimensions. The 3D-CNN and 2D-CNN layers are assembled for Hybrid SN in such a way that they utilise both the spectral as well as spatial feature maps to their full extent to achieve maximum possible accuracy. To reduce the number of spectral bands (more accurately remove the spectral density) PCA is applied over the HSI image. The HSI data cube is divided into small overlapping 3D patches, their truth labels are decided by the label of the centered pixel. The 2D convolution is applied once before the flatten layer to extract abstract high level features. The research paper confirms the superiority of the proposed method by performing experiments over three benchmark datasets and comparing the results with state of the art methods. The proposed model is shown to be computationally more efficient than the pure 3D CNN model.

The paper by Xuefeng Jiang et al. [11] emphasizes a new technique of feature extraction using transfer learning which significantly lowers the training time of the CNN used and reduces its dependency on large labelled datasets. This leverages the Markov property of the images to separate images with class tags and train the CNN on random band samples selected from the datasets. The paper by Ke Li et al. [12] explores another transfer learning approach for hyperspectral image classification using deep belief networks wherein a limited Boltzmann machine Network is trained on the source domain data and its first few layers are extracted to be used for the target domain network. The target domain network is fine tuned further and used for classification of images in the target domain. The number of layers to be transferred are also varied and chosen for best accuracy.

CNN-based frameworks can achieve high accuracy in an object classification task. But when coupled with transfer learning, they can give high performance in detection tasks as well, while reducing the required training time and computing resources [13].

Taking inspiration from these papers we use a transfer learning approach coupled with CNN and MLP models, and perform a comparative analysis in the end.

III. PROPOSED METHODOLOGY

A. Dataset Preparation

Two datasets, namely the Indian Pines and the Pavia University, are used, the Indian Pines for training the CNN model
from which the CNN stump (the convolutional layers and the pooling layers barring the fully connected layers) was extracted to be trained on the Pavia dataset. Both images have different distributions that thus makes the applicability of Transfer Learning perfectly suitable.

| S.No. | Class                  | Samples |
|-------|------------------------|---------|
| 1     | Alfalfa                | 46      |
| 2     | Corn-notill            | 1428    |
| 3     | Corn-mintill           | 830     |
| 4     | Corn                   | 237     |
| 5     | Grass-pasture          | 483     |
| 6     | Grass-trees            | 730     |
| 7     | Grass-pasture-mowed    | 28      |
| 8     | Hay-windrowed          | 478     |
| 9     | Oats                   | 20      |
| 10    | Soybean-notill         | 972     |
| 11    | Soybean-mintill        | 2455    |
| 12    | Soybean-clean          | 593     |
| 13    | Wheat                  | 205     |
| 14    | Woods                  | 1265    |
| 15    | Buildings-Grass-Trees-Drives | 386 |
| 16    | Stone-Steel-Towers     | 93      |

TABLE I: Ground-truth classes for the Indian Pines scene and their respective samples number

1) The Indian Pines (IP): This dataset[14] has an image with 145x145 spatial dimensions and 224 spectral bands in the wavelength range of 400 to 2500 nm, out of which 24 spectral bands covering the region of water absorption have been discarded. This dataset was collected by the AVIRIS sensor over the Indian Pines test site in North Western Indiana. The ground truth available is designated into 16 classes of vegetation. Two-third of the Indian Pines is agricultural landscape and the remaining one-third contains perennial vegetation. Table I described the ground truth.

2) The Pavia University (PU): The Pavia University dataset[15] was used to test the model built using transfer learning. It was acquired by the ROSIS sensor over Pavia in northern Italy. It contains 103 spectral bands with 610x610 pixels. The ground truth values for the pixels cover 9 different classes. Table II describes the ground truth classes for the Pavia University dataset and their respective sample numbers.

| S.No. | Class                  | Samples |
|-------|------------------------|---------|
| 1     | Asphalt                | 6631    |
| 2     | Meadows                | 18649   |
| 3     | Gravel                 | 2099    |
| 4     | Trees                  | 3064    |
| 5     | Painted metal sheets   | 1345    |
| 6     | Bare Soil              | 5029    |
| 7     | Bitumen                | 1330    |
| 8     | Self-Blocking Bricks   | 3682    |
| 9     | Shadows                | 947     |

TABLE II: Ground-truth classes for the Pavia University scene and their respective samples number

Images from the IP and PU datasets were processed with PCA for dimensional reduction prior to model training, which decreased the spectral dimension for each image to 30. This is done so as to facilitate faster training time for the models and reduce the memory cost. Distinct image cubes must be prepared from the images before feeding the dataset for training, and the size of image cubes behaves as a hyper-parameter. The accuracy could be improved by making the cube larger[2], but the amount of RAM needed will increase. The image cubes are taken at a size of 5x5x30 for this purpose. Furthermore, in contrast to a CNN model, which can be directly fed an image, an MLP model requires a flat matrix of input parameters, hence flattening the provided input image is necessary after reducing the dimension using PCA.

B. Model Training

All of the following models were made in 2 phases. In first phase, the models were trained on the Indian Pines dataset, and then in second phase, transfer learning was applied and models were trained on Pavia University dataset.

1) CNN Model: First, the IP dataset is used to train a CNN model which uses both 3-D and 2-D convolution operations to leverage feature map extraction from both the spatial and spectral dimensions. The ratio of train to test for the data-cubes, which are created during preprocessing of the data, is 70:30.

The first 3 layers are convolution 3-D layers, which consist of 8, 16, and 32 filters, respectively later on a convolution 2-D layer, which is the fourth layer, is applied with 64 filters and a 3 by 3 filter size. After flattening (turning all of the activations into 1-D vectors), the output of this convolution layer is fed into an artificial neural network-like architecture that primarily comprises of two dense, fully connected layers with leaky ReLU activation and a dropout regularisation of 0.4. Finally, there is an output layer that predicts classes using Softmax Activation. For training CNN, the Adam Optimizer is used.

Once the above model is trained on the IP dataset, the CNN root model will be used further, i.e., all the layers are stripped. To this convolutional stump, a new set of 3 fully connected layers with ReLU activation and a dropout of 0.4 with a softmax layer of 9 units is appended. The layers excluding the fully connected layers and the softmax layers are frozen, so that the weights for those layers are not modified and training only takes place for the fully connected layers and the softmax
layer. The patches obtained from the PU dataset are then split into the training and the test sets in a ratio of 40:60, and trained.

2) **MLP Model 1:** The MLP model, represented in fig. 1 is applied in this research, is a typical MLP model with three dense layers that each include 10,000, 5,000, and nine perceptrons. The model was initially trained on the IP dataset, and the final two layers were removed before applying the model to make predictions via transfer learning on the PU dataset. The ReLu activation function is used to activate the first and second layers, and the SoftMax activation function is used to activate the last layer that provides the class identifiers. The MLP perceptron layers were trained using Adam Optimizer.

3) **MLP Model 2:** IIT Delhi has been conducting research on the classification of terrain using hyperspectral images with the help of the DRDO. We adopted an MLP model architecture that served their needs. The model can be visualised diagrammatically as in Fig 2.

In this model the initial training was done in the IP dataset with just two Hidden Layers. Later when transfer learning was performed, the last layer along with its batch normalization extension was chopped off, and three more layers were added. The first layer is a Dense Layer of 472 perceptrons, followed by a Batch Normalisation layer. This first layer was left as it is during transfer learning, and three more layers were added with 168, 72 and 9 perceptrons respectively. These layers were the trainable layers that were trained during transfer learning. All these perceptron layers are activated using a ReLu activation function and the optimization technique used while training is Adam Optimization, much like the previous model.

4) **MLP Model 3:** This is an expansion of Model 2 with more pre-trained layers but fewer perceptrons since models need to be scaled horizontally, which involves increasing the number of layers while decreasing the strength of each layer. The visual representation of the model is given in Fig 3.

This model was trained with 5 Hidden layers on IP dataset, but when transfer learning was applied, its last two layers were chopped off, and three new dense layers were added. The layers from the saved models were not trained on the PU dataset, while the three dense layers were trained. The pretrained layers consisted of Dense layers with 1024, 512, 256 and 128 perceptrons, with each layer being followed by an appropriate Batch Normalization layer. All the perceptron layers are activated using a ReLu activation function and the optimization technique used while training is Adam Optimization, much like the previous model.

### IV. Results

A. **CNN Model**

Firstly, the IP dataset is applied to the CNN model. Test set accuracy came out to be 99.31% which is quite high. This trained model was saved to be used for transfer learning purposes in future. The test set accuracy for the PU dataset came out to be 99.93%. The prediction made are shown in Fig 5b.

B. **MLP Model 1**

The MLP model was a very basic model with varying activation functions, and it was first trained on the IP dataset. It can be seen that the average accuracy of the model is 99.70% which is acceptable. The accuracy that is received out of
the model on the PU dataset is close to 96.78% which is acceptable in the context of transfer learning. It should be noted that the accuracy decreased before and after applying the transfer learning because the initial layer was still trained with weights of the previous Indian Pines dataset, and this decrease in accuracy on applying transfer learning algorithm is very commonly observed.

C. MLP Model 2

This MLP model was first trained on the IP dataset. In the table III, we can see that the average accuracy of the model is 99.95% which is quite high. The accuracy that we received out of the model on the PU dataset is close to 98.71% which is still quite high when we see in the context of transfer learning. The accuracy was also a substantial leap from that of the basic model, and hence we can say that this model out-performed the previous model.

D. MLP Model 3

This is the third and final MLP model that has been trained on the IP dataset. The average accuracy of the model is 99.90% which is still quite high. The accuracy that we received out of the model is close to 98.19% which is quite lower than what we had already achieved with the previous models. This was probably seen because the model had three large perceptron layers that were not trained during transfer learning, so the model was actually overfitting the Indian Pines dataset and the extra few layers that were later trained on the Pavia dataset, were not able to align the weights and capture the true features.

Comparison

We have seen the output of the three different MLP models, and compared them to decide which architecture best fits our objective. The first model being a very basic model is outperformed by the second model, which though requires a higher computational power and time for training but also provides a higher accuracy. In case of the third model, we saw that mere horizontally scaling and increasing the number of perceptron layers did not result in increase in the accuracy due to overfitting over the dataset. In the table III, all the three models are compared in a tabular form, and though by looking at the output predicted images, being it of the Indian Pines or the Pavia dataset, all three seems to be exactly the same, but when we see the test accuracy and the average accuracy store we find a difference there.

With the above comparison, it is concluded that the second MLP model is the better of the three, with a test accuracy of 99.95% on IP and of 98.71% on the PU Dataset. When we compare the above result with the one obtained while addressing the same problem by using a CNN based transfer learning approach, we can conclude that the latter is faster in terms of training and convergence to the optimum result. But the CNN also captures the spatial features of the image dataset, apart from just the spectral ones, whereas the MLP model only captures the spectral features.

V. Conclusion

The MLP-based models are simple and utilized for a variety of image classification goals. Since there are no tasks to capture the spatial information, such as convolution or pooling, they require less training time than CNN. There is an upper limit to the accuracy that can be achieved by scaling the MLP design, either vertically or horizontally. After that point, the model begins to experience overfitting and performs poorly in the case of transfer learning. The thorough comparison of the three different MLP models that were used and explained clearly demonstrated these facts.

The use of transfer learning substantially decreases the training time required for a model, on larger datasets like PU. Transfer learning works by extracting weights and knowledge from smaller datasets like the IP dataset. In our case, we take advantage of the inner layers trained in the previous model and use it’s general information (in our case about the hyperspectral images) to train the outermost few layers. Thus, saving time by not having to train the inner layers again.

There was also debate about whether to use a CNN-based transfer learning model or an MLP-based model. Which model to choose depends on the application and goal and may change depending on the resources available. A CNN-based model would be more effective in this situation since the spatial properties need to be taken into account when using terrain classification for things like agricultural research. In contrast, spectral features in applications requiring military remote sensing ignore spatial ones; as a result, the MLP-based model should be chosen over CNN in this scenario.

REFERENCES

[1] S. K. Roy et al. “HybridSN: Exploring 3-D–2-D CNN Feature Hierarchy for Hyperspectral Image Classifica-
| Model                  | MLP Model 1          | MLP Model 2          | MLP Model 3          |
|-----------------------|----------------------|----------------------|----------------------|
| Layers                | 3                    | 4                    | 7                    |
| Architecture          | (18750,10000,5000,9) | (18750,472,168,72,9) | (18750,1024,512,256,128,72,32,9) |
| Trainable Params      | 50,050,009            | 12,825               | 11,921               |
| IP Test Accu.         | 99.70%               | 99.95%               | 99.90%               |
| IP Avg. Accu.         | 99.59%               | 99.96%               | 99.78%               |
| Pavia Test Accu.      | 96.78%               | 98.71%               | 98.19%               |
| Pavia Avg. Accu.      | 95.37%               | 97.84%               | 96.83%               |

TABLE III: Comparison of MLP models

[1] Indian Pines. URL: http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes.

[2] Uphar Singh et al. Agricultural Plantation Classification using Transfer Learning Approach based on CNN. 2022. DOI: 10.48550/ARXIV.2206.09420. URL: https://arxiv.org/abs/2206.09420.

[3] S. Balakrishnan et al. “Deep Learning for Hyperspectral Image Classification on Embedded Platforms”. In: 2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS), 2018, pp. 187–191.

[4] L. Shao, F. Zhu, and X. Li. “Transfer Learning for Visual Categorization: A Survey”. en. In: IEEE Transactions on Neural Networks and Learning Systems 26.5 (May 2015), pp. 1019–1034.

[5] Ying Li et al. “Deep learning for remote sensing image classification: A survey”. In: WIREs Data Mining and Knowledge Discovery 8.6 (2018), e1264. DOI: 10.1002/widm.1264. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/widm.1264. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1264.

[6] Shutao Li et al. “Deep Learning for Hyperspectral Image Classification: An Overview”. In: IEEE Transactions on Geoscience and Remote Sensing 57.9 (Sept. 2019), pp. 6690–6709. ISSN: 1558-0644. DOI: 10.1109/tgrs.2019.2907932. URL: http://dx.doi.org/10.1109/TGRS.2019.2907932.

[7] Yanan Luo et al. HSI-CNN: A Novel Convolution Neural Network for Hyperspectral Image. 2018. arXiv: 1802.10478 [cs.CV].

[8] W. d. Silva et al. “Multispectral Image Classification Using Multilayer Perceptron and Principal Components Analysis”. In: 2013 BRICS Congress on Computational Intelligence and 11th Brazilian Congress on Computational Intelligence. 2013, pp. 557–562.

[9] Beatriz P. Garcia-Salgado, Volodymyr I. Ponomaryov, and Marco A. Robles-Gonzalez. “Parallel multilayer perceptron neural network used for hyperspectral image classification”. In: Real-Time Image and Video Processing 2016. Ed. by Nasser Kehtarnavaz and Matthias F. Carlsohn. Vol. 9897. International Society for Optics and Photonics. SPIE, 2016, pp. 141–153. DOI: 10.1117/12.2227329. URL: https://doi.org/10.1117/12.2227329.

[10] Abien Fred Agarap. An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification. 2017. arXiv: 1712.03541 [cs.CV].

[11] X. Jiang et al. “Hyperspectral Image Classification With Transfer Learning and Markov Random Fields”. In: IEEE Geoscience and Remote Sensing Letters 17.3 (2020), pp. 544–548.

[12] Ke Li et al. “A Novel Method of Hyperspectral Data Classification Based on Transfer Learning and Deep Belief Network”. In: Applied Sciences 9 (Apr. 2019), p. 1379. DOI: 10.3390/app9071379.

[13] P. Pooyoi et al. “Snow Scene Segmentation Using CNN-Based Approach With Transfer Learning”. en. In: 2019 16th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). Pattaya, Chonburi, Thailand, 2019, pp. 97–100.

[14] Indian Pines. URL: http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes.

[15] Pavia University. URL: http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes.