2020

Improving Transfer Learning for Use in Multi-Spectral Data

Yuvraj Sharma
Technological University Dublin

Improving Transfer Learning for Use in Multi-Spectral Data

Yuvraj Sharma

Technological University Dublin, Ireland

A dissertation submitted in partial fulfilment of the requirements of
Technological University Dublin for the degree of
M.Sc. in Computer Science (Data Science)

2020

DECLARATION

I certify that this dissertation which I now submit for examination for the award of MSc in Computing (Data Science), is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Technological University Dublin and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: 

Date: **07 September 2020 (DD Month Year)**

ABSTRACT

Recently Nasa as well as the European Space Agency have made observational satellites images public. The main reason behind opening it to public is to foster research among university students and corporations alike. Sentinel is a program by the European Space Agency which has plans to release a series of seven satellites in lower earth orbit for observing land and sea patterns. Recently huge datasets have been made public by the Sentinel program.

Many advancements have been made in the field of computer vision in the last decade. Krizhevsky, Sutskever & Hinton, 2012, revolutionized the field of image analysis by training deep neural nets and introduced the idea of using convolutions to obtain a high accuracy value on coloured image dataset of more than one million images known as Imagenet ILSVRC. Convolutional Neural Network, or CNN architecture has undergone much improvement since then. One CNN model known as Resnet or Residual Network architecture (He, Zhang, Ren & Sun, 2015) has seen mass acceptance in particular owing to it processing speed and high accuracy. Resnet is widely used for applying features it learned in Imagenet ILSVRC tasks into other image classification or object detection tasks. This concept, in the domain of deep learning, is known as Transfer learning, where a classifier is trained on a bigger more complex task and then learning is transferred to a smaller, more specific task. Transfer learning can often lead to good performance on new smaller tasks and this approach has given state of the art results in several problem domains of image classification and even in object detection (Dai, Li, He, & Sun, 2016).

The real problem is that not all the problems in computer vision field belongs to regular RGB images or images consisting of only Red, Green, and Blue band set. For example, a field like medical image analysis has most of the images belonging to greyscale color space, while most of the Remote sensing images collected by satellites belong to multispectral bands of light. Transferring features learned from Imagenet ILSVRC tasks to these fields might give you higher accuracy than training from scratch, but it is a problem of fundamentally incorrect approach. Thus, there is a need to create network models that can learn from single channel or multispectral images

and can transfer features seamlessly to similar domains with smaller datasets.This thesis presents a study in multispectral image analysis using multiple ways of feature transfer. In this study, Transfer Learning of features is done using a Resnet50 model which is trained on RGB images, and another Resnet50 model which is trained on Greyscale images alone. The dataset used to pretrain these models is a combination of images from ImageNet (Deng, Dong, Socher, Li, Li, & Fei-Fei, 2009) and Eurosat (Helber, Bischke, Dengel, & Borth. 2017). The idea behind choosing Resnet50 is that it has been doing really well in image processing and transfer learning and has outperformed all the other traditional techniques, while still not being computationally prohibitive to train in the context of this work.

An attempt is made to classify different land-cover classes in multispectral images taken up by Sentinel 2A satellite. The dataset used here has a key challenge of a smaller number of samples, which means a CNN classifier trained from scratch on these small number of samples will be highly inaccurate and overfitted. This thesis focuses on improving the accuracies of this classifier using transfer learning, and the performance is measured after fine-tuning the baseline above Resnet50 model. The experiment results show that fine-tuning the Greyscale or single channel based Resnet50 model helps in improving the accuracy a bit more than using a RGB trained Resnet50 model for fine tuning, though it haven't achieved great result due to the limitation of lesser computational power and smaller dataset to train a large computer vision network like Resnet50.

This work is a contribution towards improving classification in domain of multispectral images usually taken up by satellites. There is no baseline model available right now, which can be used to transfer features to single or multispectral domains like the rest of RGB image field has. The contribution of this work is to build such a classifier for multispectral domain and to extend the state of the art in such computer vision domains.
Key words: Deep learning, Transfer learning, Image Analysis, Resnet, CNN, Multispectral images, ImageNet, Satellite imagery, EuroSat

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervisor, Dr. Robert Ross for his consistent support, guidance and encouragement while doing this research thesis. He has in-depth knowledge and expertise of the field of AI and Deep Learning and it has helped me a lot during the time. He has a great ease of working with and has a lot of experience of guiding students through their research, which makes him incredibly empathic and humble. He provided me proper working schedule, and this has helped me in finishing this thesis in time.

I would also like to thank Pallavi Jain, PhD student at TU Dublin, for timely help with data and providing valuable feedback in my initial work plan. She provided me with excellent ideas that have helped me in conducting this research well.

Lastly, I would like to thank my family for their always, ongoing non-questionable support in my endeavours and professional journey.

TABLE OF CONTENTS

Table of Contents

TABLE OF FIGURES

TABLE OF TABLES

1 INTRODUCTION

The Sentinel-2 satellite images are openly and freely available in the Earth Observation (EO) program known as Copernicus. A classification system using satellite images can have multiple use cases like for example, detecting land use changes and land cover changes over time, or helping to improve geographical maps, or applications to the domain of agriculture, climate change, forest fires, urban development and forest cover erosion.

These satellites images are usually from multiple channels of the sunlight and not just the visible light spectrum (red, green, blue). The data used in the research work is both, visible spectrum and RGB channel data as well as the multispectral data.

In order to apply deep learning algorithms to satellite data, first images should be processed and divided into different classes, like for example the fundamental classes of land use and land cover, and secondly data needs to be of huge size in order for neural network nodes to learn the features inherent in it. Unfortunately, the available labelled datasets are small-scale and thus don't allow efficient processing. In addition to this, images taken by satellite are multispectral in nature, meaning they can have multiple bands in an image, other than just visible bands of RGB. The data that the thesis uses has for example, thirteen frequency bands for every image.

This research work has aimed to provide a benchmark demonstrating a robust performance in classification of multispectral images which could help in developing applications for the above-mentioned domains. This work has hypothesized that a large convolutional network trained on single channel images, can learn more relevant features for multispectral image analysis, than the one trained on coloured images.

1.1 Background

With the development of remote sensing technologies, the usage of Earth Observation images has increased to a great extent in the last couple of decades. Satellite images

are used in a wide variety of applications. For example, for tracking roads from satellite images (Geman and Jedynak,1996), where the authors used every image to reduce overall entropy or uncertainty in identifying a road in a 1D representation of a satellite image. Yet another use-case is of detecting vehicles in satellite images using a hybrid between deep Convolutional Neural Networks and traditional image feature classification techniques like histograms of gradient, binary patterns, and scale-invariant feature transform (Chen, Xiang, Liu, & Pan, 2014). Another very useful application is in flood extent detection, where satellite images can be used to determine whether there is a flooding event happening in part of a city at any given time (Jain, Schoen-Phelan, & Ross, 2020a; 2020b). Other important applications include assessment of large tribal areas, comparison of two landmasses or regions, and tracking changes in sea-coast lines due to rising water levels.

The cost of launching a satellite is going down rapidly, firstly, due to the advent of companies like SpaceX and Blue Origins, which have reusable rockets, and secondly due to the general reduction in the price of electronics and hardware machinery. Coupled to this is the rapid advancement happening in the computer vision field in the last decade. Artificial Intelligence (AI) powered applications and devices have increased the demand of large-scale data, as well as piqued the interest among the general masses and governments alike. AI has enabled organisations to look towards Earth Observation (EO) for information on buildings, natural structures, urban and rural boundaries, natural calamities, both military and civil operations, forest fires, melting glaciers vanishing forest covers, and monitoring humanitarian crisis.

Satellite image classification has many challenges too, like high variability inherent in EO data, small labelled datasets, low spatial resolution outputs, and the multispectral nature of images to name a few of them. Due to these issues, most of the current image classification approaches are not suitable for handling this kind of data. And it is a research area which is still not fully captured by companies and universities alike. Normalization of satellite images or putting these images to use is also not easy, mainly due to the presence of clouds in earth observation images, or due to haze and other prevailing weather conditions, or due to the changes in lighting of an area at different times in a day and during different periods in an year.

There have been several attempts to get around small labelled dataset problem in satellite image domain. One way is to build new and accurate large labelled training sets like EuroSat and SpaceNet (Van Etten, Lindenbaum, & Bacastow, 2018). Another approach uses unsupervised feature extraction from an image (Basu et. al.,2015), using large RGB-trained CNN based networks like VGG16 for transfer learning (Pallavi, Schoen-Phelan & Ross, 2020), and lastly training CNNs over small available data and producing low accuracy results. Another issue is that the objects are very small in Satellite imagery, this is one of the key differences between natural image datasets like ImageNet and satellite image datasets. Attempts have been made to transfer features learned in classification problems of the former nature and transferring them to the later ones. However, since these two domains are primarily of different nature, thus accuracies achieved are not in high ranges.

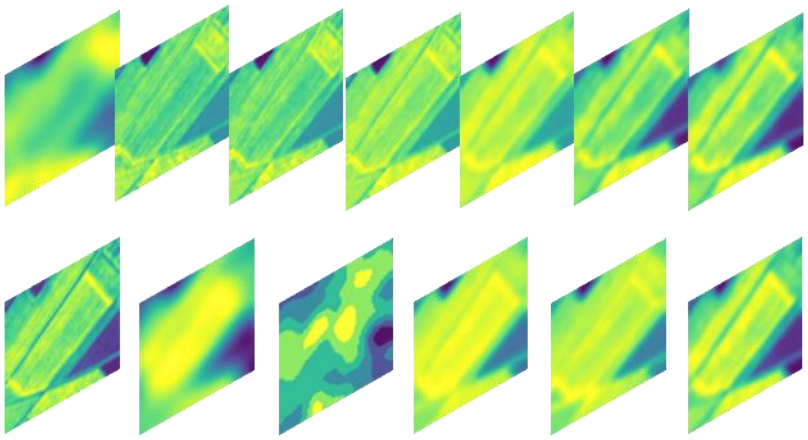

Figure 1.1 Thirteen Multispectral bands of an image as captured by Sentinel 2A satellite. Bands 01 to Band 13 (from left to right and top to bottom). A multispectral image captures image data for small number of different ranges or spectral bands.

In this research, attention has been given to this aspect of the problem - different nature of satellite images than regular natural RGB based images. A satellite image is usually multispectral, meaning it has multiple frequency channels summarized into a single image. This in turn means a lot of information gets stored in a remote sensing image than a typical RGB image (refer figure 1.1). This research approaches the problem of analysing a multispectral image, by firstly training a large network, like Resnet50, on

single channel image dataset, like a dataset consisting of greyscale images, and then using this large model to transfer features to target domain of multispectral image classification. The idea is to obtain a better classification accuracy while performing transfer learning in comparison to an RGB trained similar network.

1.2 Research Problem

The size of a labelled satellite dataset is usually very small (a few thousand images only) hence prediction accuracy of a CNN network trained on these images from scratch can't get very high. Secondly, transfer learning using a model pre-trained on RGB (coloured) images is arguably not the right approach when your target dataset consists of multispectral or multiple band images only. Of late, all the transfer learning is happening around ImageNet or other comparable RGB image databases, even for single channel grey-scale domains like medical Imaging (Cheplygina, 2019). Multispectral images and natural images are extremely different to one another, so any meaningful transfer is highly doubtful. It is also observed that the usefulness of a pre-trained network increasingly decreases as the task the network is trained on moves away from the target task (Yosinski, Clune, Bengio & Lipson, 2014). This ultimately raises the following research question as -

"To what extent can a CNN neural network, pre-trained on single channel (grey-scale) Imagenet and Eurosat** images, improves the image classification accuracy of multispectral images in comparison to a comparable model trained on colour images."*

* ImageNet dataset is a large-scale collection of natural images built upon the backbone of the WordNet structure (Deng, Dong, Socher, Li, Li, & Fei-Fei, 2009).
** The EuroSat dataset consists of Sentinel-2 satellite images, which are openly and freely provided by the EO (Earth observation) program Copernicus. This dataset covers 13 spectral bands and consists of 10 classes with a total of 27,000 labelled and geo-referenced images (Helber, Bischke, Dengel, & Borth, 2017).

Given this, we can formalise our Hypothesis as follows: If a deep neural network trained on greyscale images is fine-tuned on a dataset of single frequency images extracted from multispectral images taken by satellites, to classify images into one of the ten land-cover classes, then the accuracy is higher than both, when the model is trained from scratch and when a model pre-trained on RGB-based ImageNet is used.

Objective is to show that current methods of transfer learning from RGB images dataset (ImageNet), to a single-channel image problem domain like that of satellite images classification, are not effective enough and that use of single-channel pre-trained model can show better performance in such domains.

1.3 Research Objectives

Firstly, a model pre-trained on RGB images from mini-ImageNet and EuroSat combined will be used to train on our single-band images and performance will be noted. These single-band images are extracted for six of thirteen available bands of Multispectral images in EuroSat data. A second identical model pre-trained on grey-scale ImageNet and EuroSat combined data, will be trained and performance change will be measured again.

1.4 Research Methodologies

The research methodology applied here is quantitative. A systematic empirical investigation is performed, and mathematical models will be built using CNN and Resnet50 as statistical tools. Performance of these three models will be plotted on a graph for comparison. In addition to this, secondary research was performed, as summary from existing research and datasets already exists and I have systematically reviewed existing literature in order to synthesize my research idea.

1.5 Scope and Limitations

The scope of this work is to study whether single channel image features are better than RGB features at transferring useful knowledge to networks learning multispectral features over a small labelled dataset, using a Resnet50 architecture to learn features on ImageNet and EuroSat RGB and single channel data. Limitations of the study conducted are –

- Results from the study might not reflect characteristics of models trained on single channel images other than Greyscale.
- Resnet50 is a large neural network architecture, with skip connections, and hence requires many epochs and a large number of images to train on single channel or RGB image data. This limited the amount of computational experimentation and hyper-parameter tuning that was possible.
- This study has been performed has been done on thirteen bands contained in multispectral images of EuroSat dataset. Satellite imaging extends to Hyperspectral analysis which consists of over hundreds of bands or channels. This research might not extend to hyperspectral imaging domains.
- The study has been conducted over Land-cover images taken up by Sentinnel-2 space mission. Other target classes in satellite imaging, like large water bodies or cloud formations are not covered in this study.
- The Resnet50 architecture is used for Modeling over a combination of random subset of ImageNet and EuroSat satellite images, but the model still might have a biased inherent to the type of images used in ImageNet dataset.

1.6 Document Outline

The remained of this document is structured as follows.

1.6.1 Chapter 2: Literature Review

Review of existing literature chapter focuses on a thorough review of research that is already done in relation to transfer learning and remote sensing relevant to this dissertation. The chapter first explains what satellite imaging is and what multispectral images are. It goes on to explain current state of the art approaches in image analysis

and the rationales behind them. Later on, the chapter focuses on explaining the two datasets that are used in this study and the reasoning behind them. It also explains the limitation a small dataset size poses to deep learning approaches of problem solving, the work done by other researchers in remedying this issue of small data size. One such approach is of Transfer Learning, in which a larger network pre-trained on much larger dataset is used to fine tune a much smaller dataset on a similar task. This and other such approaches are also included in this chapter.

1.6.2 Chapter 3: Experiment design and methodology

This chapter discusses the two datasets in details, the reasoning behind picking them specifically, how they are merged for training and validation purposes and how greyscale and band wise images are extracted from the two datasets. This study uses the Resnet50 architecture to create an image analysis model from scratch over RGB and Greyscale image datasets. The chapter discusses this architecture and reasons behind using it this study. This chapter also talks about data augmentation and data pre-processing of images. Details about evaluation metrics used is also included in the chapter.

1.6.3 Chapter 4: Implementation and Results

This chapter discusses the details about implementation like layers, activation functions, optimizers used, epochs taken by individual models and validation loss obtained at convergence. Associated results with each version of the model is also discussed.

1.6.4 Chapter 5: Evaluation and Discussion

The factors that have attributed to the results obtained in the previous section are discussed here. Analysis of results related to every band is also included in this chapter. A swift comparison is been made between transfer learning obtained by RGB and Greyscale models. A decision about the acceptance or rejection of proposed hypothesis will be made based on this analysis. This chapter also outlines strengths and weaknesses of the research study.

1.6.5 Chapter 6: Conclusion and Future Work

This chapter summarize the work done in this study and the findings that have been made. It also includes recommendation for the fellow researchers who are working in the same domain or are working with spectral data with smaller data sizes. Lastly, some ideas and limitation for further research have been proposed and discussed.

2 LITERATURE REVIEW

2.1 Revolution in Image Analysis

Image classification tasks performed on ImageNet have already attained performance better than human levels (Szegedy, Ioffe, Vanhouchke & Alemi 2016) using Residual and Inception CNN networks. The ImageNet project or dataset is an ongoing research effort by Princeton university, aimed at providing researchers around the world with easy access to a large natural image database (Deng, Dong, Socher, Li, Li, & Fei-Fei, 2009). ImageNet is a database of over 15 million coloured (or RGB) images in over 22,000 classes and hence deep convolutional neural networks trained on a subset of it, gives very high accuracy (Krizhevsky, Sutskever & Hinton, 2012; Sermanet et al., 2014). Also, see figure 2.2, in page 10, for sample images from imagenet database.

There are other very famous labelled image datasets as well – like NORB (LeCun, Huang, & Bottou, 2004), Caltech 101/256 (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012; Griffin, Holub, & Perona, 2007), and CIFAR-10/100 (Krizhevsky, 2009). All these datasets have not more than a couple of hundred of images in each class. The key reason behind the vast popularity of ImageNet and also behind choosing for this study is that it is a huge dataset with objects in realistic settings and classes vary in nature from a fish to a clothing accessory, or from a dog breed to lake front, or from a furniture item to a deep sea mammal.

In 2012, two PhD students, Alex Krizhevsky and Ilya Sutskever proposed a deeper and wider Convolutional neural network model, famously known as AlexNet, as compared to the then state-of-the-art LeNet (LeCun, Bottou, Bengio, & Haffner, 1998). LeNet was the first popular CNN architecture, while AlexNet won the most difficult image classification challenge based on ImageNet database, for visual object recognition called the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 (Berg, Deng, & Fei-Fei, 2010). AlexNet was a significant leap in the field of image analysis and deep learning and is commonly referred as the point in history where

interest in and applications of deep learning increased rapidly. Figure 2.1, in page 10, shows the architecture for AlexNet.

Convolutional Neural Networks have proved to have a high learning capacity and they are greatly generalizable. One can increase the number of nodes or processing window or striding window, depth, and breadth of CNNs very easily. Yann LeCun and his team did a lot of pioneering work in establishing CNNs as default networks for Computer Vision problem domains (Jarrett, Kavukcuoglu, Ranzato, & LeCun, 2009; LeCun, Huang, & Bottou, 2004; LeCun, Kavukcuoglu, & Farabet, 2010). Later on, AlexNet established a new state of the art with this network. In addition to using CNNs, their paper pioneered several novel and highly effective strategies in running neural networks over large datasets, for example, like ReLU, Dropuout and GPU based architecture of running models.

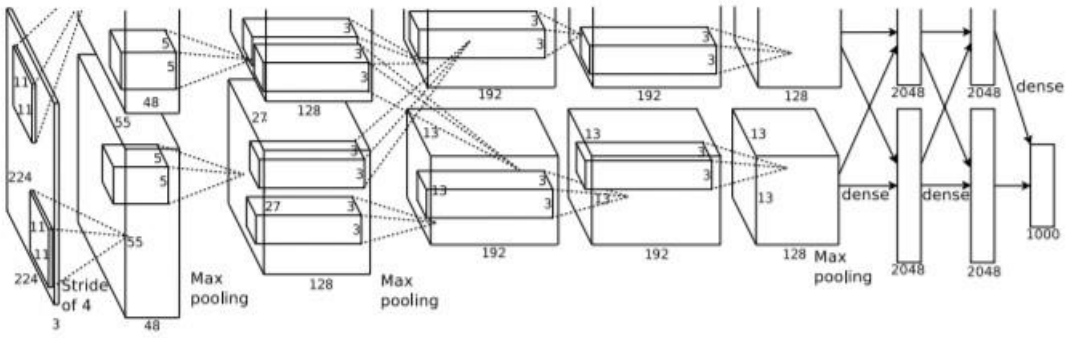

Figure 2.1 AlexNet Architecture showing Pooling, Strides and Densely connected layers (Krizhevsky, Sutskever & Hinton, 2012)

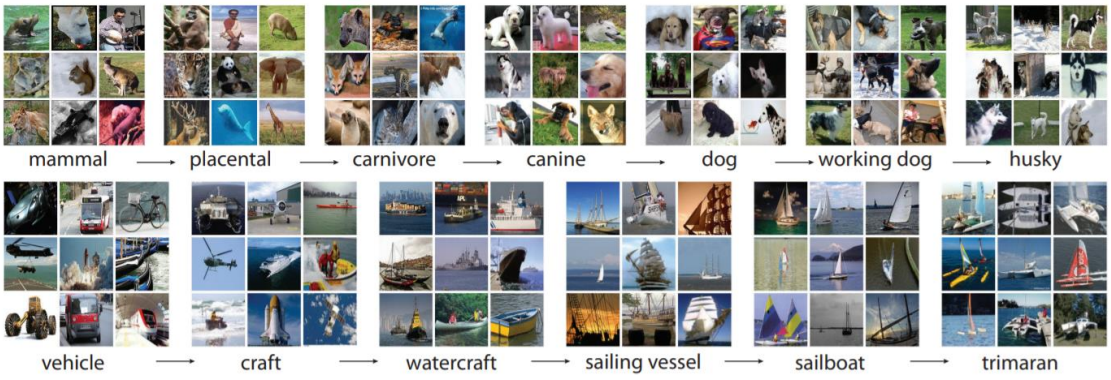

Figure 2.2 A sample of ImageNet images (Deng et. al., 2009)

Imagenet ILSVRC became both, a battleground as well as a fountain of novel state-of-the-art approaches in the computer vision domain. The following year, in 2013, another CNN model known as ZFNet, was crowned as the winner as it was further able to reduce the classification error rate to 11.2% (Zeiler & Fergus, 2013). This paper was more of a fine-tuning of AlexNet, but they laid foundations of effectively visualizing CNNs and made the intuitions behind working of CNN very clear to the computer vision community.

In summer 2014, VGGNet, came very close to winning the ILSVRC 2014 and showed the world what impact a deep network can create in an image classification problem. They were able to reduce the error rate further to 7.3% (Simonyan & Zisserman, 2015). This paper made the path for the deeper networks of future. The state-of-the-art performance reinforced the belief that a deeper network is better able to learn the image features. VGG implementations like VGG16 and VGG19 are often used in image transfer learning solutions.

GoogLeNet (Szegedy *et. al.*, 2015), which won ILSVRC in 2014, and they made the CNN network still more deeper, the competition winning configuration had 22 layers. They reduced the error rate of top-5 classifications to 6.7%. Instead of stacking the CNN layers on top of one another, the paper took an approach of stacking modules known as Inception modules one after another. Inside an inception module, all the operations like convolutions, max pooling *etc.*, happen in parallel, as can be seen in the figure 2.3, in page 12. This is why this network is also known as Inception Net.

Then in ILSVRC-2015, the winning entry was from Microsoft research team and winning entry or network was named as Residual Networks or commonly known as ResNet. This network was truly deep, with 152 layers stacked in it. The team was able to reduce the error rate to 3.6%. just like inception modules, a ResNet was build using stacked Residual blocks. This network was trained on 8 GPUs for a period of two to three weeks.

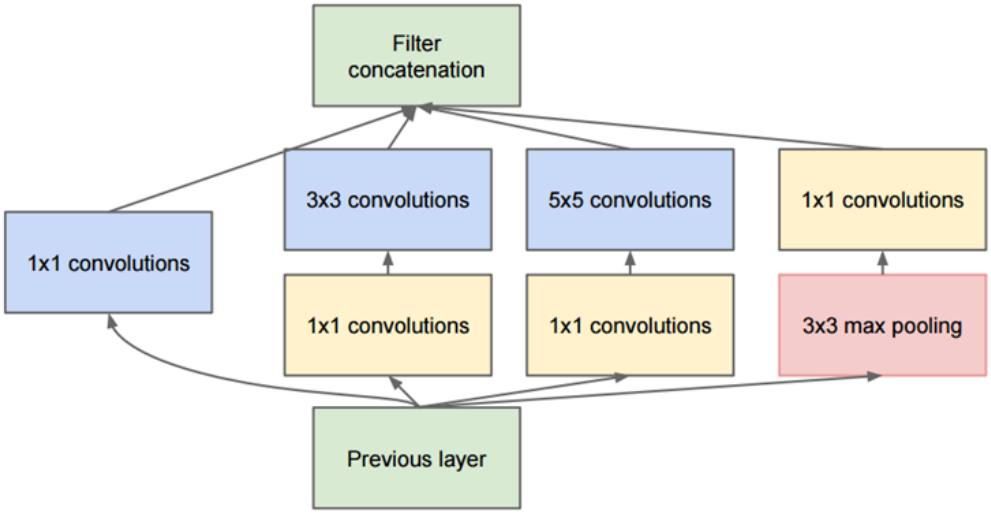

Figure 2.3 Inception module (Szegedy et. al., 2015)

2.2 Residual Networks

Resnet or Residual Networks (He, Zhang, Ren, & Sun, 2016) is one of the fastest, most widely used and generally accepted ImageNet trained deep model. This model architecture has been widely praised for its handling of problems like vanishing and exploding gradient, which crops up as a network grows deeper and deeper, i.e. as the number of layers increase. The residual architecture that won ILSVRC in 2015 had 152 network layers and attained a state-of-the art performance of 3.6 % error rate at the time. There are *Residual Blocks* in the network which are nothing, but combination of conv-relu-conv series and the network follow skip connections between these blocks. The basic idea is that during back propagation, gradient flows easily through the network without getting lost or very weak.

In their paper, authors proposed two network architectures. In first architecture, they followed an architecture similar to the then state-of-the-art VGG architecture and proposed that for the same output feature map size, same number of filters were kept, and whenever feature map size was halved the number of filters was doubled. At the end of the network, a global pooling average was used and was followed by 1000 node Softmax. In the second architecture, the authors proposed short-cut connections (shown in figure 2.4, in page 13). These shortcut connections performed the identity

mapping (F(x) + x). A random crop of 224*224 was used from the input image and standard color augmentation techniques were used. Batch normalization layers were inserted right after each convolution operation and just before activation function is realized. For propagating error, authors used Stochastic Gradient Descent or SGD with batch size of 256. They used variable learning rate which was reduced as the loss plateaued. Additionally, paper used a weight decay of 0.0001 and a momentum of 0.9 for training.

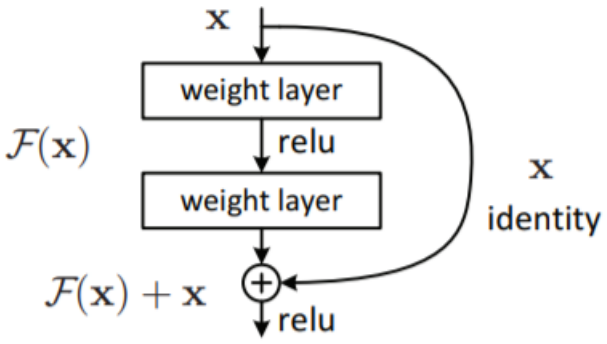

Figure 2.4 Residual blocks and Skip connections: In traditional CNNs, H(x) would be equal to F(x), but in ResNet transformation (from x to F(x)), outputs are added to the outputs of stacked layers, so adding F(x) to the input x.

2.3 Transfer Learning

If the dataset is smaller, we cannot simply apply bigger and bigger networks as was the case for the ImageNet challenge; instead, we need to apply another technique. One such way to get good accuracy is called the Transfer learning (Pan & Yang, 2010; Rusu et al., 2016; Mikolov, Joulin & Baroni 2015; Torralba & Efros, 2011). Large neural networks trained on large image datasets show that network's first layer learns features similar to Gabor filters or color blobs.

The initial layers carry information like what are the location of edges, boundaries, corners, and shapes present inside an image. Zeiler and Fergus, 2013, were the first to analyse this empirically and visually in a very famous and highly cited paper, titled Visualizing and Understanding Convolutional Networks. They used and pioneered an

approach called Deconvolutional Network or DeConvNet and used it for visualizations as shown in figure 2.5, in page 14.

These features or information contained in the initial layers are not specific for any task, rather are general in their nature and are applicable for all sorts of target images and tasks (Yosinski, Clune, Bengio & Lipson, 2014). In this landmark paper, the authors worked towards establishing some of the key concepts of transfer learning like – features transition from general to specific from first few layers to last or final few layers, features are not highly transferable to a distant target task, and lastly any kind of transfer is better than random initialization even when the source and target tasks are not that similar.

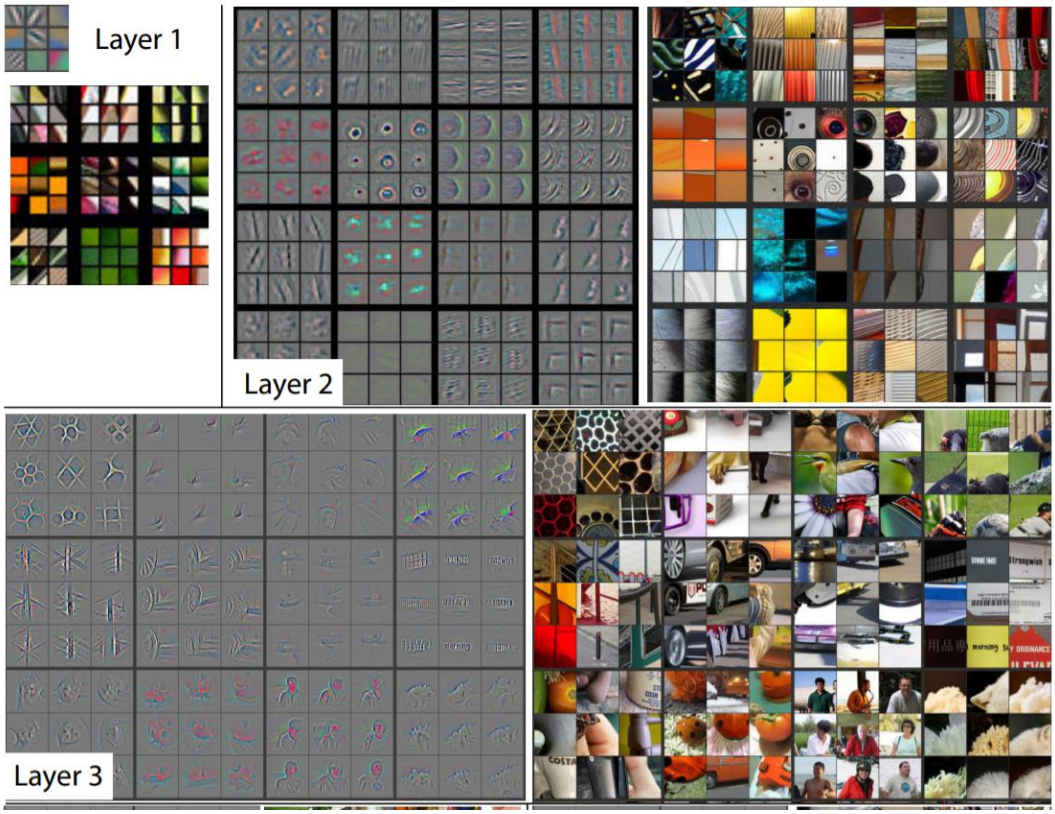

Figure 2.5 Visualization of features learned in a fully trained deep network. please note, how features are enriched as we go up the layers in a deep convolutional network

Generally, the target task is much smaller than the source task, hence overfitting due to training a large network is avoided. Transfer learning is of two types – one in which pretrained features or the features learned from the source are not touched or are *frozen*

and only the new layers of target task are trained on the target data. In the second, source layers are allowed to train, or the error is back-propagated from target task layers through the trained source layer and their learned weights are allowed to be *fine-tuned*. If the target dataset is very small, and source weights are allowed to be fine-tuned then there is a risk of overfitting. And if the target data is large enough, it is always advisable to train from scratch or go for fine-tuning.

Donahue et al., 2014, proposed a convolutional architecture for transferring features between two seemingly different tasks, and named it as Decaf network. Authors empirically validated that generic visual features outperforms a host of conventional feature coding or representations on most of the benchmark datasets like Caltech-101. Models thus pretrained on large datasets like ImageNet have given great results lately by training new models on the already extracted features, both on supervised and unsupervised machine learning domains (Razavian, Azizpour, Sullivan & Carlsson, 2014; Radford, Metz & Chintala, 2016; Zhuang, Cheng, Luo, Pan & He, 2015). Sermanet et al., in 2014, came up with a brilliant architecture named as Overfeat, where they used an Imagenet trained model as a Feature Extractor. This feature extractor was used as a sliding window over the images and object detection task as accomplished using it. This paper established that CNNs have the intrinsic capabilities of learning general to specific features of an image and that these features are transferable.

There are other methods as well of doing transfer learning, like making the representations in both source and target domains similar so that knowledge can be transferred seamlessly between them (Daume III, 2007; Sun, Feng & Saenko, 2016; Bousmalis, Trigeorgis, Silberman, Krishnan, & Erhan, 2016; Tzeng, Hoffman, Zhang, Saenko & Darrell, 2014; Ganin & Lempitsky, 2015; Ganin et al., 2016). Transfer learning can be done using unlabelled or very less labelled data in target domain too (Zhu, 2005).

Efforts have been put in to understand how neural networks are able to generalize well and how to make them more robust (Zhang, Bengio, Hardt, Recht & Vinyals, 2017; Kurakin, Goodfellow & Bengio, 2017). In recent publications, there have been attempts to implement few-shot, single-shot and zero-shot (very few source instances

to train on) transfer learnings and it has proved that common gradient descent based approach suits mainly when source domain data is large and that using an innovative LSTM based optimizer can work better in cases with small source instances (Ravi & Larochelle, 2017; Xian, Schiele, Akata, Campus & Machine, 2017).

2.4 Data Augmentation

Small labelled datasets can also practice data augmentation to increase the dataset size and thus improve the model fitting over the data (Perez & Wang, 2017). This problem is especially common in multispectral domains like medical imaging and satellite imaging.

Data Augmentation also helps in reducing overfitting as it increases the training data. It helps in increasing the dataset size by either warping or oversampling the data. This process makes sure that the labels are preserved during the transformations. Most general form of image augmentation includes data warping techniques like geometric and color transformations, like obtaining a new image by cropping, flipping, sheering, or inverting an image (Chatfield, Simonyan, Vedaldi, & Zisserman, 2014), please refer figure 2.6, in page 17 for few samples of geometric transformations. Oversampling includes mixing of two images to synthetically increase the data size. This method might not be able to preserve the labels. Also, note that oversampling helps in dealing with the problem of class imbalance by increasing synthetically the data in a class. This helps in making model less biased towards a class that has higher representation in the training data.

Furthermore, image augmentation not only has applications in increasing the size of the labelled dataset, but also helps in making the model more generalizable for real world tasks. Generalizable models are able to perform well on target datasets that are entirely new to their training datasets. Poorly generalizable models tend to overfit on the data they have been trained on. This is the principle reason why using a validating dataset is a must while training a large network over smaller datasets. To build effective models validation errors must be continuously monitored. There should be a simultaneous decrease in training and validation error. Augmentation helps in attaining

this target to a great extent, by providing to the training model much of the possible variations and distributions of datapoints or features. This minimizes the distance or difference between training and validation sets.

Figure 2.6 Traditional Transformations or Data Augmentation for images

Augmentation in computer vision problems has been happening over the last couple of decades and was first seen in LeCun, Bottou, Bengio, & Haffner, 1998. They practised data warping techniques to distort the hand-written digits in training datasets. After that, the iconic AlexNet also used image augmentation techniques to reduce the error rate by 1%, and also increased the data size by a huge 2048 times. New images were created by clipping the training images followed by random flipping and later on using a PCA based color augmentation.

Another very interesting and upcoming technique of image augmentation uses generative adversarial networks or GANs, which were introduced by Goodfellow et. al., in 2014, and Neural Style Transfer and Neural Architecture Search or NAS introduced by Gatys, Ecker, & Bethge in 2015 and enhanced by Zoph and Le in 2017. These two approaches have found use in two of the most promising and useful augmentation techniques currently – Smart Augmentation (Lemley, Bazrafkan, & Corcoran, 2017), and Autoaugment (Cubuk et. al., 2019). Smart Augmentation merges or blends two or more samples within the training data, while keeping the label information saved, based on expanding the network accuracy or minimizing the loss. While Autoaugment selects best possible augmentation method based on a search policy which aims at increasing the validation accuracy. The algorithm has obtained the state-of the-art on CIFAR-10 and ImageNet datasets.

2.5 Satellite or Remote Sensing

Earth Observation (EO) and Remote Sensing (RS) are fairly new computer vision sub-domains which have received attention from researchers around the world. The data from these domains has the capability of bringing about significant improvements in agriculture in developing countries. Use of RGB and Near-Infrared region images using low-cost and low-orbit observational systems in estimating produce and mapping the plantation areas has been advocated as best practises (Ponti *et. al.*, 2016). Researchers have also used satellite images coupled with social media images in detecting the sections of roads and urban areas covered under flood waters (Bischke, Helber, Schulze, Srinivasan, & Borth, 2017). Like these two, there are several applications of satellite data, which includes both military and civil usage.

The problem with these Earth Observation datasets is that firstly, there are very few and very small sized labelled datasets available and secondly, image features in these datasets are quite different from those from natural image datasets, which have images like cat, dog, fish, scorpion, car, truck, house, ship *etc*. Principal examples of Earth Observation or Satellite imaging datasets include – UCMerced dataset (Yang & Newsam, 2010), PatternNet (Zhou, Newsam, Li, & Shao, 2018), and NWPU-RESISC45 (Cheng, Han, & Lu, 2018). The UCMerced dataset is a fairly small dataset when it is considered for usage in building deep learning models as such models historically need large amount of data to train and predict correctly. UCMerced has 21 land-cover classes with 100 images per class, with 256*256 pixel dimensions and all images are from the RGB color space. Likewise, the other two datasets too have images in the few hundreds for every class label. Moreover, these datasets have images which are already processed and high resolution, thus this does not represent the real-world scenario of Remote Sensing images.

In a supervised problem-solving approach, the performance of a classifier depends on the size and quality of a suitably labelled training and validation dataset. Razavian, Azizpour, Sullivan & Carlsson, 2014, suggested in their paper that deep networks learn features that can be treated at par or even better than the traditional methods in the Computer Vison field like, GIST (Oliva & Torralba, 2001) and BIC (Stehling,

Nascimento, & Falcao, 2002). Going by this observation, that deep features can generalize better than other means from one dataset or task to another, of late several attempts have been made using pre-learned deep models to learn multispectral and in general Earth Observation (EO) or Remote Sensing (RS) data. (Penatti & Nogueira, & Santos, 2015; Nogueira, Penatti, & Santos, 2017; Castelluccio, Poggi, Sansone, & Verdoliva, 2015; Xia *et. al.,* 2016). All these studies have performed classification upon EO or RS datasets, using ImageNet trained large deep networks like Overfeat (Sermanet *et. al.*, 2014), CaffeNet (Jia *et. al.*, 2014) GoogLeNet (Szegedy *et. al.*, 2015) and others like them.

2.6 EuroSat: A Novel dataset

EuroSat data (Helber, Bischke, Dengel, & Borth. 2017) is a collection of multi spectral (thirteen bands) and RGB data captured by Sentinel-2A satellite to address the challenge of identifying land-use and land-cover categories in European countries. This dataset has 27000 labelled images classified into 10 land-cover classes. It has two sets of image data, the first one contains RGB color-space images, while the second one has multispectral images consisting of 13 frequency bands. This dataset is made freely available for both commercial and non-commercial purposes by the European Space Agency (ESA).

In their dataset-introducing paper, authors have performed two steps – firstly satellite images of 34 European countries was collected, and secondly, this data was divided into 27000 images of 64*64 pixel size, which were georeferenced and labelled with proper landcover classes. The 34 countries are chosen to create a wide variety of land cover samples, for example these countries include - Austria, Belgium, Cyprus, Denmark, Finland, France, Germany, Greece, Iceland, Ireland, Luxembourg, Netherlands, Norway, Poland, Portugal, Romania, Switzerland, Ukraine and the United Kingdom. The authors have also made sure that different types of sample in a particular landcover class are included. For example, different types of forests in Forest class, different types of river flows in Rivers class, and different types of industry structures in Industrial class, and so on. Moreover, this data is collected over

the year and so different lighting conditions have been accounted for in the representation. Sample of images can be seen in figure 2.7 and 2.8.

Figure 2.7 (a) Industrial (b) Residential (c) Annual Crop (d) Permanent Crop (e) River

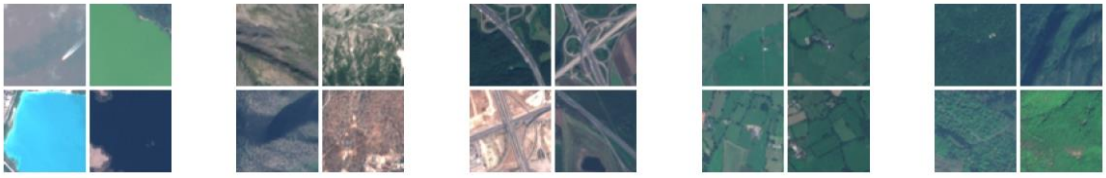

Figure 2.8 (f) Sea lake (g) Herbaceous Vegetation (h) Highway (i) Pasture (j) Forest

The authors have established that ResNet50 has given them the highest benchmarking performance in the land-cover classification task. Furthermore, the RGB color space outperformed all thirteen bands as well as band combinations like shortwave-infrared and color-infrared, when ResNet50, trained in ILSVRC datasets which consists of nothing other than RGB images.

The main applications that the paper suggests includes Land Cover Change Detection, in which a classifier can be trained to detect any changes in the observed land or sea portion over a period of time. A change can be defined as a classifier assigning a new class to the same patch of the image, for example, a forest area might be converted into annual Crop area, thus indicating deforestation. Other applications proposed is, providing assistance in mapping for an area under observation.

2.7 Summary, Limitations and Gaps of Literature

A detailed literature review of the state-of-the-art computer vision approaches was made. Research papers observing challenges with operating on smaller datasets, and transfer learning applications and current limitations were also reviewed. Lack of

availability of labelled remote sensing and earth observation data is observed to be a major issue hampering any research work in satellite imaging problem domains. Literature exploring various aspects of remote sensing and its applications, is studied as part of solving the problem of transferring relevant features to multispectral classification tasks.

The literature review also pointed out at the revolution that happened in the Computer Vision field with the advent of AlexNet in 2012. The most influential approaches in the architecture explained that CNNs are the best way to handle image data. The relevance and possibility of transferring learned features to another problem area has been a key development in the computer vision field over the last decade.

Deep learning can also be applied over smaller labelled datasets by using data augmentation methods. A brief review was done on key approaches and the benefits of augmentation specially in the sense of real-world tasks. One such task is of identifying land-cover classes in remote sensing images. This is essentially useful in cartography efforts of private and government bodies. This review also talks about current and most prevalent EO datasets and also how classification is been done in present times, using pre-trained deep networks like GoogLeNet, Resnet, CaffeNet *etc*, and other current benchmarks.

Apparently, there are no research papers or any publication until now, which talk about training a deep industry grade network like Resnet50 on an image dataset which is not from the RGB color space and then using that network to *transfer* features to another task from a completely different band or channel space. Limitations and research gap identified through the literature review can be addressed by the research question introduced in the Introduction chapter –

"To what extent a CNN neural network, pre-trained on single channel (grey-scale) Imagenet and Eurosat** images, can improve the image classification accuracy of multispectral images."*

The following chapters provide the research design, experiment methodology, implementation, and evaluation of experiment to address this very research question.

3 EXPERIMENT DESIGN AND METHODOLOGY

The purpose of this study is to test whether single channel features are better than RGB features for models trying to learn multispectral data. RGB images are combined from mini-Imagenet and EuroSat datasets. For single channel features, the greyscale images are used instead. These greyscale images are the same as the RGB ones, except they are converted to greyscale using image augmentation methods prior to the training process. Multispectral data consists of satellite captured multiple band images. The dataset that the study uses has TIFF images consisting of thirteen spectral bands each. Single bands are extracted from these tiff images and thus every image got divided into thirteen different band images. These bands are characterised as B1 to B13. This chapter outlines the data preparation and processing steps in detail.

Along with this, model architecture and evaluation criteria are also presented. An explanation of how convolutional network works is presented for better understanding of the experimental design. Experiments are conducted using the ResNet network, which has a CNN as its main building block. How Resnet network is used in the experiments, is explained in detail using illustrations. No experimental methodology is complete without an evaluation plan. Thus, the performance of all the models has been evaluated over validations sets and this part of the section is well documented at the end of this chapter.

All programming is done using Python 3.7 and some of the useful libraries that are used in the implementation include NumPy, Rasterio, TensorFlow2.0 and Keras, among others.

3.1 Design Methodology

A detailed overview of the plan and design of the experiment is elaborated in this section. To help with the basic understanding of the project please refer the figures in this section (figure 3.1 – figure 3.11). There are two sources of RGB and Greyscale images, one is the mini-ImageNet dataset and another one is EuroSat dataset. Details

about data preparation steps on these datasets is in section 3.3. Process and work-flow is covered in section 3.2.

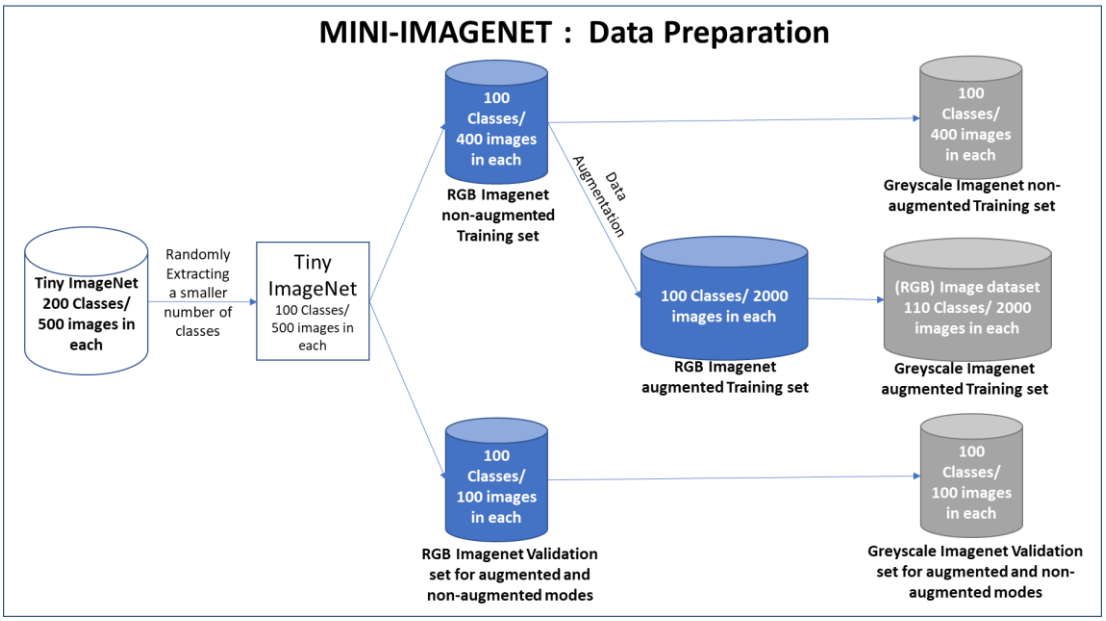

Figure 3.1 Creating mini-ImageNet Training and Validation data as two sets of Augmented and non-Augmented images for both Greyscale and RGB color-space

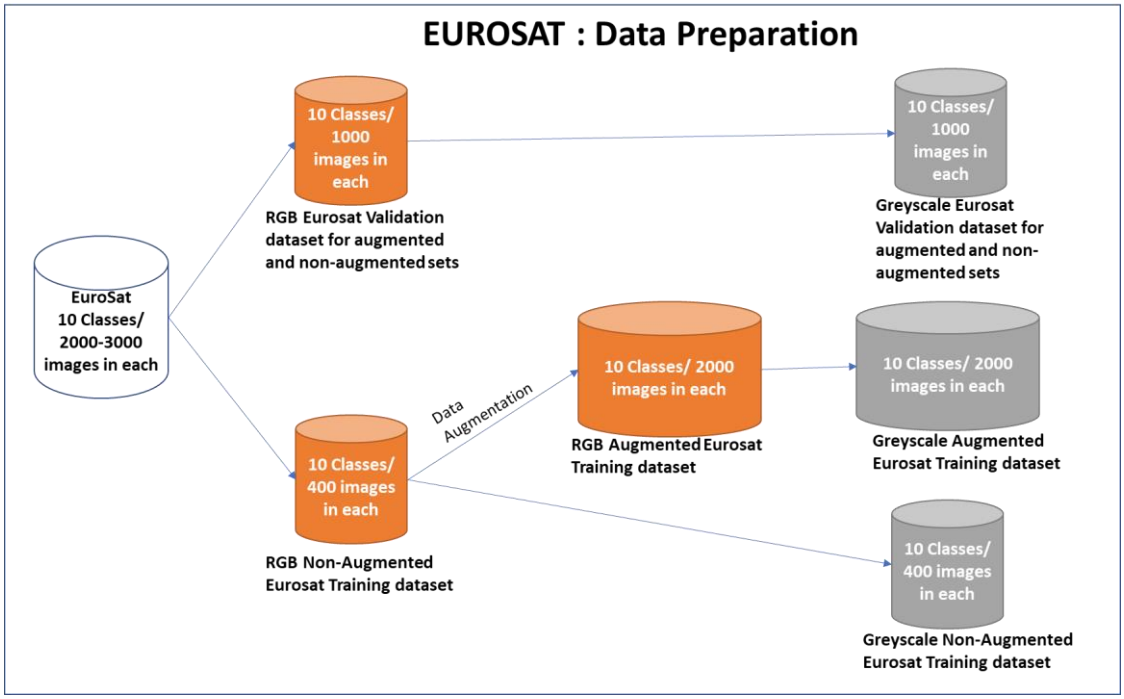

Figure 3.2 Creating mini-ImageNet Training and Validation data as two sets of Augmented and non-Augmented images for both Greyscale and RGB color-space

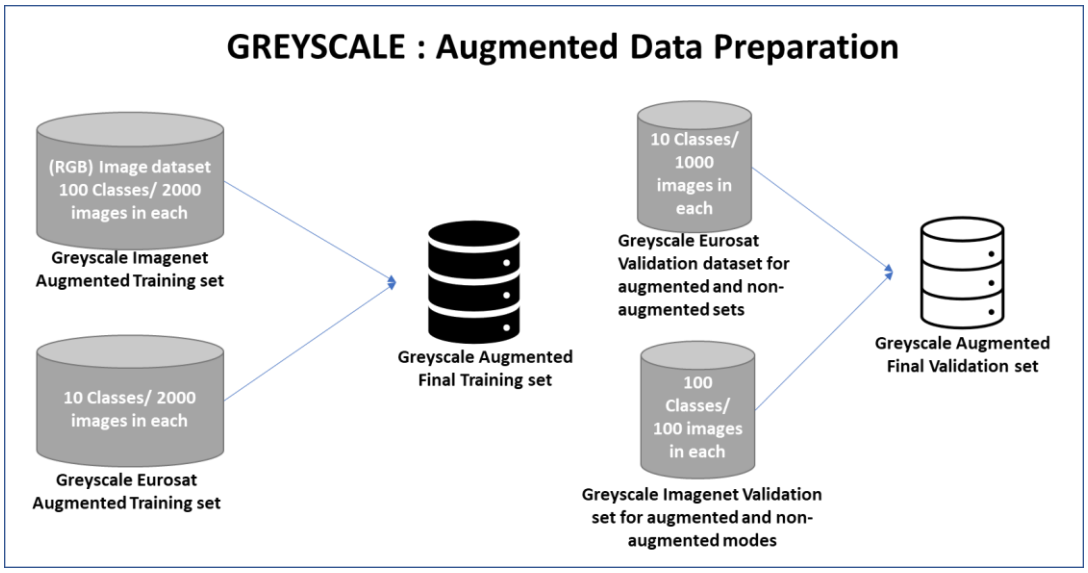

Figure 3.3 Merging Eurosat and ImageNet Greyscale augmented images to create final sets for base models Training and Validation

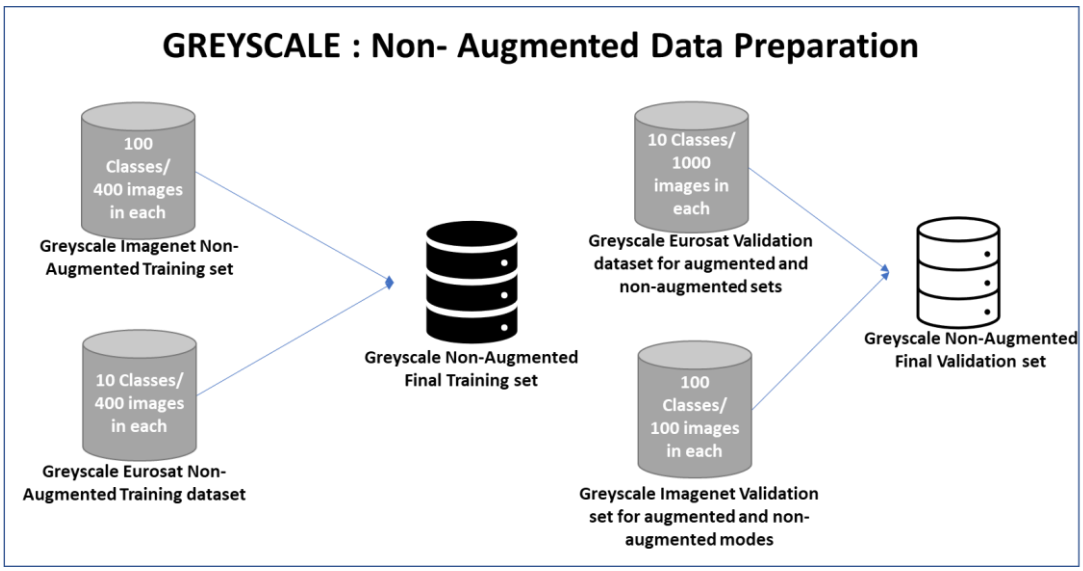

Figure 3.4 Merging Eurosat and ImageNet Greyscale non-augmented images to create final sets for base models Training and Validation

Design Methodology is such that first new models are created using Resnet50 architecture by training them from scratch over datasets from two color spaces, namely RGB and Greyscale. Also note that, there are two types of datasets in each category, one is smaller in size and consists of Non-Augmented images, while the other one is larger and consists of Augmented images. So, four Resnet50 architecture-based models were created by training from scratch on RGB and Greyscale datasets independently, please refer figure 3.7 in page 26. These four pre-trained models are then used to

transfer features or are finetuned on target images of individual bands, i.e. band B02, B03, B04, B05, B08 band B12. The performance is recorded on Tests sets and a comparative analysis is made on the outcomes (figures 3.9, 3.10, and 3.11, in page 27). This will make it four sets of Test accuracies and F1 scores; two are for RGB based feature transfer and another two for Greyscale or single channel-based feature transfer.

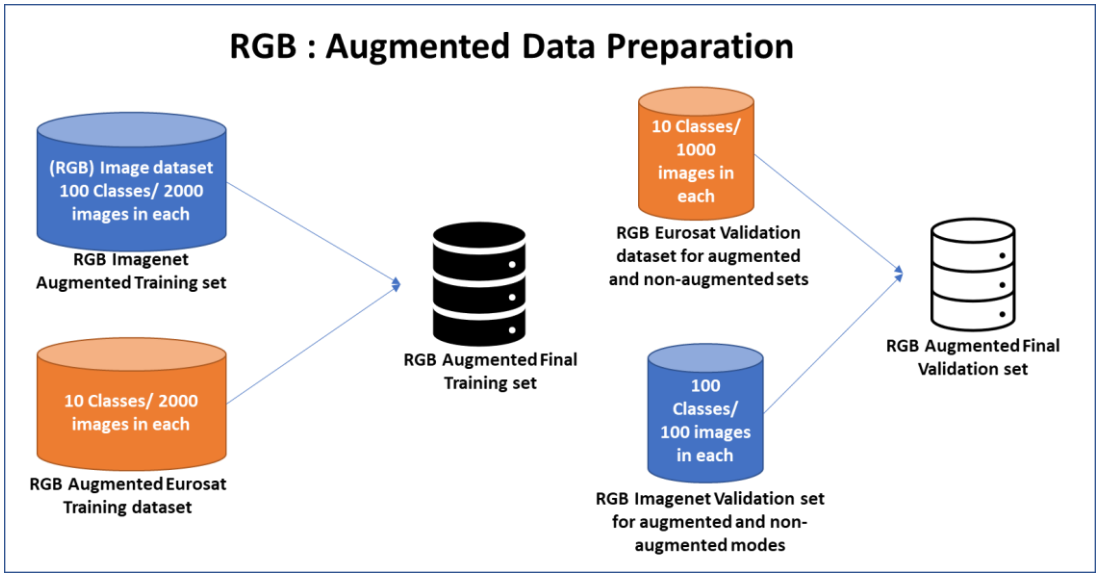

Figure 3.5 Merging Eurosat and ImageNet RGB augmented images to create final sets for base models Training and Validation

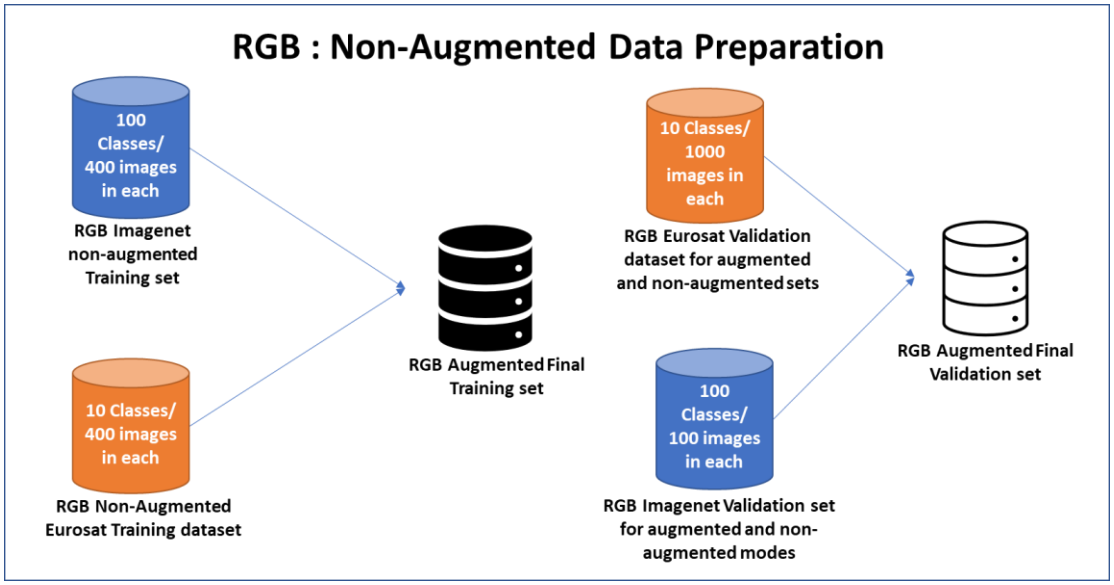

Figure 3.6 Merging Eurosat and ImageNet RGB non-augmented images to create final sets for base models Training and Validation

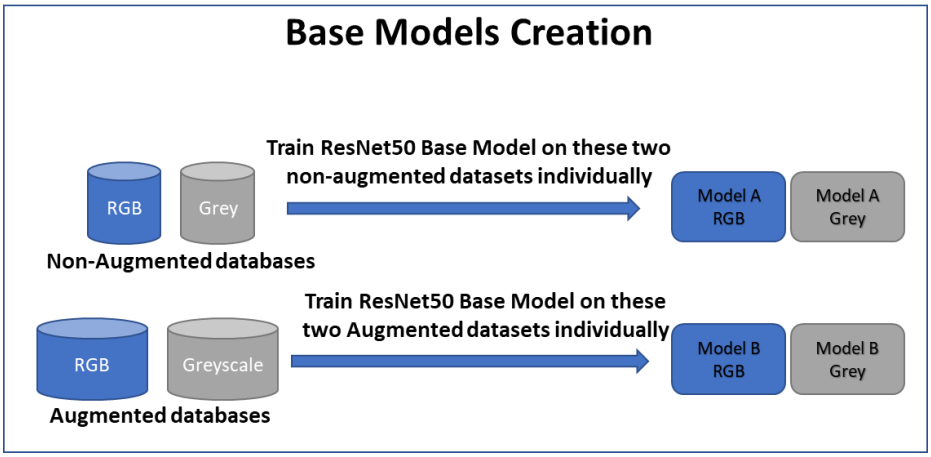

Figure 3.7 Training to create four Base Models

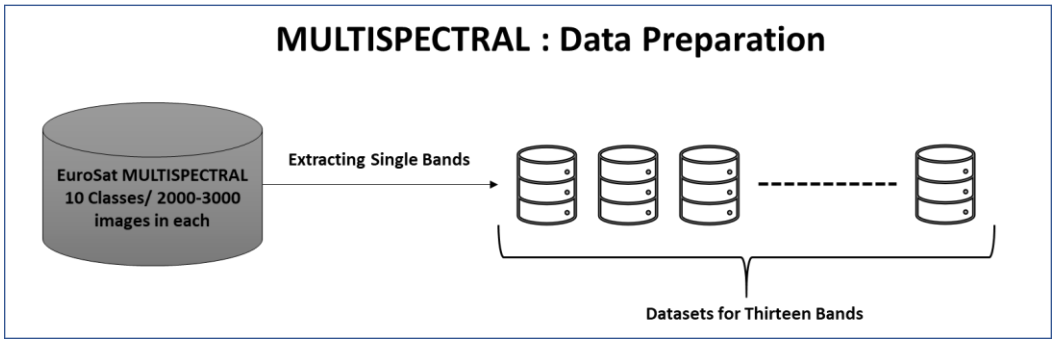

Figure 3.8 Data preparation for Multispectral images

3.2 Business Understanding

The focus of this research study is to improve the general performance of computer vision problems in the Multispectral image domain. Currently, most of the multispectral and hyperspectral domain problems are addressed using deep models like Resnet50, which have been trained in the RGB feature space. The input to an already trained Resnet type network will be three channels, each designated for individual bands of Red, Green and Blue. While applying this pre-trained network to solve multispectral domain problems, their individual spectral bands are fed as an input to the three channels of the Resnet50 model. The model then fine tunes its learned weights over these smaller target multispectral datasets. Learned features are said to be *transferred* to the new and smaller task.

The Null hypothesis can be stated as –

Null Hypothesis, Ho: If a deep neural network trained on greyscale images is fine-tuned on a dataset of single frequency images extracted from multispectral images taken by satellites, to classify images into one of the ten land-cover classes, then the accuracy is higher than both, when the model is trained from scratch and when a model pre-trained on RGB-based ImageNet is used.

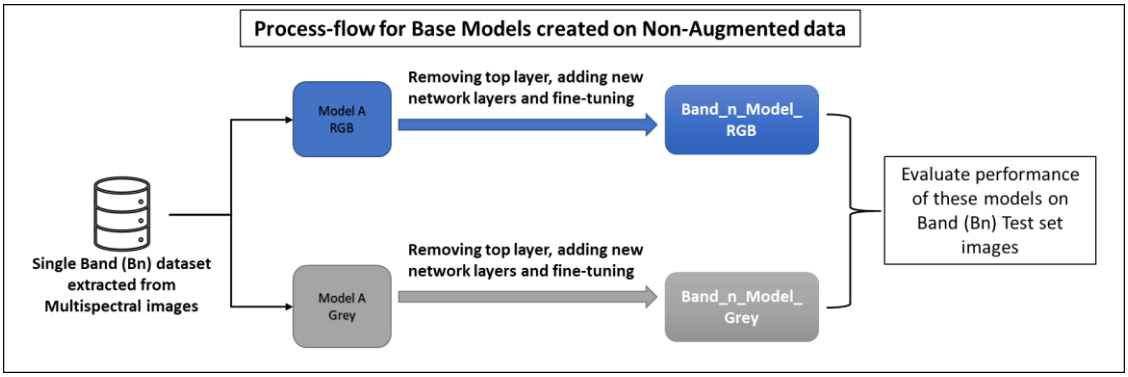

Figure 3.9 Process-Flow Diagrams – Non-Augmented

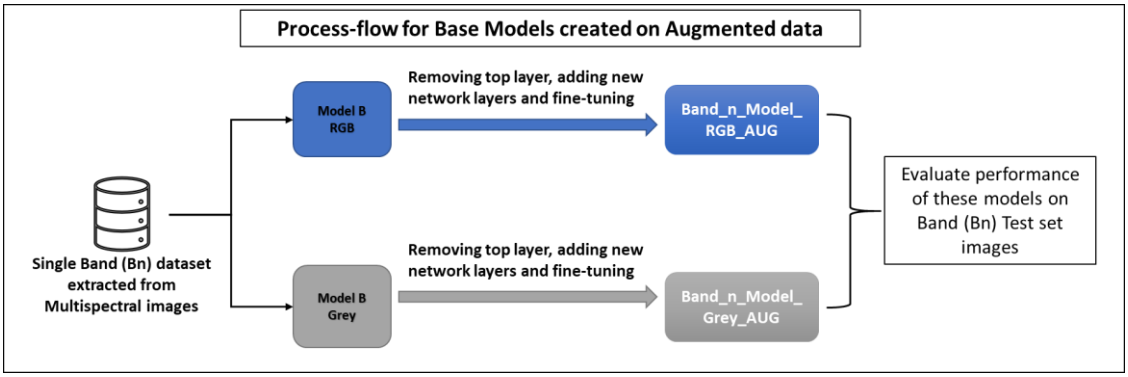

Figure 3.10 Process-Flow Diagrams – Augmented

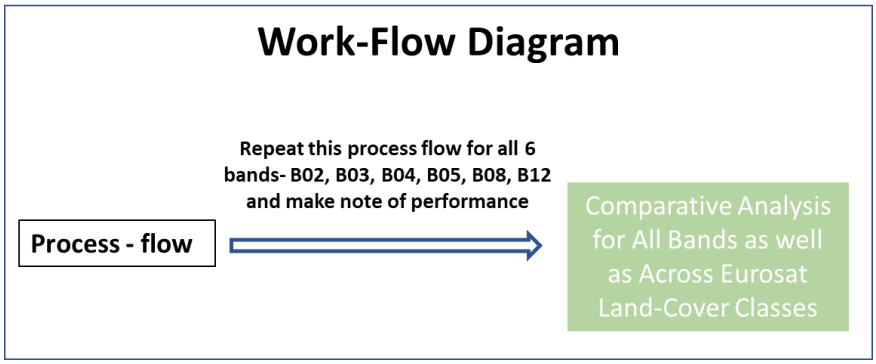

Figure 3.11 Work-Flow diagrams

3.3 Data Understanding

The experiments are conducted using multiple sets of images, where each set has distinctive features and are as per the design of research. Each image dataset is explained elaborately in below sections.

3.3.1 Dataset: mini-ImageNet

Imagenet – It is a largescale ontology of images built upon the backbone of the WordNet structure. With images attached to each of the 5247 synsets of Wordnet, there are around 3.2 million images in total. These images are very diverse and accurate to their description and are aimed to help computer vision researchers in their efforts (Deng et. al., 2009). Due to the limitations of time and computational power, a very small subset of this dataset is used for the purpose of this study. *Mini Imagenet* has 500 images each in its 100 overall classes, and each image is of the height 64 pixels and width 64 pixels. Some samples are shown in figure 3.12, in page 29.

3.3.1.1 Data Processing: RGB images

Since the images are only five hundred in each class, data augmentation is used to increase the size of the dataset to two thousand images in each class (also refer figure 3.1, in page 23).

Data augmentation is done using Keras ImageDataGenerator module, and basic geometric transformations are applied to the images. These transformations include, Shifting the image across its width and height, Shearing or tilting the image along one of the axis, Zooming in and out of the images, Flipping the image either horizontally, and lastly by Rotating the images by not more than 90 degrees at a time. Some sample images post augmentation is shown in figure 3.13. Few things can be observed here - while shifting the images, the *last* pixels are copied to fill the gaps created by the process. Also, since some of the classes in mini-Imagenet dataset are related to humans, like clothing, houses, monuments, back-packs and then some are related to land animals like cats, dogs, bears *etc.*, thus it doesn't make sense to vertically flip these images while doing the augmentations.

Figure 3.12 Mini-Imagenet image samples. Each image is of 64*64*3 dimension, where 3 stands for the three color channels of Red, Green, and Blue

Figure 3.13 Augmentations specified are applied at random to a given image. Every image is augmented to five of its type, to inflate the dataset size to 2500 images in each class.

3.3.1.2 Data Processing: Greyscale images

Greyscale images are extracted using ImageDataGenerator module of Keras and for every augmented image, an equivalent greyscale version is created. At the end of this processing step, there are exactly the same images as RGB set, except that they are all in greyscale (figure 3.1 and 3.2, both in page 23). Some samples of greyscale images thus obtained are shown in figure 3.14, in page 30). Here greyscale is used to represent the idea of single channel and these images will be used to train Resnet50 based

network from scratch, to prepare a single channel trained classifier. This will later be used for transferring features to target domains of multispectral bands.

Figure 3.14 Some Samples from Greyscale mini-Imagenet after transformation

3.3.1.3 Training and Validation sets

Both, for RGB as well as Greyscale datasets, the 500 images from each class are randomly divided into two sets of Training and Validation with four hundred images per class in training and remaining one hundred images per class in validation. Same steps are followed to create augmented datasets too, where each training set has 2000 images and validation set has 100 to 1000 images per class. Figures 3.1 and 3.2, in page 23, clearly depict this process.

3.3.2 Dataset: EuroSat

EuroSat – has two sets of images - RGB and Multispectral. There are around twenty-seven thousand images belonging to ten classes in total. These classes are the different land-use types and they vary from Residential, Industrial, Farmland to Rivers, Forests and Pastured crops. Please refer figure 3.15, in page 31 for different classes covered their sample image. This dataset is collected by Sentinel 2A satellite in order to address the challenge or requirement of land-use or landcover estimation. It is a sun-synchronous satellite which was launched in June 2015 to cover Earth's land surface with Multispectral Imager (MSI) covering 13 spectral bands listed in table 3.1, in page 31. The four bands B01, B09, B10 and B11 are intended to be used for the correction of atmospheric effects (e.g., aerosols, cirrus clouds, water vapor or snow). The remaining bands are primarily intended to identify and monitor the different land use or land cover classes. In addition to mainland, large islands as well as inland and coastal waters are covered by this satellite (Helber, Bischke, Dengel, & Borth. 2017).

Table 3.1 Thirteen bands of Multispectral Imager, their Resolution and Wavelength

Band	Spatial Resolution (m)	Central Wavelength (nm)
B01 - Aerosols	60	443
B02 - Blue	10	490
B03 - Green	10	560
B04 - Red	10	665
B05 - Rededge1	20	705
B06 - Rededge2	20	740
B07 - Rededge3	20	783
B08 - NIR	10	842
B08A - Rededge4	20	865
B09 – Water vapor	60	945
B10 - Cirrus	60	1375
B11 - SWIR1	20	1610
B12 - SWIR2	20	2190

Figure 3.15 Ten EuroSat class, respectively from top to bottom, left to right : Annual Crop, Forest, Herbaceous Vegetation, Highway, Industrial, Pasture, Permanent Crop, Residential, River, Sea Lake

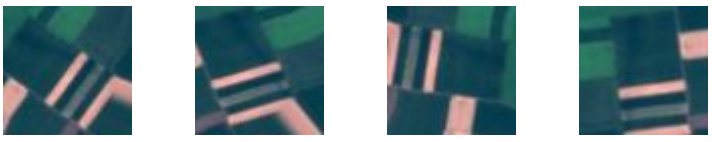

Figure 3.16 Augmented Eurosat data sample

3.3.2.1 Data Processing: Multispectral images

There are ten classes in the EuroSat data, for both RGB as well as Multispectral images, please refer to figure 3.15 for class names. In Multispectral data, the images

are stored with .TIF file extension, which makes it difficult to display using regular tools. Please refer figure 3.8, in page 26, to understand the process of extraction, and figure 3.17, to see the extracted images for all 13 bands of a sample image.

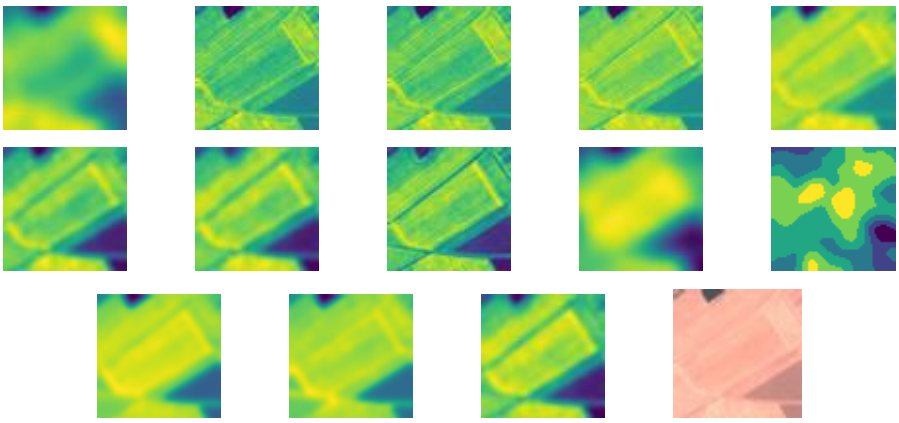

Figure 3.17 Same image extracted as 13 Bands. From left to Right, and top to Bottom - Band01, Band02, Band03, Band04, Band05, Band06, Band07, Band08, Band09, Band10, Band01, Band11, Band12, Band13, and lastly the Original RGB Image

3.3.2.2 Data Processing: RGB images

Similar to the Imagenet dataset, the images are 64 pixels in height and width. The RGB dataset is augmented in a similar manner to match the transformations in Imagenet counterparts. Here too, a vertical flip is avoided, and rotation range is kept within ninety degrees. Some sample transformations or augmented images can be seen in figure 3.12.

3.3.2.3 Data Processing: Greyscale images

A Greyscale dataset is created by transforming RGB images using the generator module from TensorFlow-Keras. These images are also 64 pixels in height and width. Some sample images from this set can be seen in figure 3.18, in page 33.

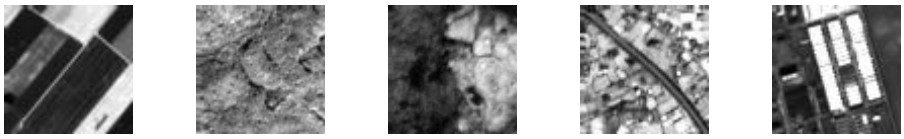

Figure 3.18 Greyscale image samples from EuroSat dataset for classes – Annual Crop, Forest, Herbaceous Vegetation, Highway , and Industrial (left to right)

3.3.1.2 Training and Validation sets

All images are selected at random for creating Training and Validation sets. For non-augmented mode, there are four hundred images in training set and thousand images in validation set for both RGB and Greyscale images. For augmented mode, data size is bigger, with 2000 images per class for training and 1000 images per class for validation sets.

3.4 Performance Evaluation

For the evaluation of models created over different sets of data, Training, Validation and Testing accuracies are used throughout the study. In addition to this, Precision, Recall and F1 Scores are constructed to compare performances between individual classes. These scores are calculated by looking at the number of misclassifications against correct classifications for EuroSat classes. Let's understand each evaluation criteria in a bit more detail –

3.4.1 Accuracy

It is a straightforward concept of estimating the correctness of a classifier and is defined as the ratio of correct predictions over the total number of predictions. Since in our case, EuroSat Multispectral data is a balanced dataset, meaning all classes have nearly equal number of distributions, Accuracy measure can be used without any concerns.

Accuracy = (True positives + True Negatives) / (True Positive s + False Positives + True Negatives + False Positives)

3.4.2 Precision

It is defined as number of correct predictions of a class divided by the total predictions over that class.

Precision = True Positives / (True Positives + False Positives)

False positives are the data points which are incorrectly labelled as target class by the model. The performance of a model is best when Precision nears to a value of 1, i.e. when FP becomes 0.

3.4.3 Recall

It is defined as the correctly predicted fraction of the target class. It can be obtained by dividing total correct predictions by total class predictions. Recall is also known as Sensitivity and its ideal value is also 1.

Precision = True Positives / (True Positives + False Negatives)

For a given class, True positives are the data points classified correctly by the model, and False Negatives are number of data points that are incorrectly classified by the model.

3.4.4 F1 Score

In an ideal model performance, both Recall and Precision would be equal to 1, or both FP and FN would be 0. A performance measure is thus needed, which can take into account both precision and recall, and can calculate a score. F1 Score is actually defined as the weighted summary of Precision and Recall or in other words it is the harmonic means of these two values. This measure is a more balanced approach than accuracy of a model, especially if there is an uneven distribution of data points between the target classes.

3.4.5 Loss

It is the summation of errors made for each sample in training or validation set. More accurately, it is the negative log likelihood and residual sum of errors for classification. The lower the loss is, better the model is at prediction a correct target class value.

3.5 Summary

This chapter has explained the design methodologies that the experiments have used. Elaborate model design shows the architecture that is used in implementing these experiments. Four base models will be used, one set of models is from RGB or Red, Green and Blue electromagnetic spectrum space and another set of models is from Greyscale or single spectrum space. The process flow of the research shows how these models will be used to transfer learned features to the target space. The hypothesis, as explained in business understanding section, concerns with which model gets a higher accuracy value in classification task on multispectral images. The classes are land-cover or land-use types.

All the datasets used in this study, their key attributes and features were covered in great details. Information included - from where the datasets are obtained, what are the transformations done upon them, how they are extracted and processed during model building and predicting, and lastly what are some of the samples. EuroSat and mini-ImageNet samples have shown how varying the datasets are. Within Imagenet there are different categories of images like animals, daily use tools, cloths etc. And then in Eurosat there are land-cover classes which range from Annual crop to Sea lake. Also, some Eurosat classes have images which are very much similar to one another, for example, images from annual crop and permanent crop classes, or those from rivers and highways. The sample images and band descriptions are there for reader's better understanding of multispectral data. This chapter has as well touched upon the creation and extraction details of new datasets, Greyscale and Individual Bands (B02-B12).

The evaluation criteria to be used in this study will monitor the effectiveness of feature transferring between different base model configurations. The next chapter of this report talks about implementation and results in this direction. Note that, this experiment design follows the hypothesis stated in this chapter and the research objectives listed at the start of the report in the Introduction chapter.

4 IMPLEMENTATION AND RESULTS

The goal of this chapter is to give implementation details of the research experiments that were described in earlier chapters. It will explain all the models that have been built and what were the results obtained from them for different spectral bands. This chapter is divided into following sub-sections –

- Model Architecture
- Results

4.1 Model Architecture

The implementation is made using Resnet50, which is a fifty-layer deep convolutional neural network. This section covers the essential details of implementing a CNN and Resnet50.

4.1.1 Convolutional Neural Network

A Convolutional Network is essentially a sequence of layers, and the network is trained using gradient descent algorithm. Every layer in a CNN has a particular role to play. These roles include, performing convolutions over an image, calculating max or average pooling over a window, and transforming pixel inputs using non-linear activations in form of densely or fully connected neuron layers. Main type of layers are thus: Input Layer, Convolutional Layer, Pooling Layer, and Fully Connected Layer. In more details :

- Input Layer: The pixel values are flattened and fed through the first convolutional layer using the input layer. These pixel values are normalized values, so that a neural network, while back-propagating, can calculate similar gradient for every input feature. Otherwise, the network might over-

compensate or under-compensate for certain features. This makes the model either learn slow or not reach the global maxima in cost.

- Convolutional layer: This will run a square window over pixel values to compute the dot product with weights suitable for detecting various features of an image like vertical or horizontal edge and corners. This results in volume reduction as the window convolve over the whole image producing single value for all pixels which come under the window once. Small Striding and Padding are ways to keep the size of image from shrinking as pixels move through a CNN network.

- Pooling Layer: This layer down-samples the data and thus keeps the network lean, simultaneously preserving the most relevant feature information like the Max pixel value or the Average pixel value in an operation. This layer operates along the dimensions of width and height of the previous layer (a Conv layer).

- Densely Connected Layer: Also known as Fully connected layer, it generally consists of Rectified Linear Units or ReLUs in order to process images faster. This layer is usually fully connected with all the preceding and following layer neurons. There are other activation function choices too, like sigmoid or tanh, but ReLU is the faster and more accurate option. This layer applies activation function elementwise and size of the input remains unchanged after the processing.

- Softmax operation: Last layer of the CNN computes the relative probabilities of classes for every input image. The number of nodes or neurons in this layer is equal to the number of classes in our classification problem, where each nodes value after activating is one of the class score. Also note that, each neuron in this layer is connected to all other neurons in the previous layer, or to say it is fully connected. A general architecture of CNN can be seen in figure 4.1, in page 38.

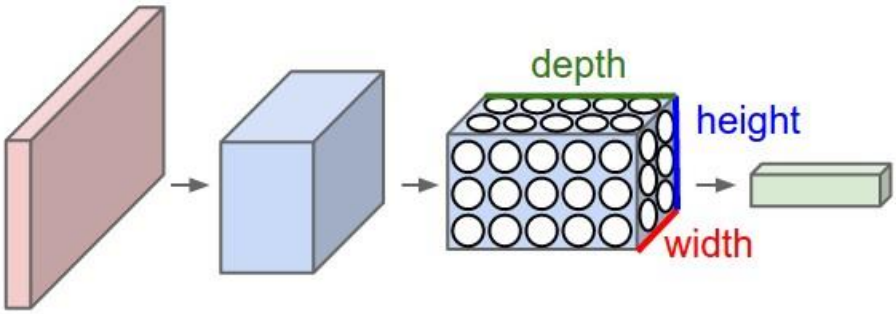

Figure 4.1 CNN architecture. A CNN outputs 3D volume at each step of the process, where width is the number of channels. The size goes on decreasing due to the nature of convolutions and also due to introduced Pooling after every few steps. Image retrieved from https://cs231n.github.io/convolutional-networks/

4.1.2 Resnet50

Residual Network (He, Zhang, Ren & Sun, 2015) was the winner of ILSVRC in 2015. The network architecture includes, skip connections and batch normalization, among other key Convolutional network features like pooling, striding, Relu, SGD *etc.* ResNets are widely used in various applications both at academic as well as industrial levels. They are often referred as Deep Residual Networks too.

ResNet50 has residual or conv blocks repeated or stacked over one another using skip connections, please refer the figure 4.2, in page 39. Also, please refer figure 4.3, in page 40 to see Error rates (in percentages, top 5 and top 1 error rates) on ImageNet validation task of ILSVRC 2015. VGG-16 (Simonyan & Zisserman, 2015), GoogleNet, and PReLU-net (He, Zhang, Ren, & Sun, 2015) are previous state-of-the-art classifiers before the advent of Resnet. ResNet-50/101/152 are networks with respective number of deep layers included in Resnets.

```
Model: "resnet50"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            [(None, 64, 64, 3)]  0
__________________________________________________________________________________________________
conv1_pad (ZeroPadding2D)       (None, 70, 70, 3)    0           input_2[0][0]
__________________________________________________________________________________________________
conv1_conv (Conv2D)             (None, 32, 32, 64)   9472        conv1_pad[0][0]
__________________________________________________________________________________________________
conv1_bn (BatchNormalization)   (None, 32, 32, 64)   256         conv1_conv[0][0]
__________________________________________________________________________________________________
conv1_relu (Activation)         (None, 32, 32, 64)   0           conv1_bn[0][0]
__________________________________________________________________________________________________
pool1_pad (ZeroPadding2D)       (None, 34, 34, 64)   0           conv1_relu[0][0]
__________________________________________________________________________________________________
pool1_pool (MaxPooling2D)       (None, 16, 16, 64)   0           pool1_pad[0][0]
__________________________________________________________________________________________________
conv2_block1_1_conv (Conv2D)    (None, 16, 16, 64)   4160        pool1_pool[0][0]
__________________________________________________________________________________________________
conv2_block1_1_bn (BatchNormali (None, 16, 16, 64)   256         conv2_block1_1_conv[0][0]
__________________________________________________________________________________________________
conv2_block1_1_relu (Activation (None, 16, 16, 64)   0           conv2_block1_1_bn[0][0]
__________________________________________________________________________________________________
conv2_block1_2_conv (Conv2D)    (None, 16, 16, 64)   36928       conv2_block1_1_relu[0][0]
__________________________________________________________________________________________________
conv2_block1_2_bn (BatchNormali (None, 16, 16, 64)   256         conv2_block1_2_conv[0][0]
__________________________________________________________________________________________________
conv2_block1_2_relu (Activation (None, 16, 16, 64)   0           conv2_block1_2_bn[0][0]
__________________________________________________________________________________________________
conv2_block1_0_conv (Conv2D)    (None, 16, 16, 256)  16640       pool1_pool[0][0]
```



```
__________________________________________________________________________________________________
conv5_block2_out (Activation)   (None, 2, 2, 2048)   0           conv5_block2_add[0][0]
__________________________________________________________________________________________________
conv5_block3_1_conv (Conv2D)    (None, 2, 2, 512)    1049088     conv5_block2_out[0][0]
__________________________________________________________________________________________________
conv5_block3_1_bn (BatchNormali (None, 2, 2, 512)    2048        conv5_block3_1_conv[0][0]
__________________________________________________________________________________________________
conv5_block3_1_relu (Activation (None, 2, 2, 512)    0           conv5_block3_1_bn[0][0]
__________________________________________________________________________________________________
conv5_block3_2_conv (Conv2D)    (None, 2, 2, 512)    2359808     conv5_block3_1_relu[0][0]
__________________________________________________________________________________________________
conv5_block3_2_bn (BatchNormali (None, 2, 2, 512)    2048        conv5_block3_2_conv[0][0]
__________________________________________________________________________________________________
conv5_block3_2_relu (Activation (None, 2, 2, 512)    0           conv5_block3_2_bn[0][0]
__________________________________________________________________________________________________
conv5_block3_3_conv (Conv2D)    (None, 2, 2, 2048)   1050624     conv5_block3_2_relu[0][0]
__________________________________________________________________________________________________
conv5_block3_3_bn (BatchNormali (None, 2, 2, 2048)   8192        conv5_block3_3_conv[0][0]
__________________________________________________________________________________________________
conv5_block3_add (Add)          (None, 2, 2, 2048)   0           conv5_block2_out[0][0]
                                                                 conv5_block3_3_bn[0][0]
__________________________________________________________________________________________________
conv5_block3_out (Activation)   (None, 2, 2, 2048)   0           conv5_block3_add[0][0]
__________________________________________________________________________________________________
avg_pool (GlobalAveragePooling2 (None, 2048)         0           conv5_block3_out[0][0]
==================================================================================================
Total params: 23,587,712
Trainable params: 23,534,592
Non-trainable params: 53,120
__________________________________________________________________________________________________
```

Figure 4.2 A Small Portion of Renet50 layers, shown as an output of summary() operation in Tensorflow Keras implementation

model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

Figure 4.3 RESNET Performance on ImageNet task in comparison to other state-of-the-art networks (He, Zhang, Ren & Sun, 2015)

4.1.3 Model A: RGB based model

Architecture: This model is using Resnet50 as the *base-model* and adding sequential TensorFlow Keras layers over it. Resnet50 means it is a residual network which is fifty layers deep, the choice of fifty layers is made for the ease of running the model over large training and validation datasets, while maintaining the effectiveness of residual architecture of using large network depth for learning more intricate and subtle features which a smaller network might not be able to learn.

Purposely, Resnet50 is used without its weights, meaning it is used with random weight initialization rather than the ImageNet competition learned weights, and the layers are *kept as trainable*. Also, the top layer or the classification layer with previous thousand nodes is replaced with a fully connected layer with number of nodes equal to the number of new classes, which is equal to hundred and ten in this study (hundred ImageNet classes and ten Eurosat classes). The idea here is to learn from scratch on the new dataset of mini-ImageNet and EuroSat combined, once on RGB and then on Greyscale colour spaces.

On top of trainable Resnet50 base-model, one fully connected ReLU layer is added with two hundred and fifty-six activation nodes in it. This Dense or fully connected

layer is followed and preceded by a Dropout layer with dropout nodes as 0.5, meaning randomly chosen half of the nodes, in the immediately preceding hidden layer, will not be used during the training process of the network.

4.1.4 Model B: Greyscale based

The Architecture for Greyscale model is exactly similar to above *RGB model's* (Model A) architecture except the fact that single channel goes in as input to the three-channel input of the base model Resnet50. Please note that, Resnet50 and other ImageNet based architectures are designed for three channels as inputs, i.e. one channel each for Red, Green, and Blue channel of an incoming coloured image. So, in case of single channel images like greyscale ones, the same pixel values are fed as input to three input streams of the Resnet50 network.

4.1.5 Models for Bands B02, B03, B04, B05, B08, and B12

Model A and Model B are used to transfer learned features to Multispectral feature space separately. A new network architecture is designed and is used as a layer on top of the base models A and B for this purpose. This network layer consists of one fully connected dense layer with Sigmoid activation function, followed by one Keras Batch Normalisation layer to normalize the inputs from this layer to the final output Softmax layer. A Dropout layer is also added to this network to reduce the overfitting of the model.

Due to the large size of overall model – Resnet50 followed by Relu Activation and Dropout layers in the Base model, then new network with one dense activation layer, one Batch normalisation layer, one Dropout layer and lastly one fully connected Softmax output layer, the overall model Overfitted the small target dataset of 27,000 Eurosat Multispectral images. Please refer table 4.1, in page 42, to further understand the number of instances for every landcover class and for training, validation and test sets. This problem of overfitting was solved using a lengthy and iterative trial and error approach. During this analysis, I was found that by changing activation function of first dense layer from ReLU to Sigmoid, the Validation Loss of the network was reduced by manifolds. After this the one of the key Compiling parameters of Optimizer

was toggled between ADAM, Stochastic Gradient Descent, and RMSprop for different number of epochs. Similarly Learning rate was varied between (10E-4) to (10E-7) and it was found lower learning rate was giving them a better fit. Further, RMSprop with learning rate of 0.001 and 0 decay, gave the best results so far of 56% Test set accuracy.

Table 4.1 Class wise training, validation, and test data-instances count

	Number of Images		
EuroSat Classes	**Training**	**Validation**	**Test**
Annual Crop	2025	225	750
Forest	2025	225	750
Herbaceous Vegetation	2025	225	750
Highway	1688	187	625
Industrial	1688	187	625
Pasture	1350	150	500
Permanent Crop	1688	187	625
Residential	2025	225	750
River	1688	187	625
Sea Lake	2025	225	750
Total	**18227**	**2023**	**6750**

With a large network architecture and small target dataset the problem of overfitting was still not solved. Further analysis proved that by using a Keras Callback function called Reduce Learning Rate on Plateau or ReduceLROnPlateau, with Patience value of two overfitting was further reduced and test accuracy was increased. This Callback option decreases the learning rate as soon as the model stops progressing or the performance plateaus. Using this functionality coupled with low learning rates, raised the test accuracy to 59%.

Lastly, it was noticed that both the training accuracy and validation accuracy were stagnating at values around 59% for long without showing any significant

improvement or degradation even after waiting for many epochs. This happens when the loss gets stuck in areas with high eigenvalue regions. In terms of gradient traversing, this would mean getting stuck into narrow valleys which are long and any movement, no matter for how long it is doesn't change the overall cost function. This happens mainly when there are correlation issues and satellite images are often prone to almost similar type of data between different pixels in same images as well as between two images. To deal with this issue, a Batch normalization layer was introduced in the new network and starting learning rate was kept as 0.0001. This tweak coupled with rescaling the images in train, test and validation, the best Test-Accuracy was observed as 60.38%.

The premise of Transfer Learning is that our pre-trained networks (Model A: RGB and Model B: Greyscale) contain rich set of descriptors or filters. To use the concept of Transfer Learning effectively, features learned from previous task of training over mini-Imagenet and EuroSat are transferred to the target task of Multispectral nature in some series of steps. This is achieved by using "Fine-Tuning" techniques, in which filters are reused by training the network in parts. The network's architecture can be understood easily from the figure 4.4. The steps that this research has followed are as follows –

Figure 4.4 Final network architecture. Resnet50 is used as a building block for creating the Base Model. This model has been trained on RGB and Greyscale images separately. Later on, this Base Model is used for transferring features learned to the target task.

- Step 1 – Train only the *head* of the network or the new network layers that have been added to the base-model and keep the rest of the layers as frozen or

non-trainable. In the figure 4.5, the section marked as (1) is the new network added on top of the *Base Model.* Thus, section marked as (2) and (3) are kept as non-trainable. The fully connected layers in section (1) are initialised with random weights and trained over the EuroSat single band images extracted from the multispectral .tif images.

Reasoning – This way only a part of the network is being trained at first and the weights correction is not back propagated into the entire network. The new layers are initialised with random weights and hence, if the whole network is allowed to train from scratch on the target data, there is a risk of losing the features and filters learned by the fully trained base model. This way training data is propagated in forward direction across the entire network, while backpropagation happens only for final layers are set as trainable (section (1) in figure 4.5). This training is done only for a few epochs (number of epochs = 5), so that the final layers can learn requisite number of features or patterns on the target data. The learning rate for this training can be kept as the default values that comes with the implementation in Keras or other packages, and is usually around 0.001 for RMSprop, or ADAM or SGD alike.

Figure 4.5 Different steps in Fine-Tuning and Feature Transfer

- Step 2 – Train only the non-convolutional layers in the Base Model, i.e. the section (1) and section (2) layers as shown in the figure 4.5. At this point, no weight update will happen for the Resnet50 model and only the top layers are getting trained using forward and backward propagation. In this way, using step 1 and step 2, the network is being *warmed up* for the task at hand.

- Step 3 – unfreezing the final residual or convolutional block in the Resnet50 model, or the terminal block in the section (3) of figure 4.5, in page 44. Also, figure 4.6 below, further elaborates the point by highlighting the portion in a block marked on extreme right side. This time the network is fine-tuned or trained over both, the final residual block, and the final non-convolutional fully connected layers.

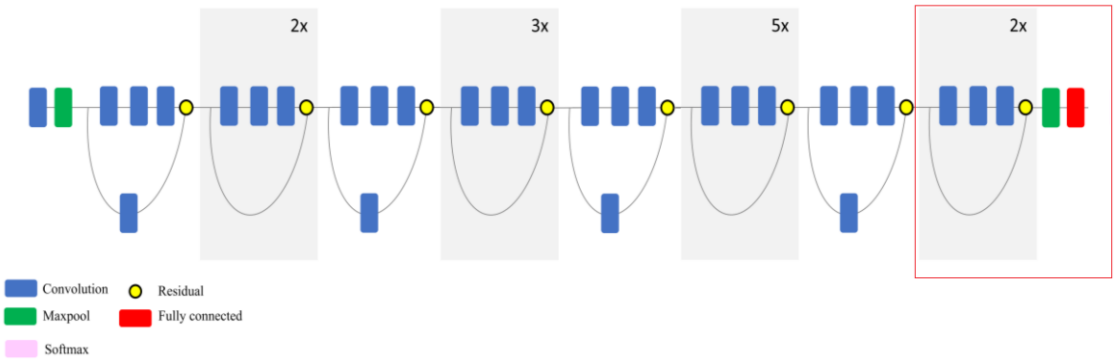

Figure 4.6 In Resnet50 model or the Section (3) of model shown in figure 4.5, last Resnet convolutional block, which is highlighted in the box, was made trainable and rest was left as non-trainable. Image from Mahdianpari et. al., 2018.

4.2 Results

This research is done to determine if networks trained on single channel images are better at transferring features to multispectral domains like satellite and medical imaging, then those trained over RGB images. The performance in land-cover classification task is taken as a measure to answer this research question. Similar steps are followed for training on all the shortlisted six bands out of the thirteen bands that are there in the multispectral images. Throughout the evaluation, the F1 score, Precision, Recall, and average Accuracy over whole dataset has been used as criteria to measure the performance of the models.

The original dataset of 27,000 remote sensing images is split randomly into 75 % Training and remaining 25% as Test rows. Out of these 75% Training, 10% is used as

Validation set. This results in 18,227 images in Training set, 2,023 images in Validation set, and lastly 6,750 images in Testing data set. So, for every band there are 27,000 images that are split into training, validation, and test sets as per above distribution percentages (Table 4.1, in page 42)

4.2.1 Model A: RGB based

Two models, with exact same architecture, have been created over the coloured image database of ImageNet and EuroSat combined. The first model is trained on non-augmented images, thus the number of training images per class is 400 for both ImageNet and Eurosat classes, while for validation it is 100 images per class for ImageNet and 1000 images per class for Eurosat (table 4.2).

Table 4.2 Class instances for small-sized Non-augmented RGB image dataset

Non-Augmented RGB	**Number of Images Per Class**	
Classes	**Training**	**Validation**
100 ImageNet Classes	400	100
10 EuroSat Classes	400	1000
Total	**44000**	**20000**

The second model is trained on the augmented and larger database of coloured images from ImageNet and Eurosat alike. The training set has 2000 images for each class, while the validation has a similar configuration to before. No augmentation is done in Validation set in order to keep the set as closely reflecting the real-world scenario as possible (table 4.3, in page 47). The architecture of both these models is the same and can be understand by looking at the figure 4.5, in page 44, where the sections marked as (2) and (3) represent them as *Base model*.

Table 4.3 Class instances for large-sized Augmented RGB image dataset

Augmented RGB	**Number of Images Per Class**	
Classes	**Training**	**Validation**
100 ImageNet Classes	2000	100
10 EuroSat Classes	2000	1000
Total	**220,000**	**20000**

For both the models, a high training accuracy, in late 90%s, has been achieved by training just over 100 epochs. However, the validation scores do not seem to be coming up correctly at the moment, but this data will be corrected and included during the final presentations. Due to the remote nature of dissertation owing to the pandemic, a dearth of local GPU clusters, limited time for dissertation efforts, and a huge size of training and validation datasets, the processing a was very challenging task. For example, the total size of training images for creating augmented models was more than 400,000 RGB and Greyscale images and total number of classes was 110 with ImageNet and Eurosat combined. Currently, validation accuracy is not that high (around 48%).

4.2.2 Model B: Greyscale based

Two models are similarly created, with exact similar architecture and parameters as in Model A mentioned in above section. One model was created on non-augmented and smaller dataset, while the other one was created on augmented and larger datasets with same number of images in both training and validation sets as used in previous section for two RGB based models. High training accuracies have been observed in similar number of epochs for both the models, while validation accuracies and losses are not that optimal at the time of writing this report. However, this will be remedied for both the models, in the final presentation. Tables 4.4 and 4.5, both in page 48, depicts the class distribution and total samples in the greyscale training and validation sets.

Table 4.4 Class instances for small-sized Non-augmented Greyscale image dataset

Non-Augmented Greyscale	**Number of Images Per Class**	
Classes	**Training**	**Validation**
100 ImageNet Classes	400	100
10 EuroSat Classes	400	1000
Total	**44000**	**20000**

Table 4.5 Class instances for large-sized Augmented Greyscale image dataset

Augmented Greyscale	**Number of Images Per Class**	
Classes	**Training**	**Validation**
100 ImageNet Classes	2000	100
10 EuroSat Classes	2000	1000
Total	**220,000**	**20000**

4.2.3 Model Band B02, B03, B04, B05, B08, and B12

Model architecture for all the bands is same and it is already discussed in section 4.1.5 in great details. These six bands are chosen out of given thirteen bands because of mainly to reasons – they were the top performers in the original paper (Helber, Bischke, Dengel, & Borth. 2017), some of the bands like B01, B09, B10 and B11 are not even meant for land observation altogether. Band01 is for detecting Aerosols in the air, Band09 is for detection of Water Vapours suspended in the atmosphere, Band10 is meant for treating Cirrus clouds (low thin clouds near earth's surface), and lastly Band11 is meant for cloud/ice/snow discrimination. Thus, the Bands that will be evaluated in this study are –

Band02 – Blue Color
Band03 – Green Color

Band04 – Red Color

Band05 – Red Edge 1

Band08 – Near Infrared

Band12 – Shortwave Infrared 2

Table 4.6 below, shows overall accuracy values over test sets for these six bands, and Aug stands for models created over larger augmented datasets. It is evident from the table that for all bands the highest accuracy was recorded when a Greyscale Augmented base model was used. Also, note that for five out of six bands, bands B02, B03, B04, B08 and B12, the Greyscale base model has trumped the RGB base model. Also, note that, for band B05 the accuracy for model with RGB as base model is only marginally better than the accuracy Greyscale base model.

Table 4.6 Accuracy values over the dataset for different bands as measured for every Base-Model types used

Bands	**Test-Set Accuracy Values for each type of Base-Model**			
	RGB	**GREY**	**RGB-Aug**	**GREY-Aug**
Band B02 – Blue	57.7	60.67	59.87	66.3
Band B03 – Green	54.82	56.19	61.27	65.83
Band B04 – Red	55.02	59.62	59.71	62.6
Band B05 – Red Edge 1	43.44	43.32	45.44	52.78
Band B08 – NIR	55.82	58.84	59.64	64.05
Band B12 – SWIR 2	43.11	45.15	47.29	54.35

The grouped bar charts, shown in figure 4.7, in page 50, clearly depicts the behaviour for all bands over different base models. On an average, the performance was worse when RGB based images were used to train the base model, and it was best when Greyscale images were used for training of the base model. Also, note that augmentation helped in increasing the performance in both cases – RGB color space and Greyscale space.

The authors of database contributing paper have observed that best performance was given by spectrums of Red (B04), Green (B03) and Blue (B02) bands. This is quiet similar to what is observed in this research. Interestingly, the band NIR or Near-Infrared (B08) has outperformed even the Red (B04) band, and the bands Red Edge 1 (B05) and Shortwave-Infrared 2 (B12) are worse performers.

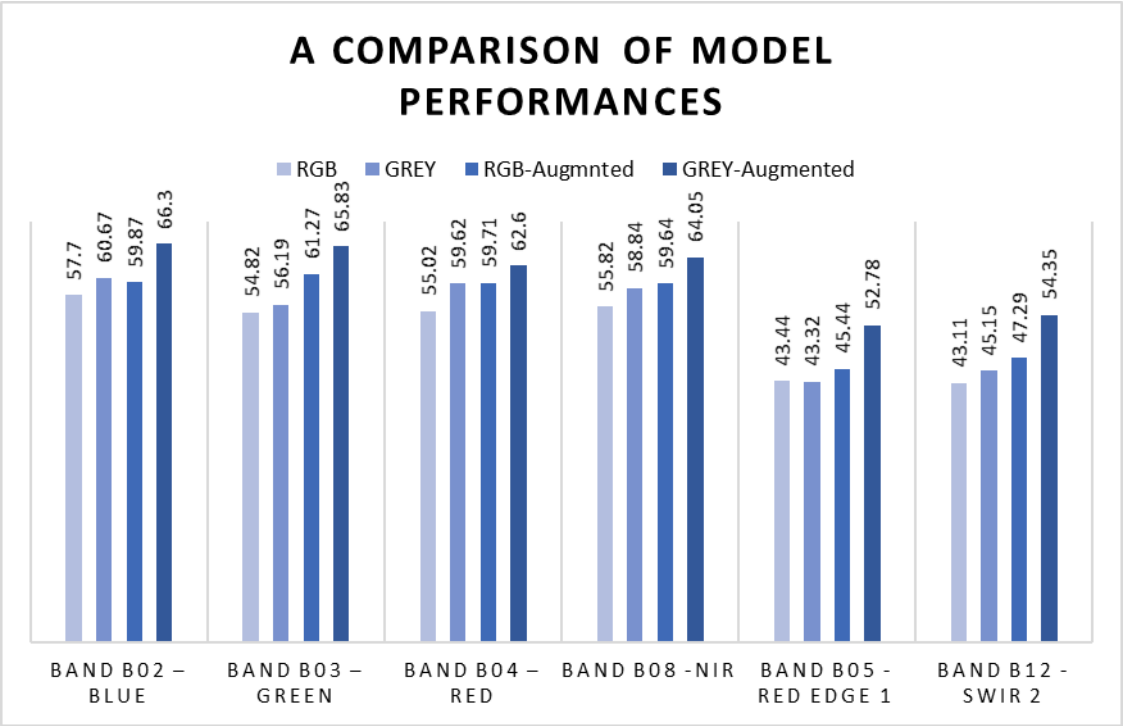

Figure 4.7 Grouped bar charts depicting the performance of different bands as well as different base models among them. Clearly Greyscale Augmented base model has outperformed in every group.

4.2.3.1 Model Band B02

As per table 4.6, in page 49, highest overall accuracy has been recorded by the model which is using base model trained on augmented greyscale images.

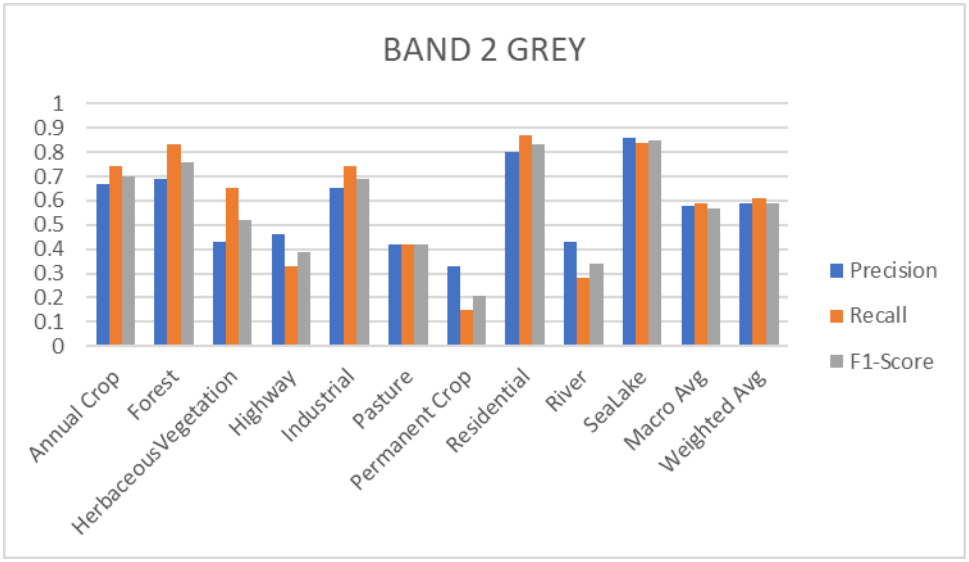

Figure 4.8 B02: Comparative performances for Greyscale images trained base model

For all the Models on band B02, classes Sea Lake, Residential, and Forest have given highest F1 scores. While Highway, River and Permanent Crop are the lowest performing classes. This (Figure 4.8, 4.9, 4.10, 4.11). Interestingly, it can be seen that class Sea Lake has high Precision, Recall and thus high F1 score. This can be explained by entirely different Reflectance of a water body from another typical *land bodies*.

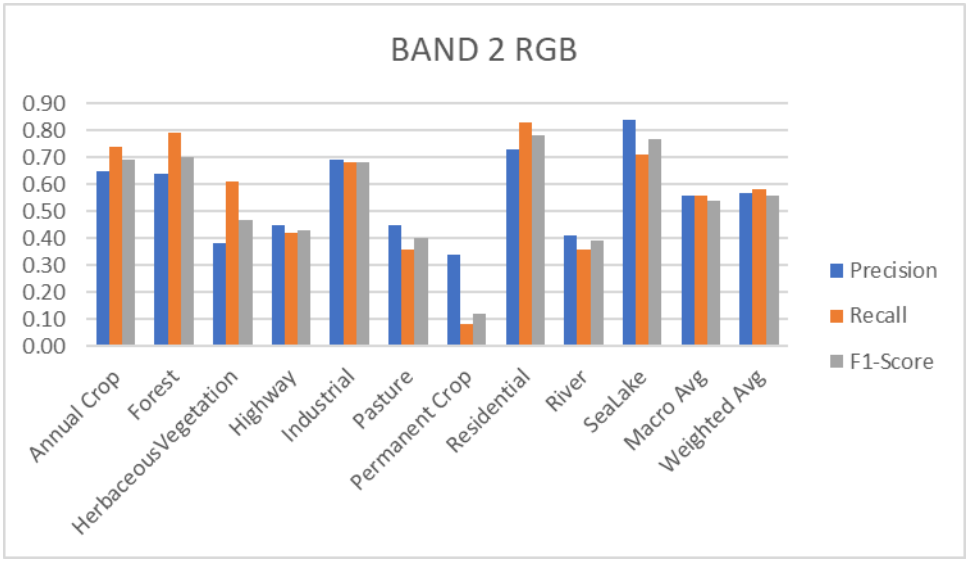

Figure 4.9 B02: Comparative performances for RGB images trained base model

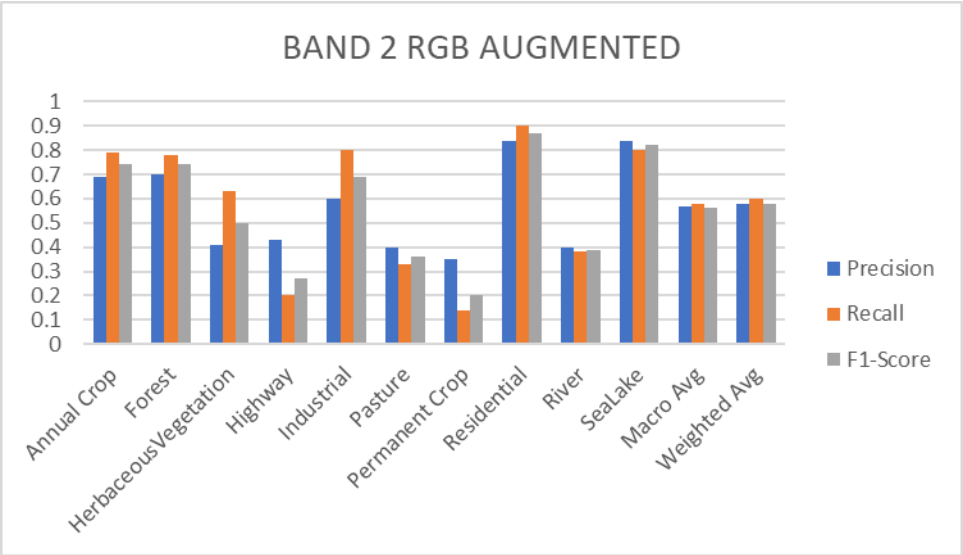

Figure 4.10 B02: Comparative performances for Augmented RGB images trained base model

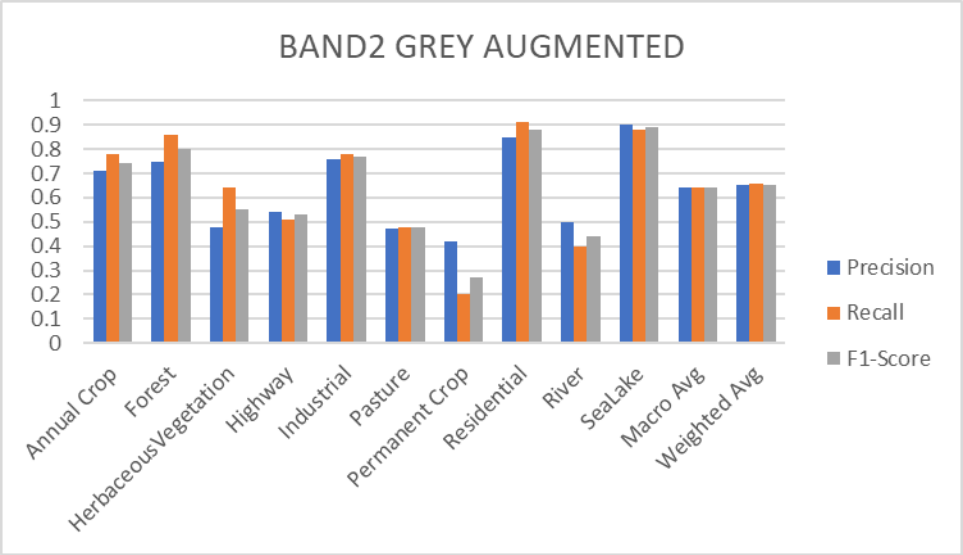

Figure 4.11 B02: Comparative performances for Augmented Greyscale images trained base model

4.2.3.2 Model Band B03

The highest F1 score is shown by Sea Lake class across all the models, and the Highway class has the lowest F1 scores across all the models.

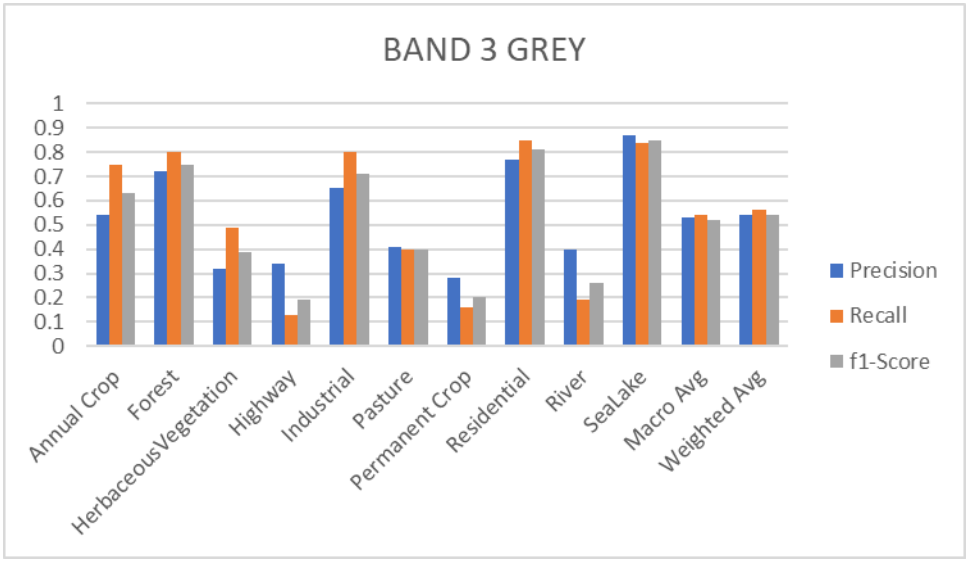

Figure 4.12 B03: Comparative performances for Greyscale images trained base model

Likewise, when base model was used as augmented greyscale one, the F1 scores rose for all the classes (figure 4.12, 4.13, 4.14, and 4.15). Also, it can be seen clearly that, greyscale augmented has trumped all other models in feature transfer.

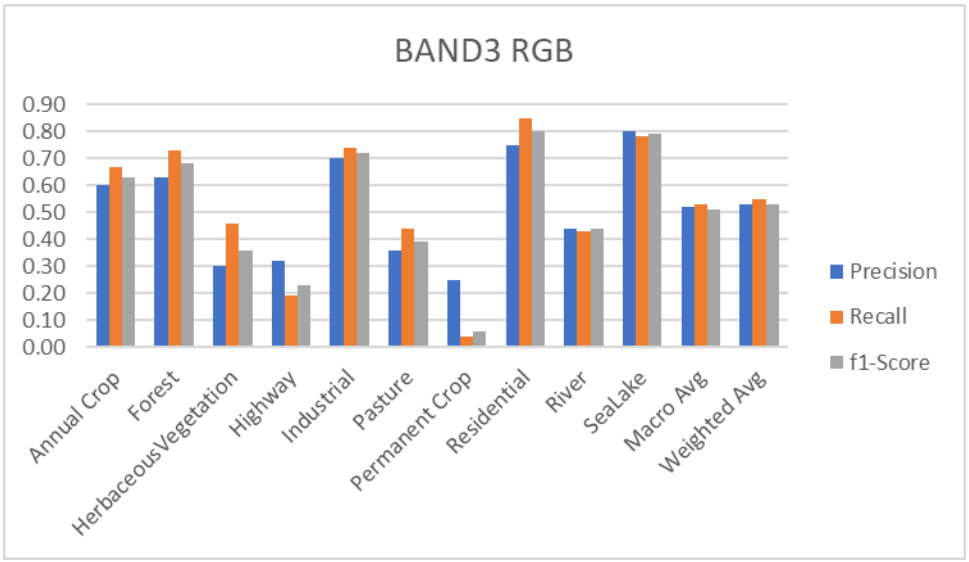

Figure 4.13 B03: Comparative performances for RGB images trained base model

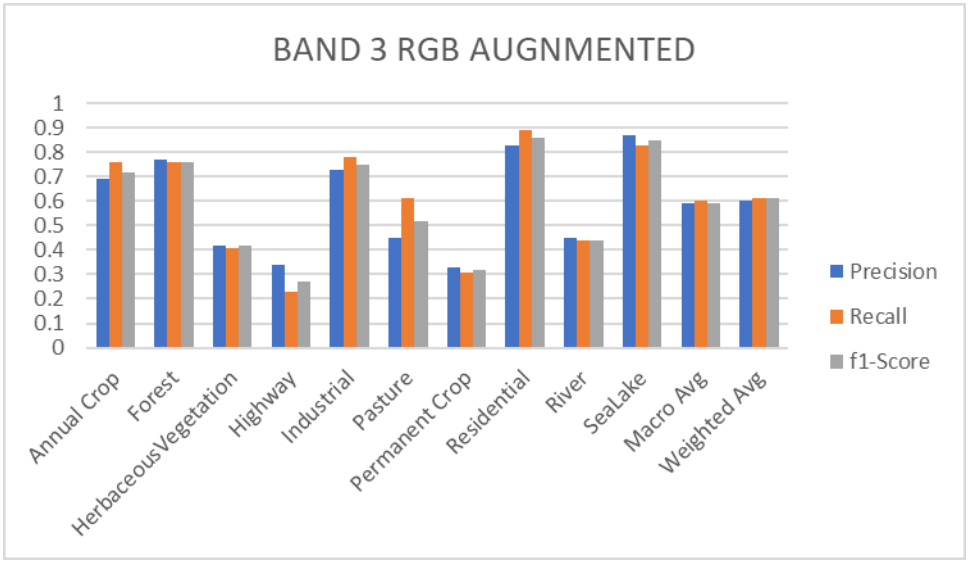

Figure 4.14 B03: Comparative performances for Augmented RGB images trained base model

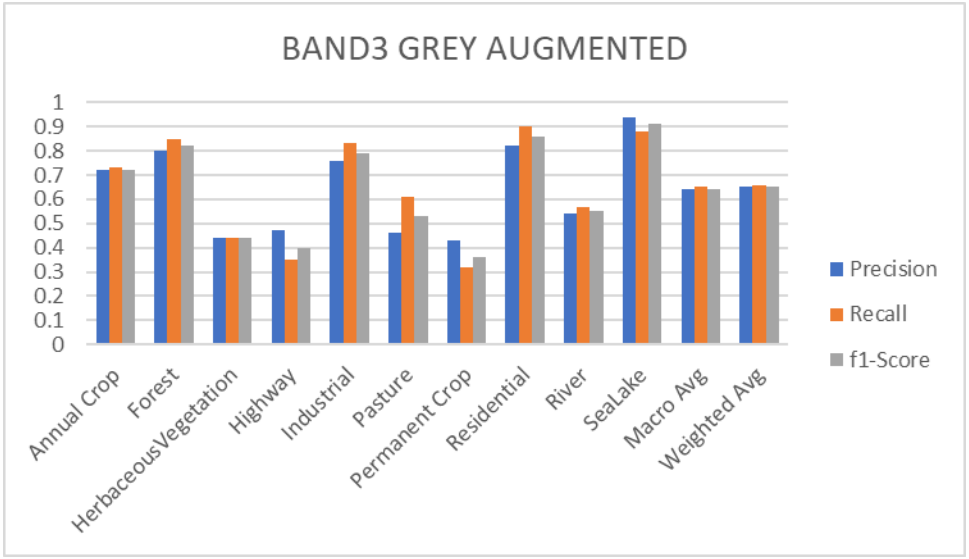

Figure 4.15 B03: Comparative performances for Augmented Greyscale images trained base model

4.2.3.3 Model Band B04

Band B04, is the Red spectrum of visible light. It has similar performance as the previous visible bands (B02 and B03). As per figures 4.16, 4.17, 4.18, and 4.19, it I evident that here too base model trained on augmented greyscale images has given best F1 Scores for all classes among the four model configurations.

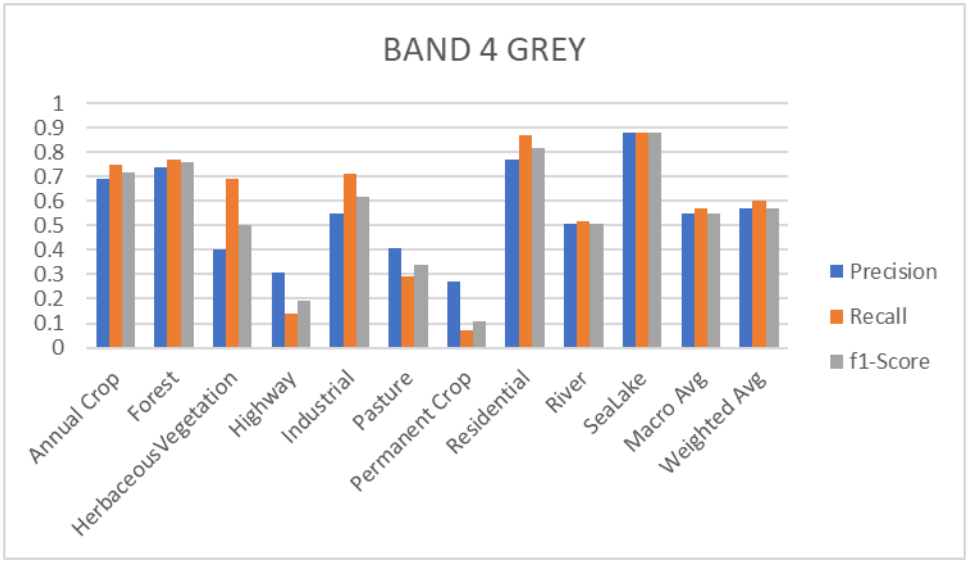

Figure 4.16 B04: Comparative performances for Greyscale images trained base model

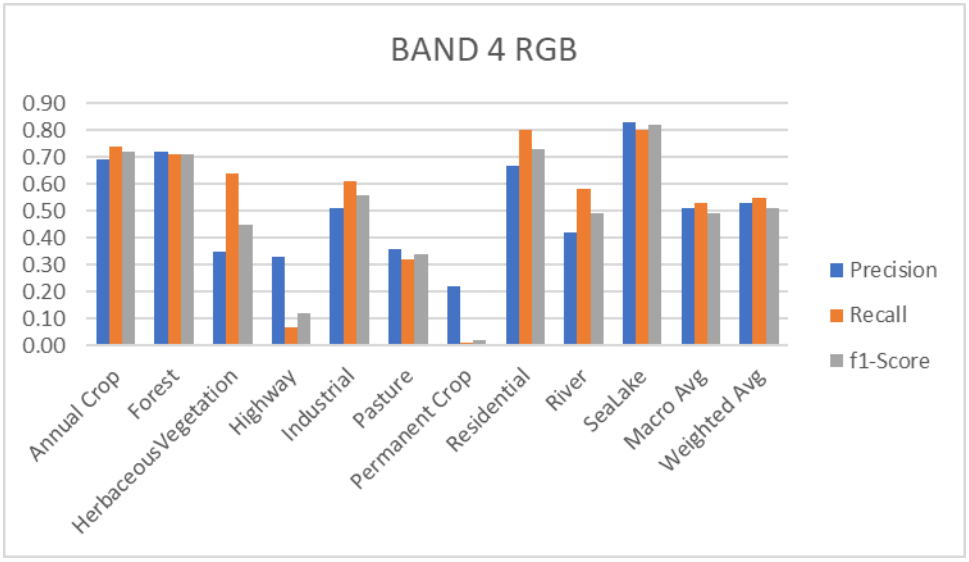

Figure 4.17 B04: Comparative performances for RGB images trained base model

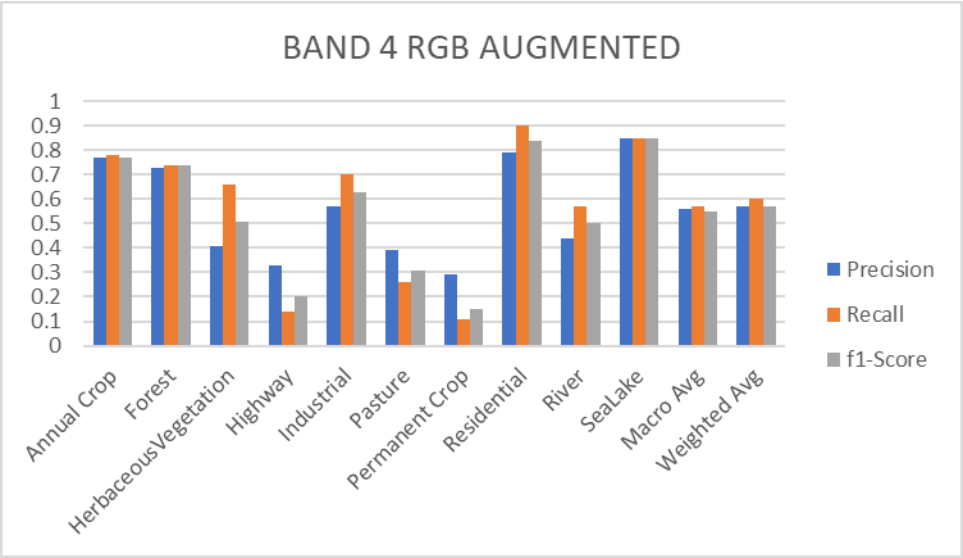

Figure 4.18 B04: Comparative performances for Augmented RGB images trained base model

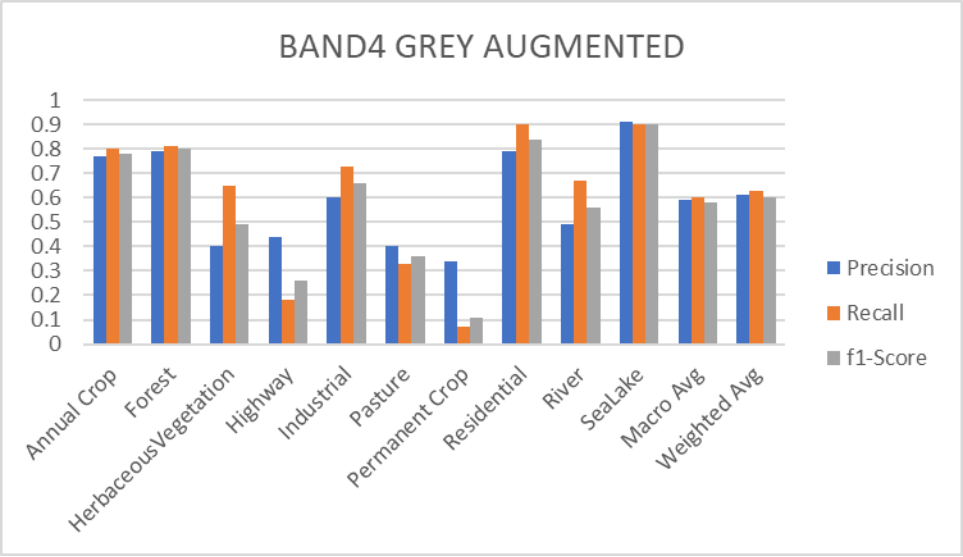

Figure 4.19 B04: Comparative performances for Augmented Greyscale images trained base model

4.2.3.4 Model Band B05

Things got interesting with Band B05 or Red Edge 1, as River is among the best performing classes for this band unlike other seen till now. Also, Forest class had high F1 scores for all bands till now, but for band B05, it is one the worse performing.

Residential and Sea Lake are high performers in this case too (figure 4.20, 4.21, 4.22 and 4.23).

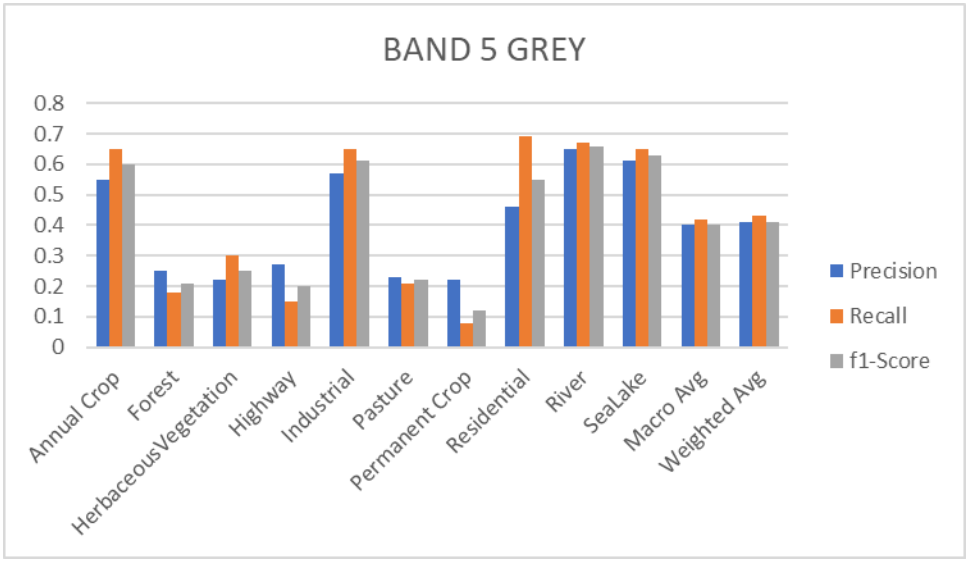

Figure 4.20 B05: Comparative performances for Greyscale images trained base model

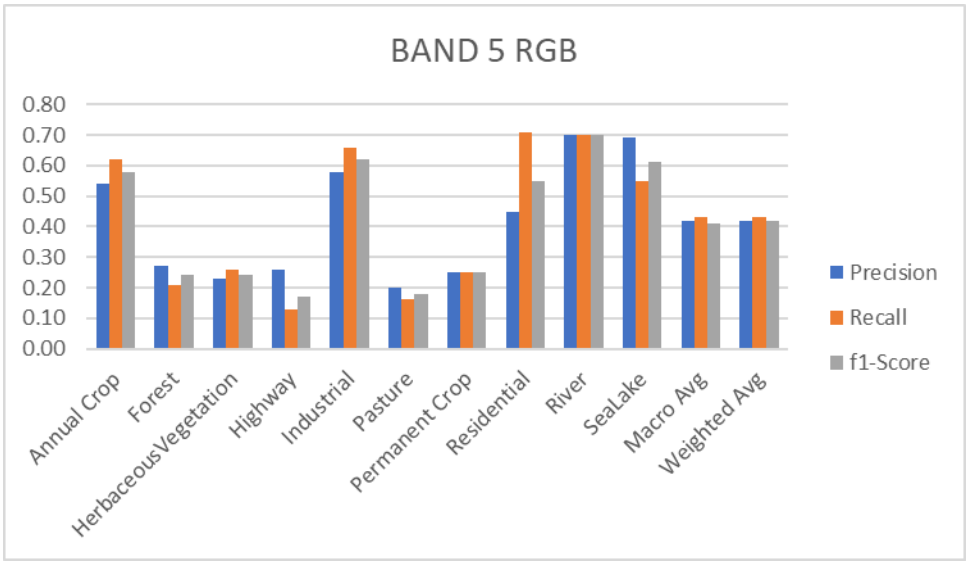

Figure 4.21 B05: Comparative performances for RGB images trained base model

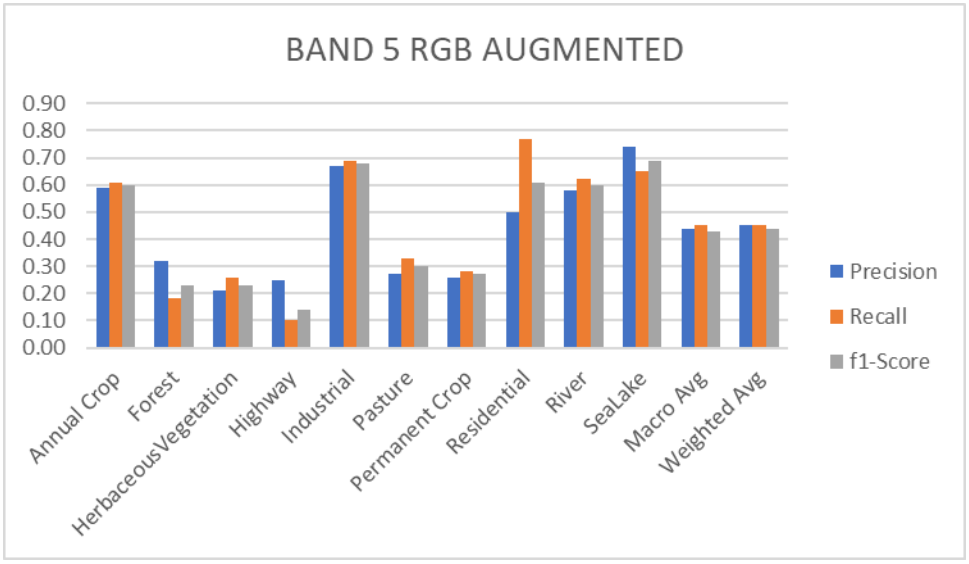

Figure 4.22 B05: Comparative performances for Augmented RGB images trained base model

As per table 4.9, in page 49, this band has performed worse among the six bands, in terms of overall accuracies. However, here too the best performing model is the one trained on augmented greyscale images. This implies that using greyscale along with augmentations can significantly increase the performance of a model on multispectral bands.

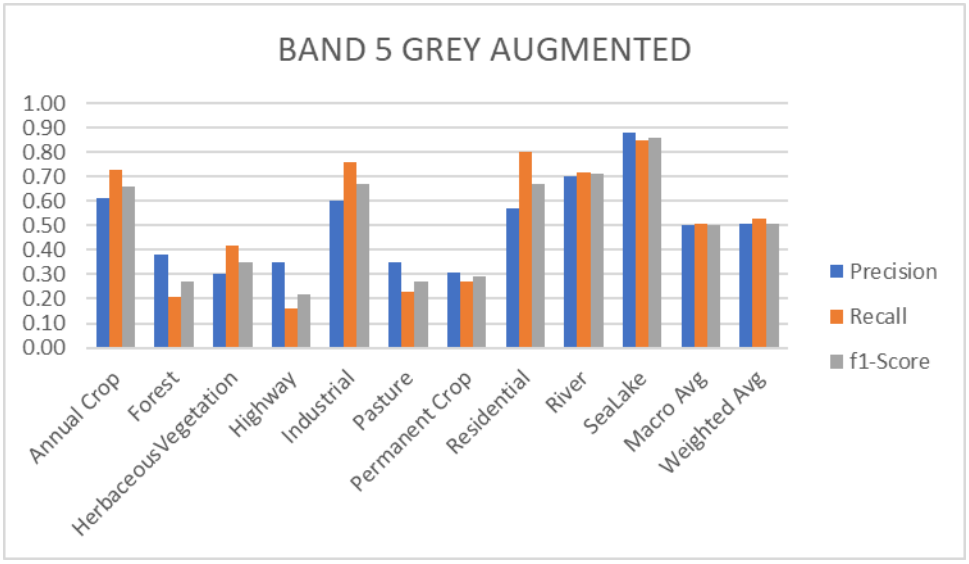

Figure 4.23 B05: Comparative performances for Augmented Greyscale images trained base model

4.2.3.5 Model Band B08

Similar to Band B05, Band B08 or Near Infrared (NIR) has River class as one of the high F1 scorers. Other than this, the scores are similar to visible bands, with Sea Lake as best performer and Permanent Crop as worst.

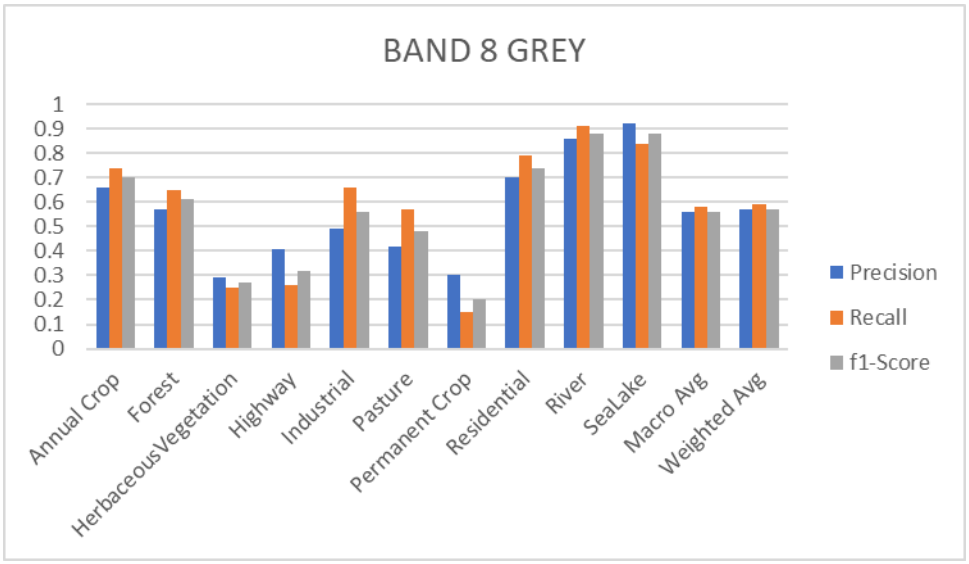

Figure 4.24 B08: Comparative performances for Greyscale images trained base model

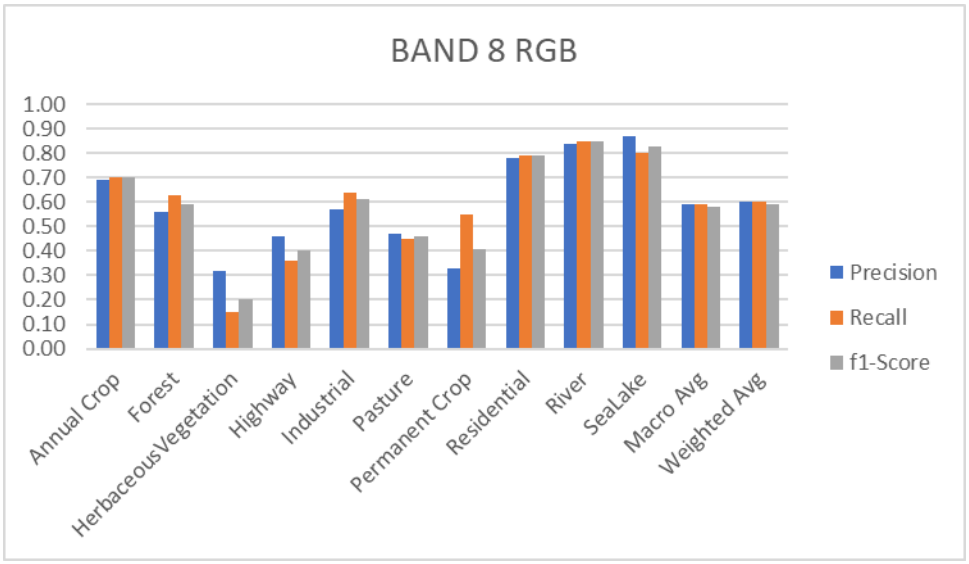

Figure 4.25 B08: Comparative performances for RGB images trained base model

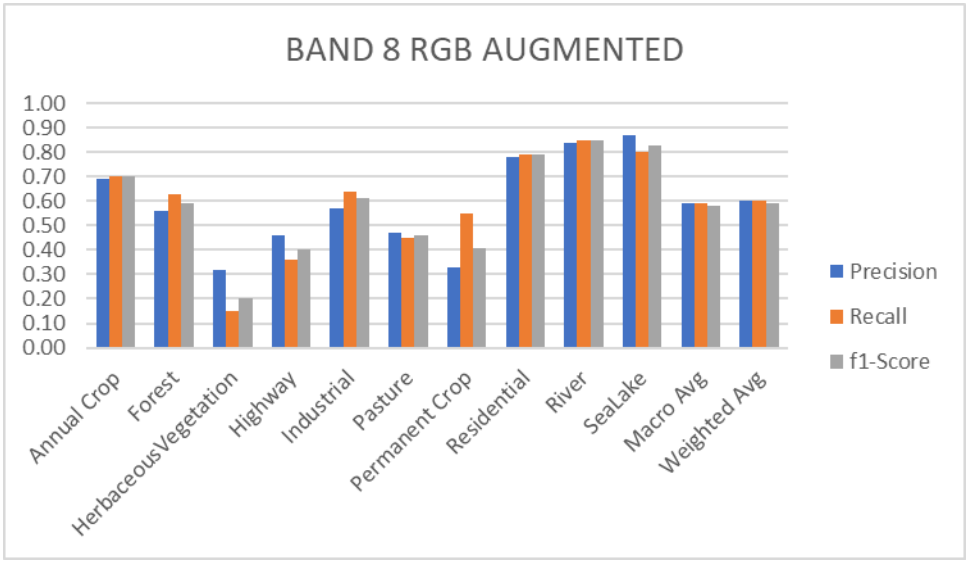

Figure 4.26 B08: Comparative performances for Augmented RGB images trained base model

Using augmented greyscale images for training base model has shown best performance, as per table 4.6, in page 49, and also from figures 4.24, 4.25, 4.226 and 4.27. notedly, Band B08 has scored even higher than visible band B04 or Red.

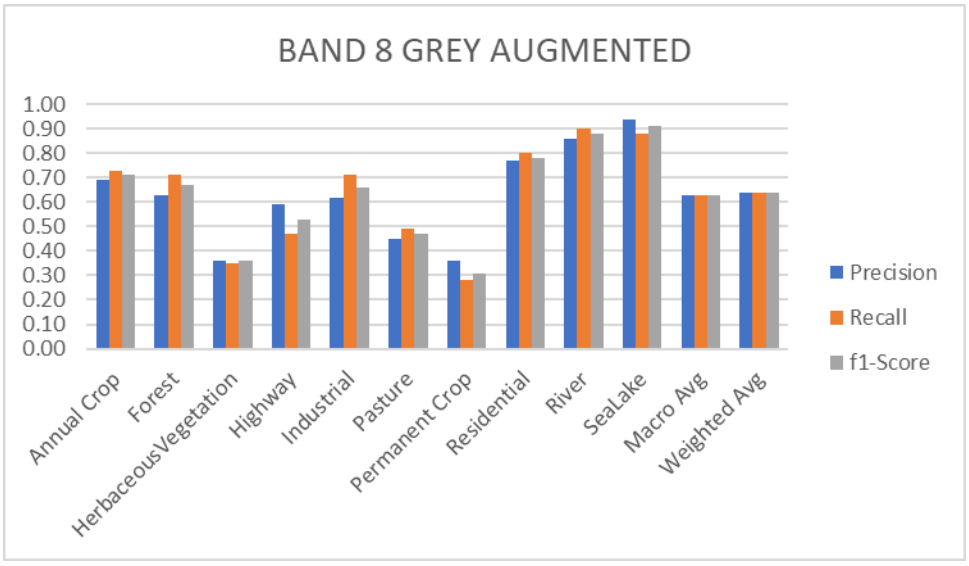

Figure 4.27 B08: Comparative performances for Augmented Greyscale images trained base model

4.2.3.6 Model Band B12

Permanent crop has F1 score, Precision, and Recall all as 0 values, meaning, not even once True Positive or True Negative has been identified by the model. This behaviour is consistent for RGB, RGB-Augmented and Greyscale models, only Augmented Greyscale images model was able to classify permanent crop class (figure 4.28, 4.29, 4.30, and 4.31).

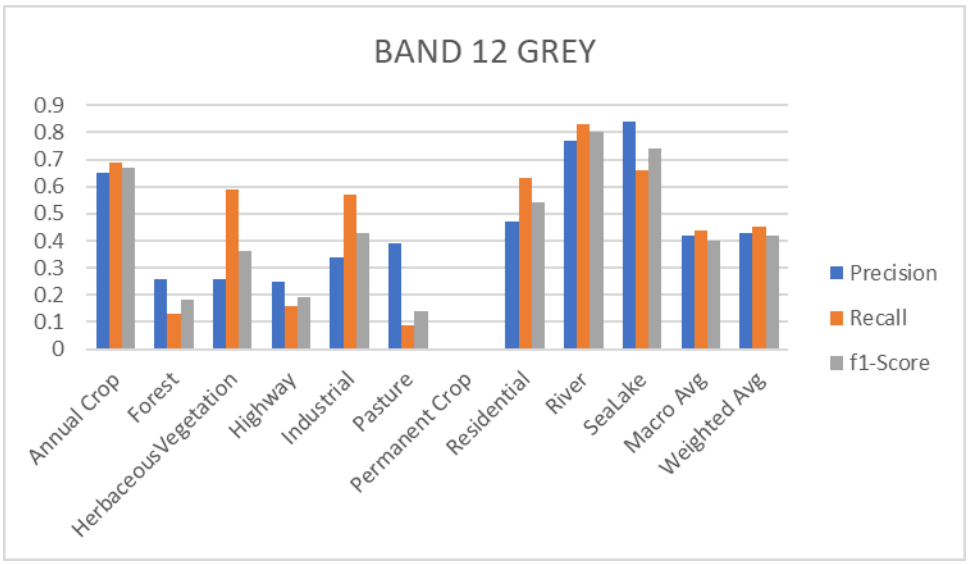

Figure 4.28 B12: Comparative performances for Greyscale images trained base model

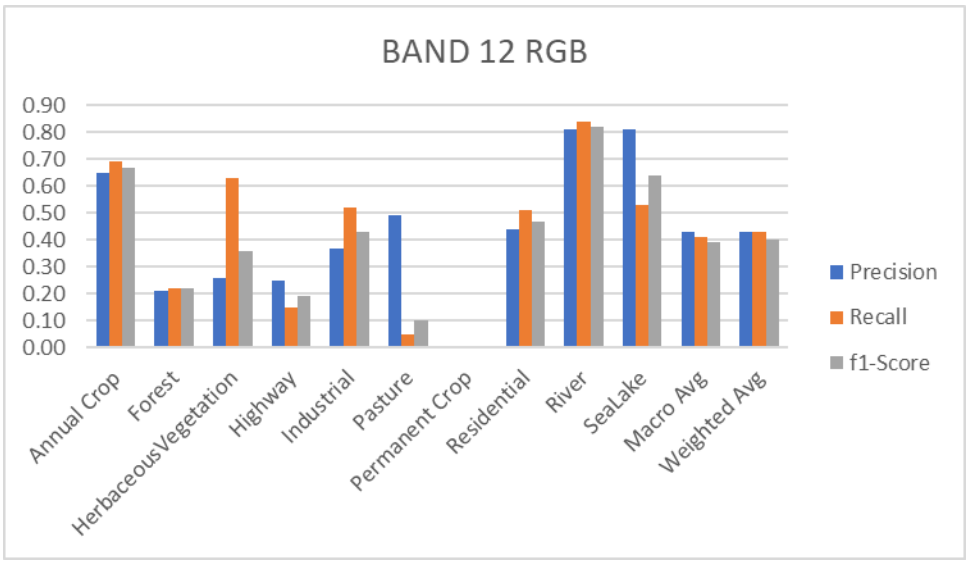

Figure 4.29 B12: Comparative performances for RGB images trained base model

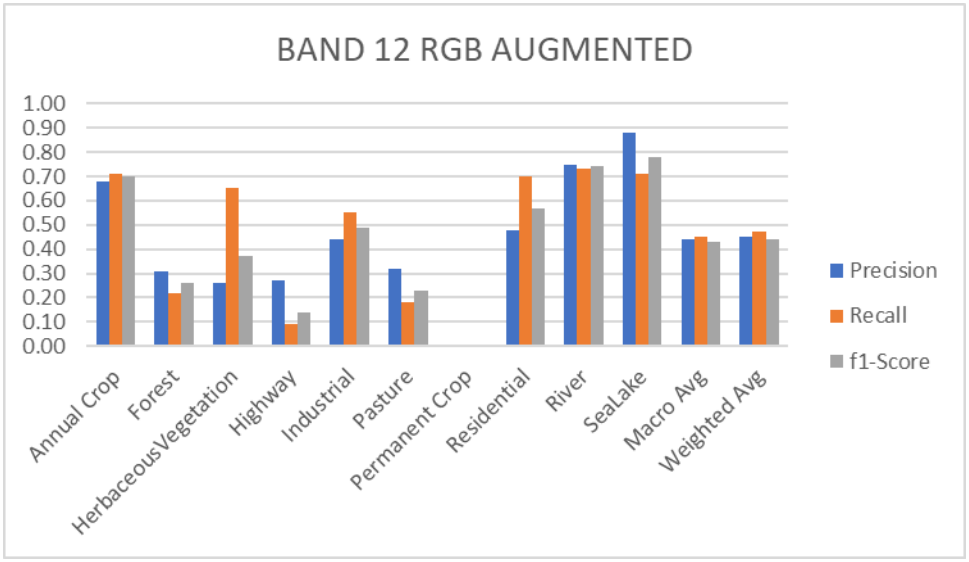

Figure 4.30 B12: Comparative performances for Augmented RGB images trained base model

Base model built using augmented greyscale images has shown most robust performance for band B12 or Shortwave Infrared 2 as well.

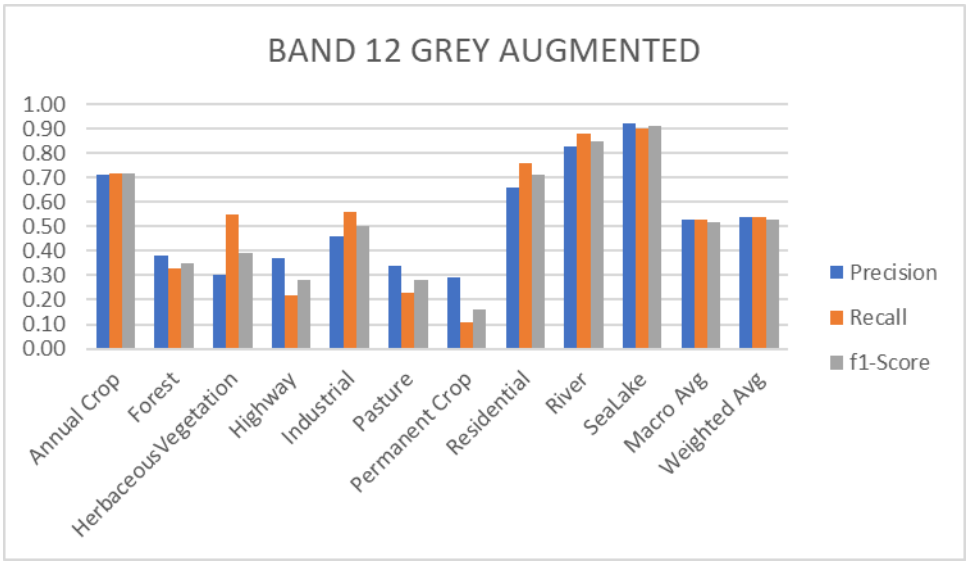

Figure 4.31 B12: Comparative performances for Augmented Greyscale images trained base model

4.3 Summary

This chapter elaborates upon the model architecture, the data distribution among classes and the readings from various experiments. Resnet50 has been used to build the base models in this research and it has been fine-tuned by unfreezing the last residual convolutional blocks. Non-convolutional layers of base model are retrained by running on the target task for small number of epochs. This was done to ensure that patterns learned from the previous or source task are not completely unlearned. Furthermore, the new layers that were added on top of base model were trained from scratch first for small number of epochs and later on for more number of epochs to completely learn the features from base model as well as from the target data.

The bands chosen for the analysis in this study are the top performers in the original paper and the results from this study and original paper look consistent. Original paper used Resnet50 trained on the Imagenet ILSVRC task, and thus has features only from coloured space. This study however analyses use of single channel source instead of RGB and results seems very exciting.

5 EVALUATION AND DISCUSSION

5.1 Introduction

Evaluating on only six of the 13 Sentinel 2A multispectral bands, because these are the top performers among all 13 bands. Band01, Band 09, Band10, and Band11 are not meant for landcover observation. Band01 is for detecting Aerosols in the air, Band09 is for detection of Water Vapours suspended in the atmosphere, Band10 is meant for treating Cirrus clouds (low thin clouds near earth's surface), and lastly Band11 is meant for cloud/ice/snow discrimination.

5.2 Evaluation of Results

While training the base models, Training accuracy was used as a criterion to judge the performance of the models. A very deep network with many layers, like Resnet50 was trained over large augmented and non-augmented image datasets. Due to the time constraint, remote nature of dissertation and lack of availability of local GPUs, very high Validation score was not possible to attain during model training for creating base models. However, very high training accuracies, in 90s, were attained over all the model configurations. This does make the base models less generalizable, however, models were able to learn enough features that can be transferred to a target task at a later stage.

For target task of land classification, using single band images from six bands (B02, B03, B04, B05, B08, and B12), F1 scores and overall accuracies were used to judge the model performances. Total twenty-four models were created over six training sets belonging to six bands. None of the model created achieved accuracies higher than 66%, though individual F1 scores of certain classes like Sea Lake and Residential did attained values in initial 90s.

Data augmentation done using Keras ImageDataGenerator created a huge difference in base model's capability to transfer general or relatable features. Dataset sizes were

increased to five folds using just geometric transformations as augmentation techniques. Augmented base models were the best performers for every band in both RGB and Greyscale feature space. Moreover, Greyscale models outperformed the RGB base models on every band and hence this is in line with the research question posed at the start – *Can a CNN model, pre-trained on single channel (grey-scale) images, improve the image classification accuracy of multispectral images, in comparison to a comparable model trained on colour images.* The answer to this question is yes, it can improve the classification accuracies. The major problem with these results is that the extent of positive impact of using single channel greyscale trained model to transfer features is yet to be fully understood due to time constraints and lack of computational resources available locally.

5.3 Strengths of Results

There are several strengths of these results. Firstly, it is observed during the study that the size of networks used does impact the final results. Using deep networks like Resnet50 and getting positive results has opened up the possibilities of using still larger networks to train far more generalizable single channel base models. On the same lines, since increasing the dataset size using image augmentations has increased the final F1 scores and overall and weighted accuracies, this knowledge can be used for future augmentation efforts in computer vision problems. More types of augmentations and a higher number of image counts in training and validation sets can be used expecting a positive outcome.

Further, the results have shown that *fine tuning* techniques have the ability to change the model performance as well as its ability to generalize well and in lesser number of epochs. In some cases, it took lesser than 25 epochs to reach the highest possible training and validation accuracies without causing overfitting.

Initial results suggest that all six bands that were used for the study have shown better performance with single channel base models, thus it can be generalized to a greater number of multispectral and hyperspectral bands. Also, since using greyscale as a

single channel gave great performance, other single channels can also be used to train the base models.

5.4 Limitations of Results

Higher computational power or easy availability of local GPUs and clusters could have given much more generalizable base models. Due to lack of time and computational resources, it was difficult to train the base models with still larger image databases, and more hyperparameter tunings. This would have given better end results. Online resources were used to supplement the computational power but due to a limitation of GPU runtime on such portals, long running models with larger number of parameters could not be tested. Reesnet50 was used mainly because of its faster speed in training over large image databases, other deep networks like Inception networks, VGG and GoogleNet could have also been tested with better local cluster availability. Also note that, a very large network like Resnet50 made training for base models a very difficult and lengthy task on personal machines.

Also, if time would have permitted, not just single bands but band combinations like shortwave infrared (SWIR 1, SWIR 2, and Red) and Color Infrared (NIR, Red, and Green) could have also been tested using available single channel base models. For same reasons, the study was conducted only using only six of the available thirteen multispectral bands.

6 CONCLUSION AND FUTURE WORK

6.1 Research and Experiment Overview

This research is done to improve the transferability of large computer vision models to target domains which work on single or multichannel images and have smaller labelled datasets. This research will help in improving deep learning model's performance in fields like Earth Observation and Medical Imagining. The idea was conceived when working on multispectral images, the researcher had to use RGB trained models for feature transfer instead of one which is more relevant to spectrums involved in the problem set.

Originally the datasets that were used are mini-ImageNet and Eurosat datasets. Mini-Imagenet is a natural image dataset which consists of 200 classes with 500 images in each, while Eurosat is the Remote Sensing image dataset which consists of 10 classes with 27,000 images in total. Actually, Eurosat has two components to it, one is the set of RGB images of different land-cover classes, while another one is the dataset of multispectral images of same RGB regions, but with .tif extensions. These two RGB datasets, Eurosat and mini-Imagenet are fused together to form one large database. Greyscale images are extracted from this large database and using them another Greyscale database is created. The idea is to use this as a single-channel images source. Later on, using both, RGB and Greyscale images, two new datasets were created by augmenting the available images, this increased the database size by a factor of five. Total classes were 110, with 100 were Imagenet classes while 10 Eurosat were classes.

This research was done by first creating four models using these four datasets. Same architecture or network configurations were used for all the four models. So, at the end of this exercise, one model was created on RGB images, another one was created on Greyscale images, another on Augmented RGB and yet another on Augmented Greyscale datasets. These were treated as base models to transfer features to multispectral target tasks.

Six bands were extracted from multispectral images consisting of 13 bands each and six new datasets of 27,000 images each were created using them. Four *base models* created earlier were used to transfer learned features and patterns to new target networks and performance was measured in the land-cover classification task. Performance measures like Precision, Recall, F1 Scores, overall weighted Accuracies and validation and training loss were used throughout the study.

Results have shown that single channel trained base models are better at transferring more relevant features to multispectral problem space like satellite imaging. Whole code is written by self, using Python 3.7, Tensorflow2.0 and Keras.

6.2 Contributions and Impact

Using this knowledge, of single channel trained base models being better at transferring features to a multispectral task with smaller labelled datasets than RGB trained base models, better remote sensing applications can be developed. Remote sensing has many applications in flood detection, coastline detection, urban and rural planning, and also in military and scientific research. All these sectors stand benefitted from this research.

6.3 Future work and Recommendations

Using the base models created during the research, similar analysis can be conducted on band combinations rather than using single extracted bands. Similarly, more single channels can be used to train the base models to test the best transferability among all available bands.

Lack of clusters limited the dataset size for creating base models and also the network size. More computational resources can be used to train with still large ImageNet ILSVRC images in greyscale mode to build a comparable Resnet50 as the standard one available in RGB mode. Likewise, more parameter tuning ca be experimented like dropout values, activations, different learning and decay rates, training with

momentum, different optimizers, different number, and type of hidden layers as well as different number of nodes between them.

7 BIBLIOGRAPHY

A. Berg, J. Deng, and L. Fei-Fei. Large scale visual recognition challenge 2010. www.imagenet.org/challenges. 2010

A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto, 2009.

A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV, 42(3):145–175, 2001.

A. Van Etten, D. Lindenbaum, and T. M. Bacastow. SpaceNet: A Remote Sensing Dataset and Challenge Series. arXiv:1807.01232 [cs], July 2018. arXiv: 1807.01232

B. Bischke, P. Helber, C. Schulze, V. Srinivasan, and D. Borth. The Multimedia Satellite Task: Emergency Response for Flooding Events. In MediaEval, 2017.

B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. In ICLR, 2017.

Bousmalis, K., Trigeorgis, G., Silberman, N., Krishnan, D., & Erhan, D. (2016). Domain separation networks

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in IEEE International Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

Chen, X.; Xiang, S.; Liu, C.L.; Pan, C.H. Vehicle detection in satellite images by hybrid deep convolutional neural networks. IEEE Geosci. Remote Sens. Lett. 2014, 11, 1797–1801.

Cheplygina, V. (2019). Cats or cat scans: transfer learning from natural or medical image source datasets? In arxiv:1810.05444 [cs.cv].

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation policies from data. CVPR, 2019.

D. Geman and B. Jedynak, “An active testing model for tracking roads in satellite images,” IEEE Trans. Pattern Anal. Machine Intell. vol. 18, pp. 1–14, Jan. 1996.

Daume III, H. (2007). Frustratingly easy domain adaptation. In In acl.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database

Donahue, J., Jia, Y., Vinyals, O., Hofman, J., Zhang, N., Tzeng, E., & Darrell, T. (2013). Decaf: A deep convolutional activation feature for generic visual recognition. In Corr, abs/1310.1531.

G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007. URL http://authors.library.caltech.edu/7694.

G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580, 2012

G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, and L. Zhang. Aid: A benchmark dataset for performance evaluation of aerial scene classification. arXiv preprint arXiv:1608.05167, 2016.

Ganin, Y., & Lempitsky, V. (2015). Unsupervised domain adaptation by backpropagation. In proceedings of the 32nd international conference on machine learning. (vol. 37).

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Lempitsky, V. (2016). Domain-adversarial training of neural networks. In Journal of machine learning research, 17, 1{35.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., & Brendel, W. (2019). Imagenet-trained CNN's are biased towards texture; increasing shape bias improves accuracy and robustness.

Gong Cheng, Junwei Han, and Xiaoqiang Lu. Remote sensing image scene classification: benchmark and state of the art. Proceedings of the IEEE, 105(10):1865–1883, 2017.

Goodfellow, Ian J., Pouget-Abadie, Jean, Mirza, Mehdi, Xu, Bing, Warde-Farley, David, Ozair, Sherjil, Courville, Aaron C., and Bengio, Yoshua. Generative adversarial nets. NIPS, 2014

He, K., Girshick, R., & Dollar, P. (2018). Rethinking imagenet pre-training. In arxiv preprint arxiv:1811.08883.

He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)

Introducing EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. Patrick Helber, Benjamin Bischke, Andreas Dengel. 2018 IEEE International Geoscience and Remote Sensing Symposium, 2018.

J. Dai, Y. Li, K. He, and J. Sun. R-fcn: Object detection via region-based fully convolutional networks. arXiv:1605.06409, 2016

J. Lemley, S. Bazrafkan, and P. Corcoran. Smart augmentation-learning an optimal data augmentation strategy. IEEE Access, 2017.

Jain, P. Schoen-Phelan, B. Ross, R. (2020b) Tri-Band Assessment of Multi-Spectral Satellite Data for Flood Detection.

Jain, P.; Schoen-Phelan, B.; Ross, R. Automatic flood detection in SentineI-2 images using deep convolutional neural networks. In Proceedings of the 35th Annual ACM Symposium on Applied Computing, Brno, Czech Republic, (15 September 2020a); pp. 617–623.

K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: Delving deep into convolutional nets, arXiv preprint arXiv:1405.3531.

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In ICCV, 2015.

K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In International Conference on Computer Vision, pages 2146–2153. IEEE, 2009

K. Nogueira, O. A. Penatti, and J. A. dos Santos. Towards better exploiting convolutional neural networks for remote sensing scene classification. Pattern Recognition, 61:539–556, 2017.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015

Kornblith, S., Shlens, J., & Le, Q. V. (2018). Do better imagenet models transfer better? In arxiv preprint arxiv:1805.08974.

Kung Jr, L. (2010). Understanding the biology of silage preservation to maximize quality and protect the environment. In In proceedings, 2010 california alfalfa forage symposium and corn/cereal silage conference, pages 1-2, visalia, ca.

Kurakin, A., Goodfellow, I., & Bengio, S. (2017). Adversarial examples in the physical world.

L. Gatys, A. Ecker, and M. Bethge. A neural algorithm of artistic style. Nature Communications, 2015

LeCun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proc IEEE 86:2278–2324

M. Castelluccio, G. Poggi, C. Sansone, and L. Verdoliva. Land use classification in remote sensing images by convolutional neural networks. arXiv preprint arXiv:1508.00092, 2015.

M. Mahdianpari, B. Salehi, M. Rezaee, F. Mohammadimanesh, and Y. Zhang, "Very deep convolutional neural networks for complex land cover mapping using multispectral remote sensing imagery," Remote Sensing, vol. 10, no. 7, p. 1119, 2018

Mikolov, T., Joulin, A., & Baroni, M. (2015). A roadmap towards machine intelligence. In arxiv preprint arxiv:1511.08130.

Moacir Ponti, Arthur A Chaves, F´abio R Jorge, Gabriel BP Costa, Adimara Colturato, and Kalinka RLJC Branco. Precision agriculture: Using low-cost systems to acquire low-altitude images. IEEE computer graphics and applications, 36(4):14–20, 2016

Ngiam, J., Peng, D., Vasudevan, V., Kornblith, S., Le, Q. V., & Pang, R. (2018). Domain adaptive transfer learning with specialist models. In arxiv preprint arxiv:1811.07056.

O. A. Penatti, K. Nogueira, and J. A. dos Santos. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 44–51, 2015

O'Byrne, P., Jackman, P., Berry, D., Franco-Penya, H., French, M., & Ross, R. (2019). Multispectral visual crop assessment under limited data constraints. In Conference papers.

Pan, S. J., & Yang, Q. (2010). A survey on transfer learning. In Ieee transactions on knowledge and data engineering -22(10) (p. 1345{1359).

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. 2017. EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification. arXiv preprint arXiv:1709.00029 (2017)

Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. In arxiv preprint arxiv:1712.04621.

R. de O. Stehling, M. A. Nascimento, and A. X. Falcao. A compact and efficient image retrieval approach based on border/interior pixel classification. In CIKM, pages 102–109, 2002.

Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised representation learning with deep convolutional generative adversarial networks

Raghu, M., Zhang, C., Kleinberg, J., & Bengio, S. (2019). Transfusion: Understanding transfer learning for medical imaging. In arxiv:1902.07208 [cs.cv].

Ravi, S., & Larochelle, H. (2017). Optimization as a model for few-shot learning. In In iclr 2017.

Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S. CNN Features off-the-shelf: An Astounding Baseline for Recognition. CoRR, abs/1403.6382, 2014.

Rusu, A. A., Vecerik, M., Roth• orl, T., Heess, N., Pascanu, R., & Hadsell, R. (2016). Sim-to real robot learning from pixels with progressive nets. In arxiv preprint arxiv:1610.04286.

Saikat Basu, Sangram Ganguly, Supratik Mukhopadhyay, Robert DiBiano, Manohar Karki, and Ramakrishna R. Nemani. 2015. DeepSat - A Learning framework for Satellite Imagery. CoRR abs/1509.03602 (2015)

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun., Y. (2014). Overfeat: Integrated recognition, localization and detection using convolutional networks. In In iclr.

Sun, B., Feng, J., & Saenko, K. (2016). Return of frustratingly easy domain adaptation. In proceedings of the thirtieth aaai conference on artificial intelligence (aaai-16).

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2016). Inception-v4, inception-ResNet and the impact of residual connections on learning. In arxiv preprint arxiv:1602.07261.

Torralba, A., & Efros, A. A. (2011). Unbiased look at dataset bias. In In 2011 ieee conference on computer vision and pattern recognition (cvpr).

Tzeng, E., Hofman, J., Zhang, N., Saenko, K., & Darrell, T. (2014). Deep domain confusion: Maximizing for domain invariance. In Corr.

Weixun Zhou, Shawn Newsam, Congmin Li, and Zhenfeng Shao. Patternnet: a benchmark dataset for performance evaluation of remote sensing image retrieval. ISPRS Journal of Photogrammetry and Remote Sensing, 2018.

Xian, Y., Schiele, B., Akata, Z., Campus, S. I., & Machine, A. (2017). Zero-shot learning – the good, the bad and the ugly.

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashionmnist: a novel image dataset for benchmarking machine learning algorithms. In arxiv preprint arxiv:1708.07747.

Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, “Caffe: Convolutional architecture for fast feature embedding,” arXiv preprint arXiv:1408.5093, 2014.

Y. LeCun, F.J. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In Computer Vision and Pattern Recognition,

2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on, volume 2, pages II–97. IEEE, 2004.

Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on, pages 253–256. IEEE, 2010

Yi Yang and Shawn Newsam, "Bag-Of-Visual-Words and Spatial Extensions for Land-Use Classification," ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), 2010

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In In advances in neural information processing systems, pages 3320-3328.

Zeiler, M. D. and Fergus, R. (2013). Visualizing and understanding convolutional networks. CoRR, abs/1311.2901.

Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2017). Understanding deep learning requires rethinking generalization.

Zhu, X. (2005). Semi-supervised learning literature survey.

Zhuang, F., Cheng, X., Luo, P., Pan, S. J., & He, Q. (2015). Supervised representation learning: Transfer learning with deep autoencoders. In Ijcai international joint conference on artificial intelligence, 4119{4125.