Technische Universität München
Methodik der Fernerkundung
Fakultät für Luftfahrt, Raumfahrt und Geödäsie
Univ.-Prof. Dr.techn. Mag.rere.nat. Roland Pail

Performance and Transferability Assessment of Convolutional Neural Network (CNN) Based Building Detection Models for Emergency Response

Dharani Deivasihamani

Master's Thesis

Master's Course in Earth Oriented Space Science and Technology

Supervisor(s):

1. Prof. Dr.-Ing. Günter Strunz
   Deutsches Zentrum für Luft- und Raumfahrt (DLR)

2. Dr. Marc Wieland
   Deutsches Zentrum für Luft- und Raumfahrt (DLR)

Date of Submission: July 18th, 2022

STATEMENT OF AUTHORSHIP

This thesis is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions.

Munich,

(DHARANI DEIVASIHAMANI)

ACKNOWLEDGEMENT

This Master's program has introduced me to some wonderful people without whom, its completion would have been extremely difficult. This is my sincere effort to thank all of them for having been there for me.

But first I would like to thank my family. A special thank you to my lucky charm all along, my mother for her complete support throughout my life nevertheless the last three years. If not for her having taken care of everything else and myself, I would not have been able to focus on my coursework with ease. She is my biggest pillar of strength. Her constant reassurance that everything will be alright, helped me get out of those moments of self-doubt.

A special mention to my father for checking in on me, asking if I had completed the pending work that I keep complaining about every now and then, from small assignments to this thesis. He has been a role model in showing me that things always work out for those with a clear moral compass, discipline and humility.

Another special thanks is to my brother for taking care of my family when I am here. The thought of him being around for my parents and grandparents when I am 7000 kilometers away definitely puts my mind at ease and has helped me work on my Master's in peace.

I would like to thank my supervisor, Prof. Dr.-Ing. Günter Strunz, who provided me with this interesting thesis topic. Having heard from peers in the department and having seen his Skype status turn red all the time, I can only imagine how busy his schedule is. Despite his busy schedule, he has been available for constant feedback on the progress of this thesis and has shown keen interest in my work.

Next, I would like to express my immense gratitude to Dr. Marc Wieland, my immediate supervisor at DLR, for having been extremely kind and approachable throughout the period of my Master thesis, for his patience with all my questions and doubts. Without his guidance and suggestions, I could not have successfully pulled this off.

My ESPACE peer and "Dewd", Sindhu must be thanked especially for her patience with me through all my meltdowns and also for her clever suggestions and support throughout my Master's. Another person who needs to be mentioned is Cocoa puffs for her sarcasm and memes that kept me going from the beginning of our program.

Finally, a very big thank you to Sowndharya, Sneha and Kiruthika, for being my few other pillars of strength after my family, for being my go-to people and for having given me the most precious of things, their unconditional love, time, and support all along.

ABSTRACT

Remote Sensing data from Earth Observation (EO) is used for a wide variety of applications. Over the last decade, in the event of a natural calamity, the importance of using geo referenced products from satellite and aerial imagery has been on the rise. They play a vital role in helping the first responders by providing valuable information in the form of hazard zone maps that help in relocation of people, in post disaster evaluation to get a better understanding of the impact on the disaster zone and in the rehabilitation and reconstruction of damaged property.

In remote sensing-based emergency mapping, there are major limitations during the acquisition and processing of earth observation data. In most cases, satellite data can be acquired only from that set of EO satellites that are in orbit over the hazard zone during the time of the disaster. This can be compensated by deploying sensors on board airplanes and Unmanned Aerial Vehicles (UAVs) like drones for data acquisition. This gives rise to an archive of multi modal data that have different acquisition geometry, radiometry, acquisition conditions and Ground Sampling Distance.

This forces the data processing and analysis team to be equipped with methods that can readily handle such versatile data. With the dominance of artificial intelligence in earth observation, this thesis focuses on developing a Convolutional Neural Network (CNN) model that provides a robust performance for detecting exposed buildings when subjected to optical data from different kinds of sensors and platforms.

This thesis starts with an approach of training a region-based network to obtain a baseline model, which then is improved gradually by using advanced techniques like data augmentation and fine tuning. A comprehensive performance evaluation is carried out under consideration of different training-testing scenarios. Furthermore, the influence of tile-size on the detection performance is tested. The resultant model after improvements is tested on an independent validation dataset acquired during rapid mapping activation of the Centre for satellite-based crisis information (ZKI) during the floods in Germany, July 2021. Contrary to intuition, the model owning the implementation of augmentation technique on the xView global dataset, shows the best performance for transferability. Due to resource limitation, the pipeline has been trained with a small sliver of the available dataset. The model weights obtained by retraining on the entire dataset with much powerful machines will provide new benchmarks for transferability models in object detection.

By combining the resultant exposure with hazard information, we can get a first insight into which areas are likely to be affected in the event of a catastrophe. The importance of this work is that it provides an up-to-date picture of the building stock compared to Open Street Map or cadastre data, at different phases of the disaster.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

1.1 MOTIVATION

Earth Observation (EO) data from remote sensing is widely used for creating geo referenced products to help emergency responders during a catastrophe. It is also used to analyze the aftermath of the disaster by time series analysis on pre and post disaster imagery that helps in the reconstruction of the damaged areas.

Disaster responders and the humanitarian community increasingly use Earth Observation (EO) satellite systems to assess the impact of and to plan and coordinate emergency response activities after major natural disasters around the world. EO systems provide response and relief workers with tools to lift the “fog off disaster”. EO satellites help overcome operational uncertainties after major disasters that impede emergency response because of limited, incomplete, and often contradictory ground information [Voigt et al., 2016].

Geo-information derived from remote sensing satellite data have been used for years to help organizing and coordinating rescue activities [Voigt et al., 2016]. After a natural disaster or humanitarian crisis, rescue forces and relief organizations are dependent on fast, area-wide, and accurate information on the damage caused to infrastructure and the situation on the ground [Yuan et al., 2021] .

Natural calamities are extreme events that cause unexpected damages to buildings, communication systems, agricultural land, forest, natural environment etc., The economic losses due to natural disasters has increased by a factor of eight over the past four decades [Westen, 2000]. Satellite and Aerial remote sensing data can be used in a wide range of scenarios pertaining to natural calamities. It provides a beneficial solution as it offers information over large area in short periods of time. It can be utilized in the various phases of disaster management, such as prevention, preparedness, relief, and reconstruction, especially in rapid assessment of disaster situations.

According to [Voigt et al., 2016], the availability of scientific and commercial polar orbiting EO satellite systems has increased during the past 15 years. These satellites are equipped with imaging sensors in the visible and near- to mid-infrared part of the electromagnetic spectrum or in the radar frequencies. Systems useful for disaster extent and impact mapping have a ground sampling distance (GSD) in the range of 0.3 m to more than 300 m. A team of experienced image analysts can take from 6 to 16 hours to extract the relevant information from new satellite imagery and turn it into geo-information products, such as maps, for situation centers or decision-makers. Re-programming the satellite systems and collecting imagery over the disaster site typically takes 1 or 2 days and is considered one of the more time-consuming parts of the

overall process. Many elements of the Satellite Emergency Mapping (SEM) production chain are becoming automated. When we look closely at the activation trend of the International Charter for Space and Major Disasters, the number of activations has increased from seven activations in 2000 to 123 activations in 2014.

1.2 THESIS OBJECTIVES:

In the event of a natural disaster, there are several factors that cannot be controlled pertaining to remote sensing data acquisition. The temporal resolution of the satellite in orbit restricts the amount of data available during a natural calamity. There is a limit to the amount of data that can be collected as it might take several hours to even days for the satellite to fly over the same ROI (Region of Interest). Sometimes, disasters occur for a few minutes and hence in such a situation, the above scenario tends to be costly. The above situation can be supported by additional data collected using optical sensors on board aerial imaging platforms like airplanes, helicopters, UAVs etc., that can be dispatched immediately and can be flown multiple times in the ROI.

The data processing team must be prepared to handle data that has been collected using different kinds of sensors, hence with different spatial resolutions, from different platforms hence with different acquisition geometry. During a natural calamity, the data processing team will be on a time crunch. In the context of object detection, while generating predictions to detect damaged buildings on this data, there will not be any time to train a model on this newly collected data.

Object detection is an important part of automating risk assessment from remote sensing data as identifying and localizing different kinds of object classes like buildings, cars etc., that are affected by a disaster, is an important part of resultant geo products like hazard zone maps that help the rescue teams. Object detection and classification are also important for estimation of the economic impact in post disaster analysis. From the second half of last decade, Convolutional Neural Networks (CNN) are the new state of the art object detection algorithms as they can automate the entire detection process with minimum amount of annotated data and provide predictions with much improved accuracies.

Hence, the core focus of the thesis is to develop an object detection model using CNN architecture to detect buildings exposed to natural disasters from aerial and satellite imagery datasets and make it transferable across different kinds of optical datasets.

The CNN model that is successfully developed to detect damaged buildings must also have robust performance on data that the model has not been introduced to before. This helps in impact assessment as we can get a first insight on the areas that are likely to be affected. In the event of a calamity, fusing the exposure results from the model with the already available hazard information will ultimately provide, up to date information on the building stock, during

different phases of the disaster. For example, informal settlements need to be identified as they are uniquely vulnerable to environmental hazards and believe that mapping their locations is important for finding those who are most at risk to sea-level rise and flooding.

2. STATE OF THE ART

This chapter outlines the state-of-the-art object detection algorithms. It starts with a review of the evolution of object detection from traditional methods in the past to heavily dominant CNN approaches in the current time. The chapter then tightens on the building detection methods from satellite and aerial imagery using

- Generic methods
- CNN based methods
- CNN based object detection methods on transferability assessment,

relevant to the objectives of the thesis.

2.1 OBJECT DETECTION EVOLUTION:

As mentioned in [Zou et al., 2019], most of the early object detection algorithms were built based on handcrafted features. Due to the lack of effective image representation at that time, people have no choice but to design sophisticated feature representations, and a variety of speed up skills to exhaust the usage of limited computing resources. Some of the traditional methods include Viola Jones Detectors, Histogram of Oriented Gradients (HOG) Detector, Deformable Part-based Model. Now the field is dominated by CNN architectures like one stage detectors, two stage detectors (Region Proposal networks). Prominent examples of one stage detector would be You Only Look Once (YOLO), Single Shot Multi box detector, Retina Net. Similarly for two stage detectors, very popular examples would be SPP Net, Fast RCNN (Region-based Convolutional Neural Network) and Faster RCNN as shown in [Figure 1]. Their performance across different standard datasets for object detection like VOC07 (Pascal Visual Object Classes - 2007, VOC12(Pascal Visual Object Classes - 2012) and COCO (Microsoft Common Objects in Context) is shown in [Figure 2].

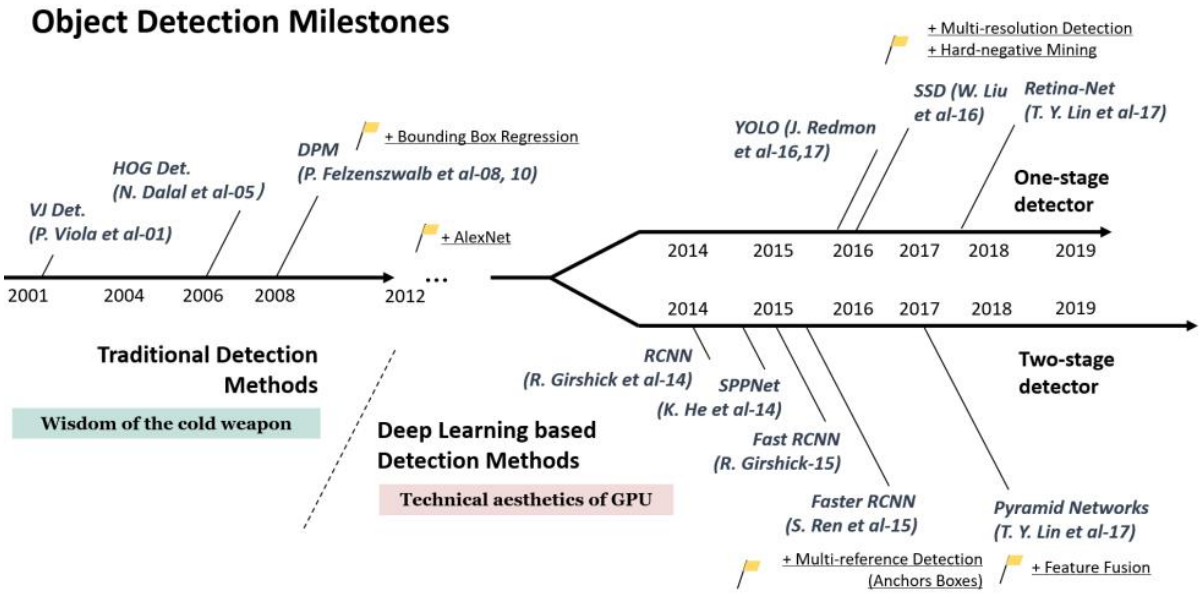


Figure 1 Different methods of object detection [Zou et al., 2019]

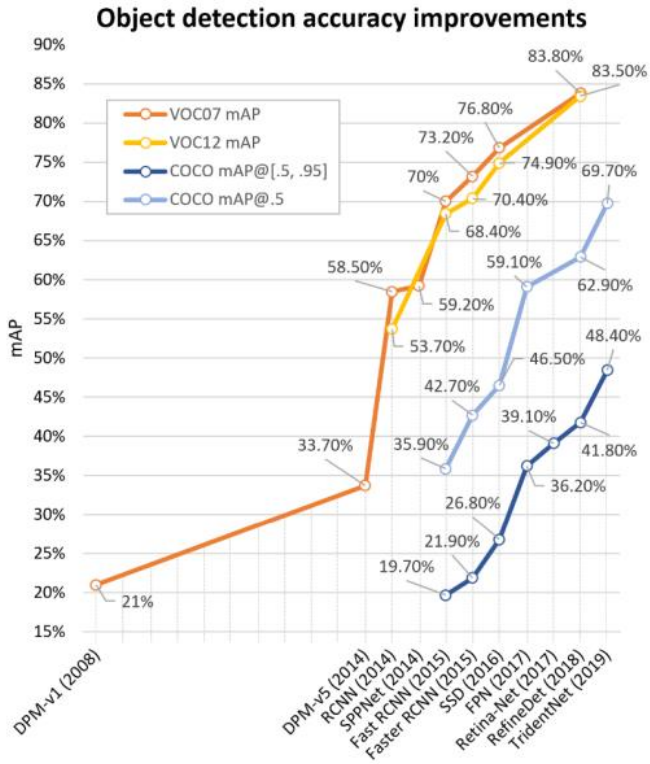


Figure 2 Performance of Object detection algorithms across different datasets [Zou et al., 2019]

2.2 GENERIC METHODS:

By generic methods, this part of the chapter deals with building detection methods that does not involve machine learning or CNN based approaches. These were the methods that existed before CNN methods took the reins of object detection.

In [Sirmacek and Unsalan, 2008], invariant colour features and shadow information are used for building detection. This method uses aerial images in the Red Green Blue (RGB) bands obtained over Istanbul area. The pixel value of red, blue and green bands are used to formulate two invariant colour feature indices which are then used along with shadow information and a box fitting approach to detect buildings from a simple thresholding operation. 162 out of 177 buildings are correctly detected but a few treetops are also detected as buildings bringing the false positive count to 25.

In [Saeedi and Zwick, 2008], automated building detection from aerial and satellite images is performed by implementing a two-step process. The first step is straight line extraction using Burns line detector algorithm which utilizes both the gradient magnitude and gradient orientation to form line support regions and eventually straight-line segments. The second step involves a line linking algorithm for linking the lines that belong to the same rooftop. Then a rooftop hypothesis is finally performed to distinguish these buildings. This paper uses a total of 6 scenes which has a total of 58 buildings of which 47 were detected with an accuracy of more that 90 percent. Number of false positives is 1 and number of false negatives is 4.

2.3 CNN BASED METHODS:

In [Chen et al., 2017], building detection is performed by using a Region based CNN. The focus of this paper is to induce directional feature to the detected buildings as horizontal bounding boxes are used to detect buildings even if the buildings are in a different orientation. This paper introduces a new oriented layer network for detecting the rotation angle of building based on the successful Visual Geometry Group Network (VGG-net) R-CNN model; the oriented rectangle is proposed to leverage the powerful R-CNN for remote-sensing building detection. The training dataset comprises of 2000 images and total number of aerial images used for testing is 4. The reported precision is 98.3 percent and recall is 96.9 percent.

In [Li et al., 2021], a straightforward comparison of state-of-the-art single stage detection models like Yolo V3, Single Shot Detector (SSD) and a two stage detection model (Faster RCNN) on greenhouse detection using satellite imagery has been performed. Two different datasets, one from Gaofen-1 satellite with a resolution of 2meters and a total of 413 samples and the other one from Gaofen-2 satellite with a spatial resolution of 1m and a total of 964 samples were used for the purpose. The Yolo v3 model seems to perform better with a Mean Average Precision (mAP) of

90.4 percent (when both the datasets are combined) than the SSD and Faster RCNN models with a difference of 1.1 percent in mAP.

In [Bai et al., 2020], an optimized faster RCNN method is used for building detection from remote sensing images. The dataset used is from low-altitude remote sensing images taken by UAVs. For the collected data set, the annotations were adopted in the same format as VOC2007 (Visual Object Classes 2007). Faster RCNN architecture comprises of two stages. A Region proposal network and a Fast RCNN together makes Faster RCNN. The residual layers of RPN is proposed to be replaced by a Dense Residual Net and the ROI pooling layer by ROI Align layer that uses bilinear interpolation, to improve performance. This modified Faster RCNN is claimed to have better performance by the authors. This model achieves an mAP of 82.1 percent after proposing the above modifications. The dataset seems to be quite small. Some of the accuracy comparisons are based on detecting a single object in a single image. The angle of acquisition is not nadir for a lot of images. But this method claims to preserve edge information by incorporating the above modifications.

2.4 TRANSFERABILITY ASSESSMENT CNN BASED METHODS:

Transferability assessment does not have much of a strong existing literature background in object detection with bounding boxes. The closest attempt is in instance segmentation and classification approaches.

In [Yang et al., 2021] , transferability of CNN models across different 3 different datasets during an earthquake is evaluated. The baseline models that are adjusted are VGG16, Inceptionv3, ResNet50 and DenseNet121 which are classification CNNs and not used for object detection. The geographical transferability of the model as to check its generalization across different ROIs is also evaluated. The first dataset is xBD satellite imagery. The second one is IKONOS satellite imagery of Wenchuan earthquake. This dataset was partitioned into 6 sub regions to evaluate the generalization across these regions. The third dataset is a single aerial image scene tiled into a few images. VGG16 and DenseNet11 perform slightly better while making predictions across different ROIs, but the model does not perform as good as it did across the same ROI. [Yang et al., 2021] mentions that the pre-trained model based on the xBD dataset had poor prediction on damaged buildings in the Beichuan area, and consequently, it could not be directly applied to an aerial photo. This indicates that even if the aerial photos have close to sub-meter spatial resolution compared to the xBD dataset, the performance of the model trained with samples from satellite images is not satisfactory when the model is applied to the aerial photos. Therefore, a small number of building samples obtained from the aerial images after the Earthquake are mixed with satellite-image samples to help the pre-trained network learn the characteristics from the aerial photo. The performance of the network is claimed to improve after the above step. But this ultimately shows the model performance is not robust in the transferability scenario.

In [Majd et al., 2019], a modified CNN is used to classify the input images into different kinds of objects namely, road, building, vegetation and shadow. This framework was utilized to classify different images from same sensor [the ISPRS two-dimensional (2-D) semantic labelling challenge dataset] (Aerial, Vaihingen, Germany) and also was tested on another sensor [pan-sharpened orthorectified multispectral sub-image from Worldview-2 (WV2) satellite imagery in an area of Tunis. According to [Majd et al., 2019], the object-based classification method attempts to classify the whole image into meaningful regions while preserving the precise edges of the targets of interest. It cannot produce satisfactory results, due to the lack of a certain level of expert knowledge on segmentation. Due to the demonstrated superiority of the CNNs in understanding a scene, it is essential to combine object-based classification with the Deep Learning (DL) method for building extraction in high-resolution imagery. The transferability of learned features increases as the distance between the base task (building type detection in the involved training dataset) and target task (building type detection in the other dataset) decreases.

2.5 RESEARCH QUESTIONS BASED ON LITERATURE REVIEW:

The CNN based object detection methods have a close relevance to the thesis objectives. Based on the results of the literature survey, a few research questions can be framed as follows,

[1] Based on the availability of abundant literature for two stage methods (region-based CNN to be specific) over single stage methods, for object detection from Satellite and Aerial imagery, where does the performance of the RCNN architecture in the form of a pretrained network on custom satellite and aerial imagery datasets stand?

[2] How does transforms induced on the imagery in the dataset affect the performance of a CNN model? Does it improve or deteriorate the performance of the existing model weights?

[3] What is the impact of fine tuning an RCNN model for object detection? Does it improve or deteriorate the performance of the existing model weights?

[4] What is the overall transferability of the model to a dataset (with different resolution and acquisition geometry), which it has never seen before?

3. THEORETICAL BACKGROUND

3.1 REMOTE SENSING

As described in [Wong et al., 2021], Remote sensing, itself, has rapidly evolved since the launch of the first Earth-observation satellite, Landsat, in 1972. Technological advancements over the years have gradually improved the ground resolution of satellite images, from 80 meters in the 1970s to 0.3 meters in the 2020s. Apart from the ground resolution, improvements have been made in many other aspects of satellite remote sensing. Also, the method and techniques of information extraction have advanced. However, to understand the latest developments and scope of information extraction, it is important to understand background information and major techniques of image processing. Satellite Remote Sensing enables us to recover contact-free large-scale information about the physical properties of the earth system from space [Zhu et al., 2021].

3.2 OPTICAL REMOTE SENSING

Electromagnetic radiation (EMR) from the sun which is absorbed and reflected differently at different wavelengths by various materials is the information carrier. The atmosphere has certain windows of transparency for EMR as shown in [Figure 3].

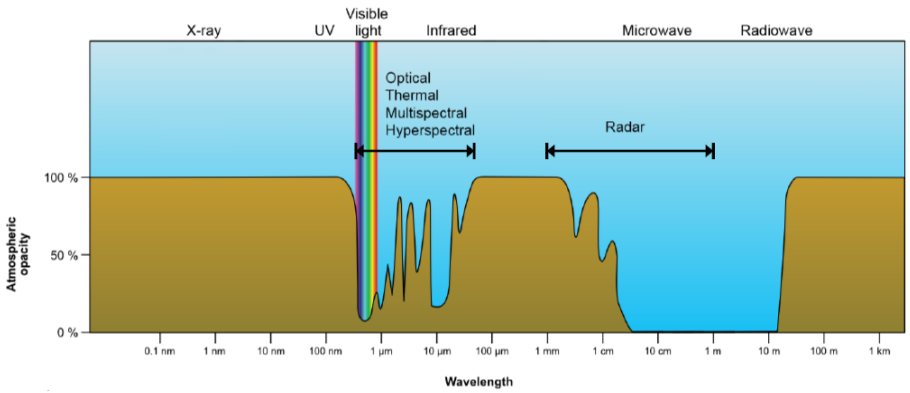


Figure 3 Atmospheric opacity vs Wavelength [Zhu et al., 2021]

Optical remote sensing makes use of visible, near infrared and short-wave infrared sensors to form images of the earth's surface by detecting this solar radiation reflected from targets on the ground as depicted in [Figure 4].

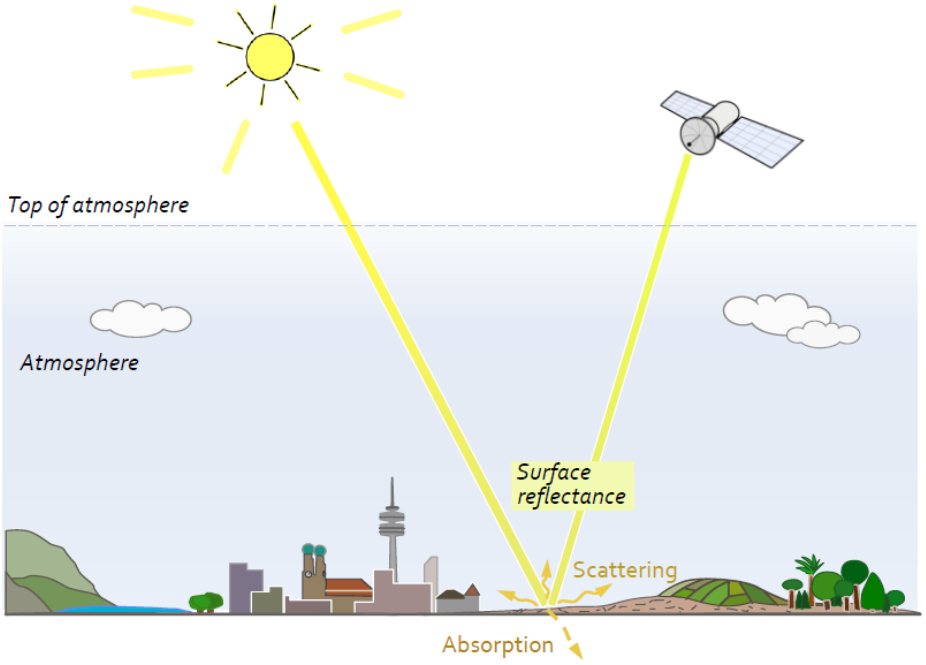


Figure 4 Optical Remote Sensing [Zhu et al., 2021]

Different materials reflect and absorb differently at different wavelengths. Thus, the targets can be differentiated by their spectral reflectance signatures in the remotely sensed images as shown in [Figure 5].

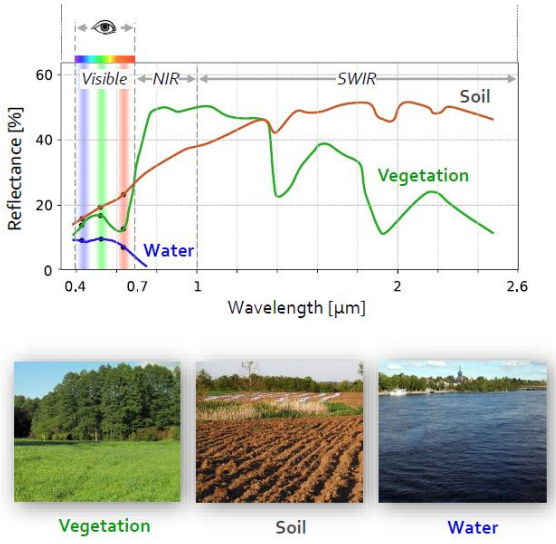


Figure 5 Spectral Reflectance Signatures of different objects [Zhu et al., 2021]

The sensors used on board satellites and airplane platforms for satellite and aerial imagery can be broadly classified into active and passive sensors based on the source of recorded EMR. Passive Sensors are the ones that record solar EMR reflected by objects. Examples would include Electro-optical cameras, Infrared/thermal cameras, Passive microwave sensors etc., Active sensors transmit the source EMR themselves and record its reflection by the objects of interest like Lidar and Synthetic Aperture Radar. Optical remote sensing makes use of passive sensors as it records the spectral reflectance of objects that reflect the solar radiation.

Based on the number of spectral bands used during the imaging process, optical remote sensing is divided into the following categories:

- Panchromatic
- Multispectral
- Superspectral and
- Hyperspectral remote sensing

PANCHROMATIC REMOTE SENSING:

It involves the acquisition of reflectance from the red, blue and green bands of the electromagnetic spectrum but as a single channel detection. This essentially means that the brightness value of each pixel observed comprises of the combined reflectance in red, blue and green bands and the image looks black and white. This allows for greater spatial resolution at the expense of being able to differentiate color. It stretches between 450 to 800 nanometers (nm) starting from the blue edge to the red edge of the spectrum.

MULTISPECTRAL REMOTE SENSING:

It involves the acquisition of reflectance in three to ten wide bands including blue, green, red and NIR wavelengths but as multi-channel detectors. Thus, for each pixel, the brightness information in multiple bands are available as a multi-layer image.

SUPERSPECTRAL REMOTE SENSING:

This involves recording of reflectance in more than ten bands when compared to multispectral sensors. The bands have narrower bandwidths, enabling the finer spectral characteristics of the targets to be captured by the sensor.

HYPERSPECTRAL REMOTE SENSING:

In this form of remote sensing, spectral reflectance is recorded in hundred or more contiguous bands. These numerous narrow bands in hyperspectral sensors provide a continuous spectral measurement across the entire electromagnetic spectrum. Therefore, this makes them more sensitive to subtle variations in reflected energy. Hyperspectral remote sensing has a high

spectral resolution that makes it possible to detect the spectral properties of objects and minerals. It gives better capability to see the unseen.

3.3 ACQUISITION PLATFORMS:

The type of remote sensing image that has been acquired depends on the type of optical sensor on board the acquisition platform. Based on the platform used for image acquisition, there are two kinds of optical remote sensing imagery namely

- Satellite imagery
- Aerial Imagery

SATELLLITE IMAGERY:

Satellite images are acquired from sensors mounted on a satellite. EO satellites cover a larger footprint with a larger swath width because of its orbit and altitude which is usually 200 to 600 kilometers (km) above Earth. Since it is always a tradeoff between the swath width and the resolution, larger the swath width, lower is the resolution of the acquired satellite imagery. This provides a natural restriction on the spatial resolution that can be achieved with satellite imagery. But in course of time, improvements in the optical sensor field has paved the way for VHR satellites that can provide products with resolution close to 20 centimeters.

AERIAL IMAGERY:

Aerial images are acquired with similar sensors but the platform for acquisition is usually a low flying airplane, helicopter, or Unmanned Aerial Vehicles (UAVs) like drones. These aircrafts fly at a much lower altitude, very close to the Earth and hence it covers a much smaller footprint. Since the swath width is small, very high-resolution imagery with resolution close to 10 to 15 centimeters can be obtained.

Hence based on the type of sensor and the type of platform used, different kinds of remote sensing datasets are created like Satellite Hyperspectral imagery, Aerial Multispectral imagery etc.,

3.4 MACHINE LEARNING FOR REMOTE SENSING:

Machine learning (ML) is the ability to learn without being explicitly programmed. A way in which ML algorithms differ from traditional algorithms is that traditional algorithms would take an input, apply a logic and provide an output whereas an ML algorithm would engulf both input and output, find out a logic and use it for a new input to predict its output. If it is simply put, ML tries to imitate the way humans learn, in understanding the pattern or model behind immense quantity of data that is ultimately the solution to a problem. It tries to fit, model and formulate

the equation from the answer and reuse the equation for generating predictions for a similar problem.

[Lary et al., 2016] states that Learning incorporates a broad range of complex procedures. ML is a subdivision of artificial intelligence based on the biological learning process. The ML approach deals with the design of algorithms to learn from machine readable data. ML covers main domains such as data mining, difficult-to-program applications, and software applications. It is a collection of a variety of algorithms (e.g., neural networks (NN), support vector machines, self-organizing map, decision trees, random forests, case-based reasoning, genetic programming, etc.) that can provide multivariate, nonlinear, nonparametric regression or classification. The modeling capabilities of the ML-based methods have resulted in their extensive applications in science and engineering.

[Camps-Valls, 2009] explains that the amount of remote sensing data that is available for problem solving and for real time applications is immense and hence machine learning comes into picture. Remote sensing data processing deals with real-life applications with great societal values. For instance, urban monitoring, fire detection or flood prediction from remotely sensed multispectral or radar images have a great impact on economic and environmental issues. To treat efficiently the acquired data and provide accurate products, remote sensing has evolved into a multidisciplinary field, where machine learning and signal processing algorithms play an important role.

MACHINE LEARNING METHODS:

The major difference between supervised and unsupervised methods is the availability of labelled information. Labelled information means the output for a certain record is known and is used as input to train the model. Once the model is trained, then it can be used to generate predictions on a similar problem. The two key learning paradigms in image processing tasks are supervised and unsupervised learning. ML is generally used for classification, regression, detection, clustering, association and segmentation issues. The ML algorithms used for the above can be classified into Supervised and Unsupervised methods.

SUPERVISED LEARNING:

[O'Shea and Nash, 2015] states that supervised learning is the learning through pre-labelled inputs, which act as targets. For each training example there will be a set of input values (vectors) and one or more associated designated output values. The goal of this form of training is to reduce the model's overall error, through correct calculation of the output value of training example by training. When a user knows the kind of expected output, it is supervised learning.

UNSUPERVISED LEARNING:

As mentioned in [O'Shea and Nash, 2015], unsupervised learning differs in that the training set does not include any labels. Success is usually determined by whether the network is able to reduce or increase an associated cost function. However, it is important to note that most image-focused pattern-recognition tasks usually depend on classification using supervised learning. When the user does now know what the expected output is, then it is unsupervised learning. In Unsupervised learning, the properties of the data is used for grouping and the total number of classes is unknown.

Classification and Regression fall under supervised ML techniques and Clustering and Association falls under unsupervised methods as depicted in [Figure 6].

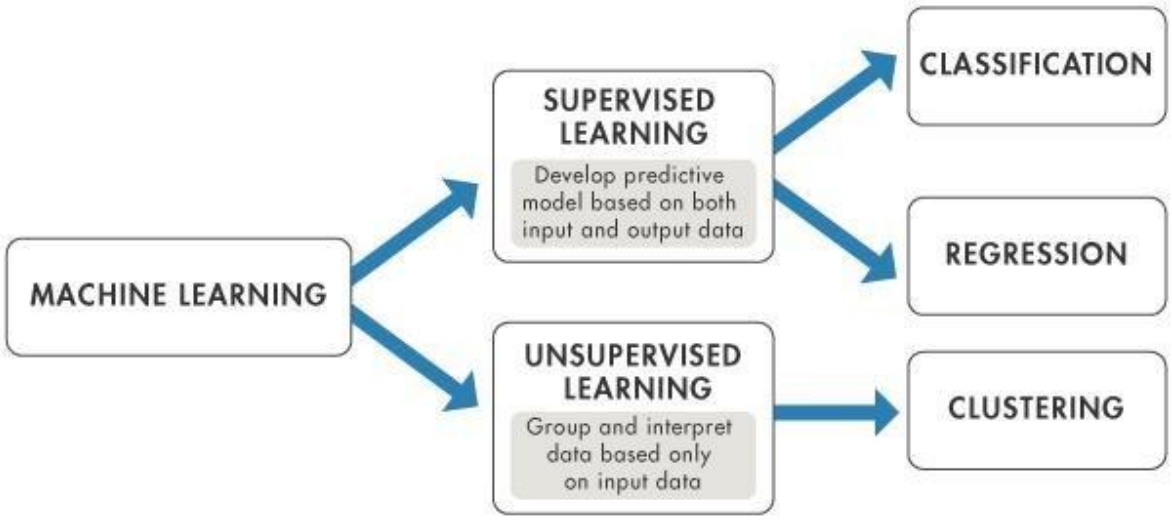


Figure 6 Supervise and Unsupervised learning [Bunker and Thabtah, 2019)]

This thesis focuses on bounding box regression and uses annotated (labelled) information for object detection under supervised learning. Further information on both supervised and unsupervised learning can be found in [Camps-Valls, 2009].

3.5 DEEP LEARNING:

Deep learning (DL) is a smaller part of ML which is in turn a smaller subsection of Artificial Intelligence (AI) as seen in [Figure 7]. Neural Networks (NN) in DL closely mirrors the functioning of neurons in human brain for problem solving. These neurons form a neural network which has multiple layers to take enormous amounts of data as input, process and finally provide the output at the output layer.

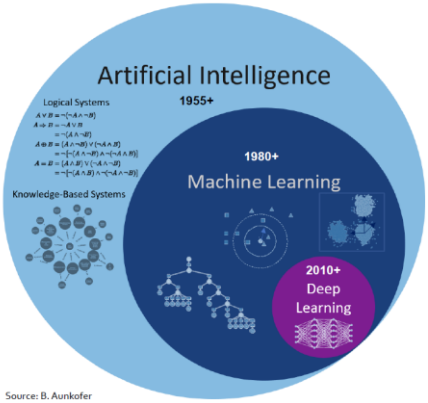


Figure 7 Relating AI, ML & Deep learning [Zhu et al., 2021]

Deep learning networks have an input layer and a fully connected output layer with a network of layers in the middle. Each of these layers have a number of neurons. Each neuron in a layer is connected to each neuron in the next layer. The simplest visualization for a small single layer NN is shown in [Figure 8].

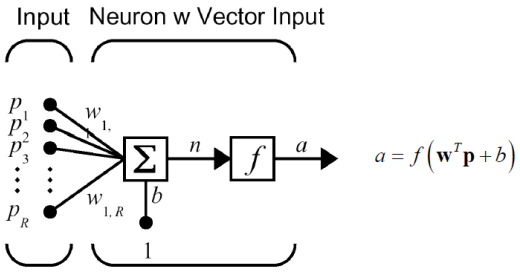


Figure 8 Single neuron network [Bamler et al., 2021]

P is the input matrix. W is the weight matrix and b is the bias. At the cumulation point, we end up with W*P + b (w1p1 + w2p2 + w3p3 + …. + wRpR + b). This value is then passed through f which is the activation function. It is the most important feature of the NN as it decides whether the particular neuron is activated or not. Different kinds of activation functions like linear, sigmoid and Rectified Linear (ReLU) are used for this purpose. The output of the activation function ‘a’ is provided as input to the parallel neuron in the next layer. When these layers are stacked together, they form a multi layered NN as shown in [Figure 9].

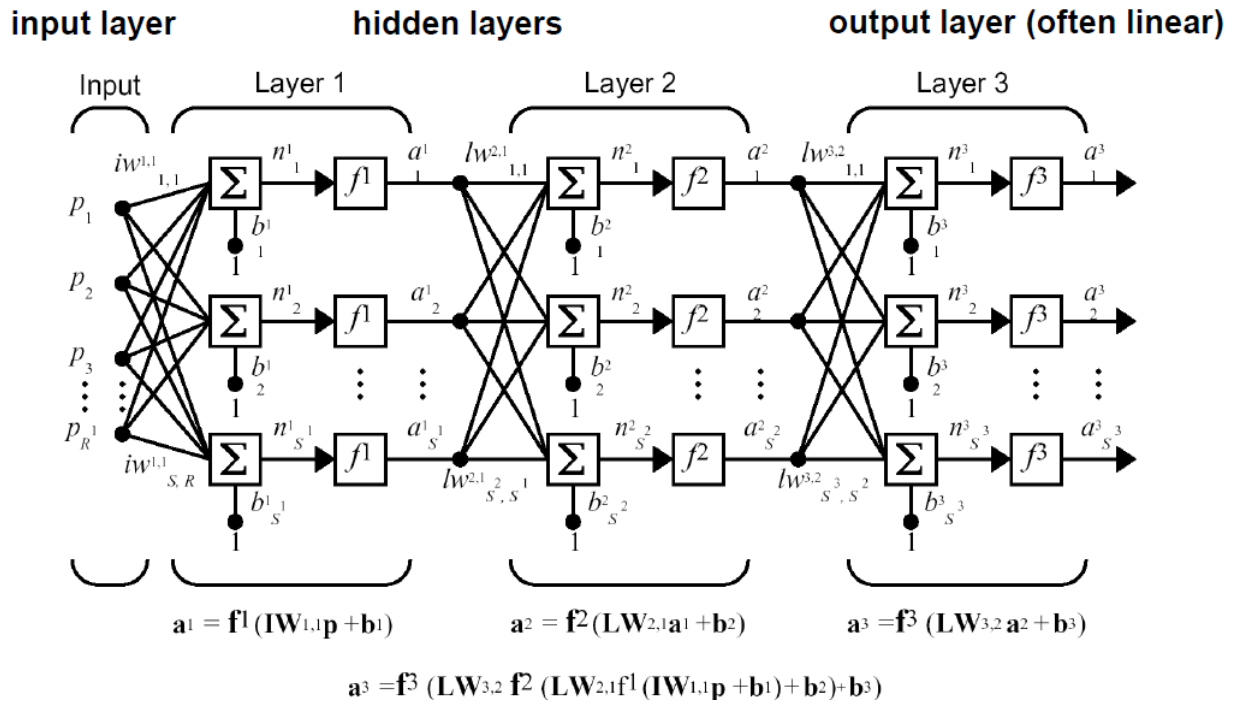


Figure 9 Multiple layer Neural Network [Bamler R 2021]

CONVOLUTIONAL NEURAL NETWORKS:

As explained in [O'Shea and Nash, 2015], CNNs are quite analogous to the general Artificial Neural Networks (ANNs) as they follow the same procedure of receiving an input and performing the operation of scalar product followed by a nonlinear activation function. The work is satisfiable till slightly large image data comes into picture for applications like object detection and pattern recognition. For example, when image tiles of size 64 X 64 are used as input, the number of weights on just a single neuron of the first layer increases substantially to 12,288. Thus, ANNs struggle with the computational complexity that arises with this and to be able to handle this complexity, the network would need to be much bigger than the regular size.

In CNN architecture, the layers within the CNN are comprised of neurons organized into three dimensions, the spatial dimensionality of the input (height and the width) and the depth. The depth does not refer to the total number of layers within the ANN, but the third dimension of an activation volume. Unlike standard ANNs, the neurons within any given layer will only connect to a small region of the layer preceding it. In practice this would mean that for the example given earlier, the input 'volume' will have a dimensionality of 64 × 64 × 3 (height, width and depth), leading to a final output layer comprised of a dimensionality of 1 × 1 × n (where n represents the possible number of classes) as we would have condensed the full input

dimensionality into a smaller volume of class scores filed across the depth dimension[O'Shea and Nash, 2015].

ARCHITECTURE:

[O'Shea and Nash, 2015, Page 4,5] shows CNNs are comprised of three types of layers. These are convolutional layers, pooling layers and fully connected layers. When these layers are stacked, a CNN architecture has been formed as shown in [Figure 10].

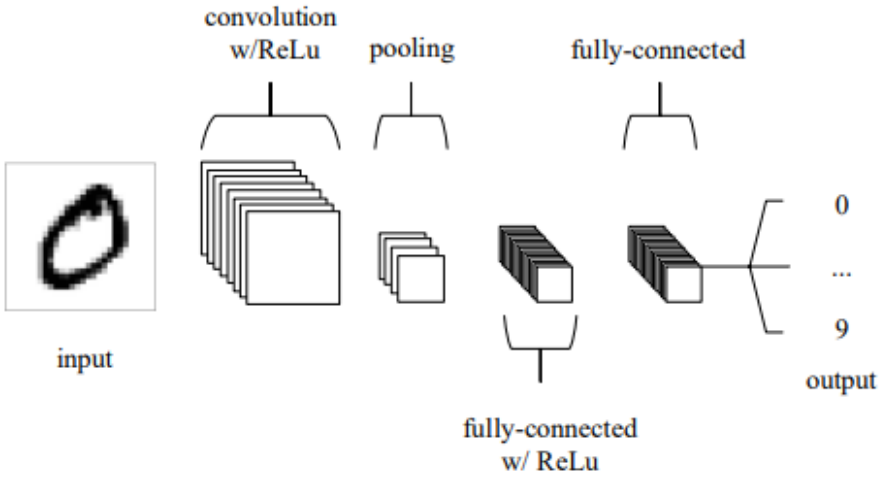


Figure 10 A simple CNN architecture comprising 5 layers [O'Shea and Nash, 2015]

The basic functionality of the example CNN above can be broken down into four key areas.

1. As found in other forms of ANN, the input layer will hold the pixel values of the image.

2. The convolutional layer will determine the output of neurons of which are connected to local regions of the input through the calculation of the scalar product between their weights and the region connected to the input volume. The rectified linear unit (commonly shortened to ReLu) aims to apply an 'elementwise' activation function such as sigmoid to the output of the activation produced by the previous layer.

3. The pooling layer will then simply perform down sampling along the spatial dimensionality of the given input, further reducing the number of parameters within that activation.

4. The fully connected layers will then perform the same duties found in standard ANNs and attempt to produce class scores from the activations, to be used for classification. It is also suggested that ReLu may be used between these layers, as to improve performance.

CNN models show that they generalize well to a wide range of tasks and datasets, matching or outperforming more complex recognition pipelines built around less deep image representations [Simonyan and Zisserman, 2014]. As mentioned in [Krizhevsky et al., 2012], to learn about thousands of objects from millions of images, we need a model with a large learning capacity. However, the immense complexity of the object recognition task means that this problem cannot be specified even by a dataset as large as ImageNet, so object detection model should also have lots of prior knowledge to compensate for all the data that is missing. Convolutional neural networks (CNNs) constitute one such class of models. Their capacity can be controlled by varying their depth and breadth, and they also make strong and mostly correct assumptions about the nature of images (namely, stationarity of statistics and locality of pixel dependencies).

4. DATASETS

The thesis objective focuses on building detection using a transferable CNN model from satellite and aerial imagery that has been obtained from different sensors in different acquisition conditions and regions of interest. For this purpose, three different datasets (Bundesamt für Kartographie und Geodäsie) BKG, xView and a disaster dataset from Zentrum für satellitengestützte Kriseninformation (ZKI) were provided.

4.1 BKG DATASET:

Due to a framework agreement, the Center for Satellite-based Crisis Information (ZKI) has the option for accessing geodata from the Federal Agency for Cartography and Geodesy (BKG) for research purposes. The digital orthophotos of the "Digital Orthophotos 2020 (DOP20)" collection have selected samples that have been stratified from different data sources that are representative of the building detection problem in the context of crisis mapping. [Figure 11] shows the distribution of samples of these samples across different regions of Germany.

The data pool includes

- aerial photographs of the entire German federal territory with a ground resolution of 0.2 m from the year 2020 in the RGB, NIR and CIR bands
- georeferenced surrounding polygons of building outlines for federal territory of Germany, which is official house perimeters, HU-DE.
- the Corine Land Cover Product (CLC) of the Copernicus Land Monitoring Service with 37 land cover classes

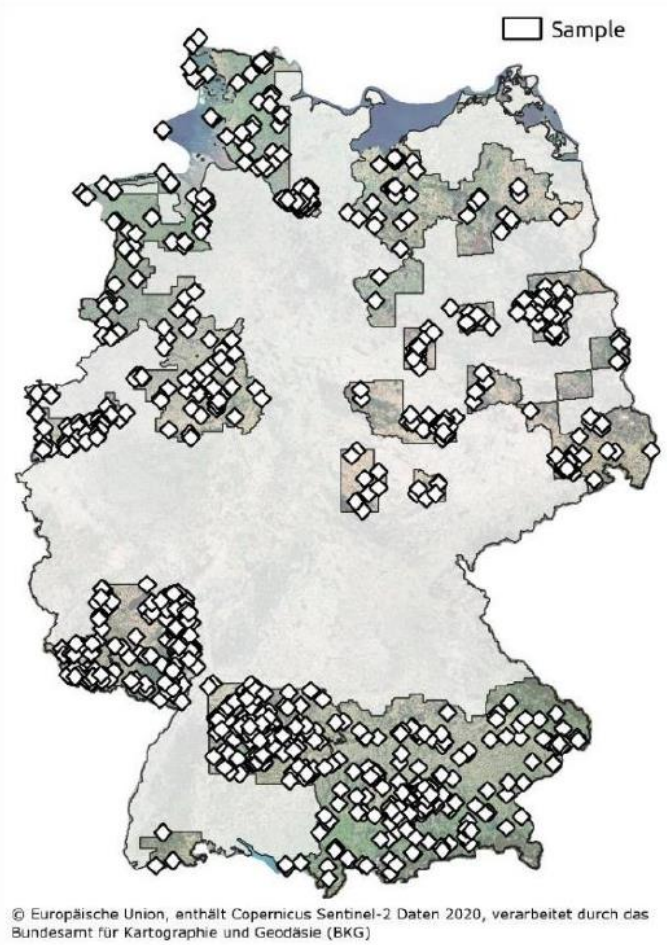


Figure 11 Distribution of the samples across Germany. Areas where no 2020 aerial photographs are available are shown semi-transparently.

The dataset comprises of several object classes treated as hyper classes e.g., Building which is associated to several instances of the same class. E.g., Church, School etc., are different instances of the building object hyper class. The number of object instances per sub class of building hyper class is shown in [Figure 12].

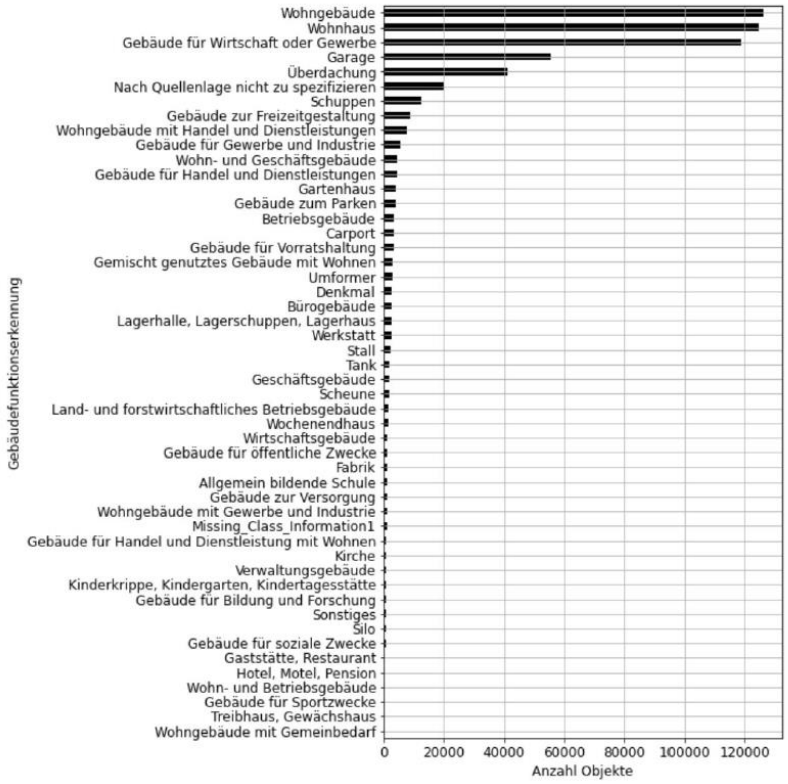


Figure 12 Objects per category (Building Hyperclass)

Each scene has been tiled to 81 image tiles tiled to a uniform tile size of 512 X 512 pixels and each of their corresponding geospatial information has been provided in a GEOJSON file. The images are provided in Tagged Image Format (.TIF) and the annotations for ground truth are provided in the standard Microsoft Common Objects in Context (COCO) format in a JavaScript Object Notation (JSON) file. The COCO dataset provided by Microsoft is used widely for object detection and instance segmentation. This dataset contains photos of 91 object types with a total of 2.5 million labeled instances in 328000 images [Lin et al., 2014].The annotation for this dataset is used as a standard benchmark by different object detection models and is termed the COCO format. The "COCO format" is a specific JSON structure dictating how labels and metadata are saved for an image dataset. The total number of image tile samples in the

- Training Split is 51, 595
- Validation Split is 15,390
- Testing Split is 15,200.

Sample images with annotations for this dataset can be found in [Figure 13], [Figure 14].

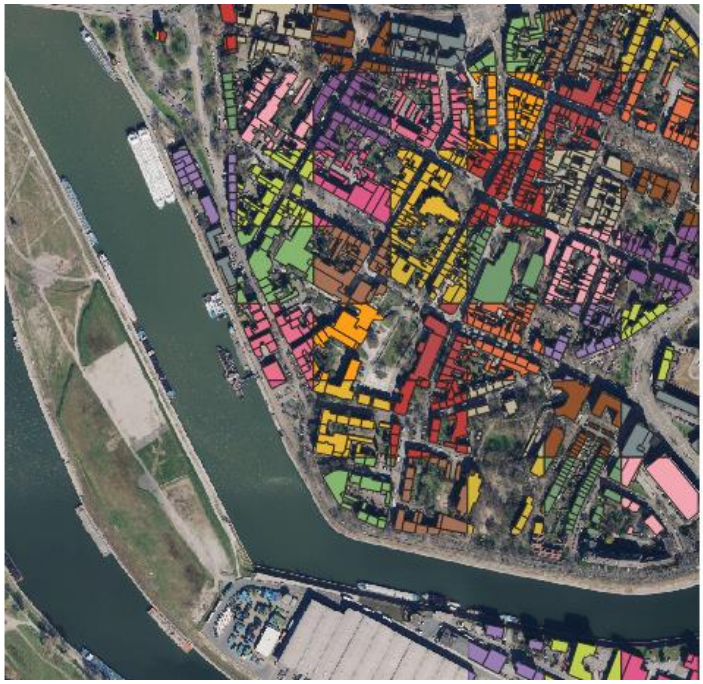


Figure 13 Sample image with annotations for scene dop20_rgb_32342_5702_1 in BKG dataset. (Each polygon represents a different building. The colours do not indicate any key difference)

Figure 14 Sample image with annotations for scene dop20_rgb_32491_5766_1 in BKG dataset (Each polygon represents a different building. The colours do not indicate any key difference)

4.2 XVIEW DATASET:

xView is one of the largest and most diverse publicly available object detection datasets to date, with over 1 million objects across 60 classes in over 1,400 square kilometers of imagery. It is collected from WorldView-3 satellites at 0.3m ground sample distance, providing higher resolution imagery than most public satellite imagery datasets. Several object detection datasets exist in the natural imagery space, but there are only a very few for overhead satellite imagery. The public overhead datasets in existence typically suffer from low class count, poor geographic diversity, few training instances, or too narrow class scope. xView remedies these gaps through a significant labeling effort involving the collection of imagery from a variety of locations and the use of an ontology of parent and child-level classes [Lam et al., 2018]. XView datasets used 846 square kilometers in its training phase, 281 square kilometers in its validation phase and 273 square kilometers in its testing phase. [Figure 15] shows distribution of xView dataset across the globe.

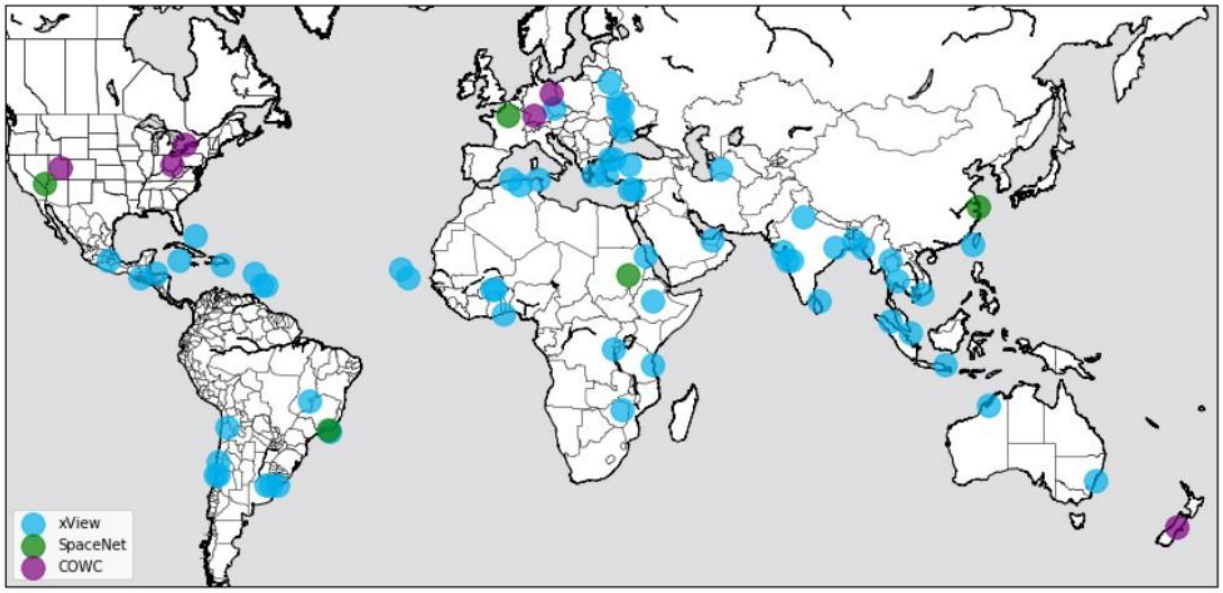


Figure 15 Distribution of xView data across the globe [Lam et al., 2018]

It includes objects of 8 object classes (like a hyper class), e.g., Passenger Vehicle, Building, Truck, Seagoing Vessel, Engineering Vessel, Fixed Wing, Rail Vehicle Class and Unclassified. Each object class has at least two instances (like a sub class). XView Data has a total of 62 sub classes. E.g.: The rail vehicle class has five sub classes namely passenger car, freight car, flat car, tank truck and locomotive. The building class has five sub classes namely buildings, buildings with damage, stables, huts and airplane hangar. These sub classes were grouped together to extract the building class to be used for the experimental setup of this thesis. The buildings sub class has

103 polygons and is the most data-rich sub class along with the Small Cars sub class as seen in [Figure 16].

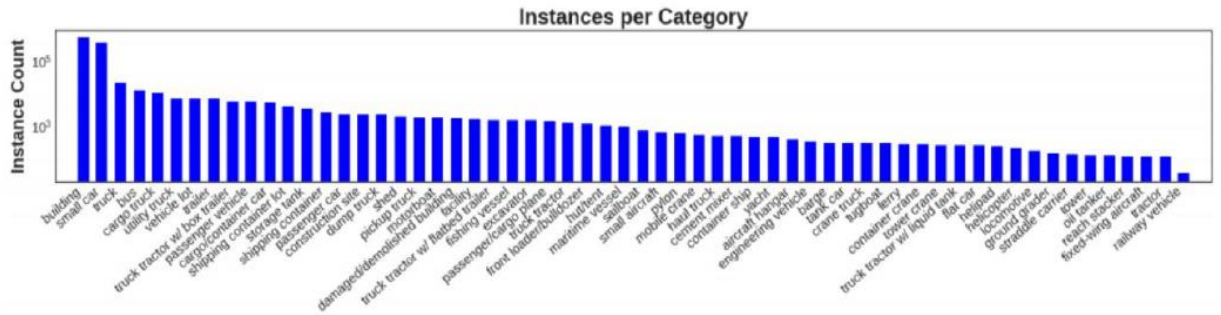


Figure 16 Number of object instances per sub class for all classes in the xView Dataset [Lam et al., 2018]

Each scene has been tiled to 25 or 30 image tiles tiled to a uniform tile size of 512 X 512 pixels and each of their corresponding geospatial information has been provided in a GEOJSON file. The images are provided in TIF format and the annotations for ground truth are provided in the standard Microsoft COCO format in a JSON file for the xView dataset as well.

The total number of image tile samples in the

- Training Split is 23,321
- Validation Split is 1132
- Testing Split is 998

Sample image tiles of the xView dataset with their annotations in GEOJSON files visualized in Quantum Geographical Information System (QGIS) geospatial software can be seen in [Figure 17] and [Figure 18].

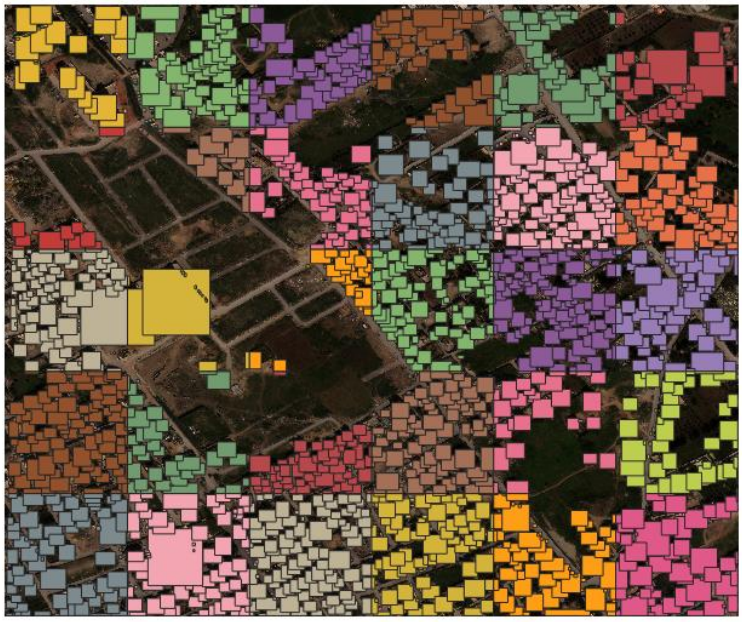


Figure 17 Sample image file of scene 102 in xView dataset (Each polygon represents a different building. The colours do not indicate any key difference)

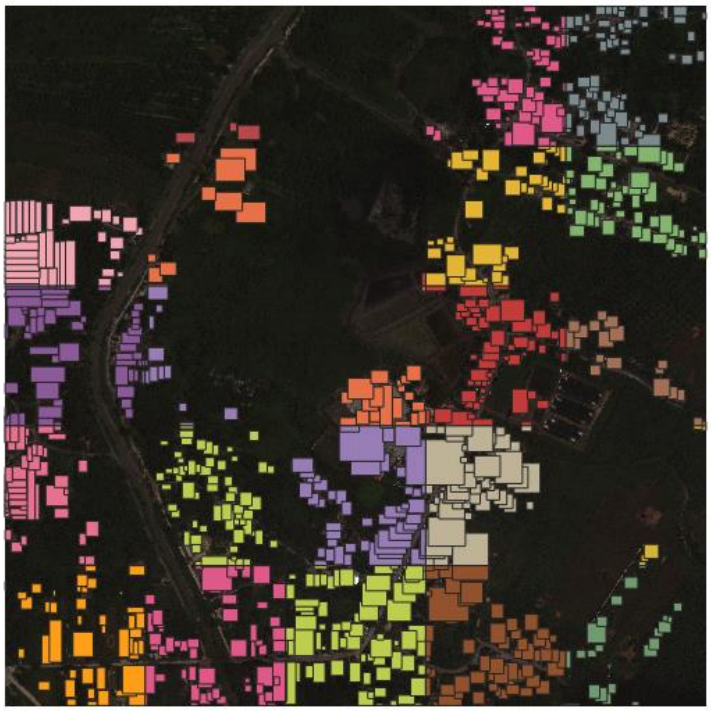


Figure 18 Sample image file of scene 389 in xView dataset (Each polygon represents a different building. The colours do not indicate any key difference)

4.3 ZKI DATSET:

During July 2021 in western Germany, prolonged heavy rain had caused strong flooding in the North Rhine-Westphalia and Rhineland-Palatinate regions. The Zentrum für satellitengestützte Kriseninformation (ZKI), which is the Center for Satellite based Crisis Information supports the helpers with a range of activities pertaining to rescue operations like processing and evaluating satellite data. Deutsches Zentrum für Luft- und Raumfahrt (DLR) which is the German Aerospace Center acquires aerial photographs within a very short time to be processed into maps. The ZKI passes on this valuable information to the Federal Office for Civil Protection and Disaster Assistance (BBK) for further assistance. In addition to the flood information derived from Sentinel 1 satellite for regions in North Rhine-Westphalia using automated methods, the ZKI presents, DLR aerial image data with resolutions of ten to 15 centimeters recorded with a camera from the Remote Sensing Technology Institute (IMF).

The in-house "3K Camera" and its successor, the "4k Camera System" on an EC 135 helicopter, were used for this purpose. The flood plains in Rhineland-Palatinate region, including the entire Ahr Valley from Müsch to the mouth in Sinzig, were again recorded by DLR on the 20th of July 2021 using the aerial camera system MACS (Modular Aerial Camera System) equipped with multiple sensors in the visible and near-infrared spectral range from the Institute of Optical Sensor Systems in Berlin. It has an integrated data processing unit that generates georeferenced images in real time, which are processed directly into different geodata products later. This georeferenced dataset of digital orthophotos was used as an independent test set and hence was not used in the training phase of the experiments. The trained model weights would be tested on this dataset to evaluate their transferability performance as it is the main objective of the thesis.

The annotations for this dataset provided as a JSON file. Unlike the BKG and xView datasets, this dataset includes additional information about the level of damage of the buildings along with the bounding box coordinates in COCO format. Different class values for buildings with different levels of damage is provided. Since this thesis does not focus on classification of damaged buildings, all the building classes are used for generating predictions. The view of destroyed roads in Ahr valley during the flooding is shown in [Figure 19]

Figure 19 Destroyed places and roads in the Ahr valley (view from the helicopter on 16.07.2021)

All the contents of this dataset, the image tiles (TIF files) which are tiled to a uniform tile size of 512 X 512 pixels, and their corresponding GEOJSON files are stored in the same folder. The total number of samples in this dataset is 1126 that has been entirely used as test split. The TIF file of the scene dlr_luftbild_16_07_2021_rheinland_pfalz_1 visualized in QGIS without its annotations is seen in [Figure 20] and a small, zoomed layer of the same scene with its annotations visualized in QGIS can be seen in [Figure 21]. The polygons in different colours simply represent the predictions on different buildings. The different colours do not represent any key difference.

Figure 20 ZKI Scene dlr_luftbild_16_07_2021_rheinland_pfalz_1

Figure 21 A zoomed in view of ZKI Scene dlr_luftbild_16_07_2021_rheinland_pfalz_1. (Each polygon represents a different building. The colours do not indicate any key difference)

5 METHODOLOGY

5.1 OBJECT DETECTION:

Object detection is one of the key focused machine learning algorithms because of its demanding application in a wide variety of fields like autonomous driving, agriculture, damage assessment etc., The problem definition of object detection is to determine where objects are located in a given image (object localization) and which category each object belongs to (object classification) [Zhao et al., 2019].Object detection can be understood as a combination of object classification and object localization. Object localization refers to localizing the position of an object in an image using a bounding box. Classification refers to distinguishing the object from other object classes present in the image. Object detection is localizing the object using a bounding box and classifying the detected object within the bounding box to one of the object classes such as building, car, person, background etc.,

Generic object detection aims at locating and classifying existing objects in any one image and labeling them with rectangular Bounding Boxes to show the confidences of existence [Zhao et al., 2019]

5.2 MODEL SELECTION:

As explained in Chapter 2, the state-of-the-art object detection algorithms can be classified into two categories namely single stage detectors and two stage detectors. Single stage detectors carry out the entire object detection pipeline in a single CNN structure whereas a two-stage detector has a Region proposal Network (RPN) and then an object detection network. Few examples of single stage detectors are SSD, Yolo etc., Similarly examples of two stage detectors are RCNN, Fast RCNN, Faster RCNN etc.,

It is always a trade – off between the processing speed and the accuracy of the model. Single stage detectors focus mainly on improving the processing speed and hence the accuracy is slightly lower compared to two stage detectors. These models are preferred in real time detection where predictions have to be generated immediately on the fly or in an on-board processing scenario where a slight decrease in accuracy is affordable.

Two stage detectors have a comparatively better accuracy, but their processing times are longer. These models are preferred in a situation where the datasets are already acquired and available, hence there is a liberty on the processing time.

For the thesis objectives, two models that were closely competing were Yolo v5 and Faster RCNN. Faster RCNN model has been selected for this thesis based on the following reasons.

- The datasets have been pre – processed and provided hence a slight increase in processing time is affordable.
- The models trained on these datasets for the thesis might be used later for damage assessment and hence any decrease in accuracy is not affordable.
- Based on the literature survey, during the beginning of the thesis, several papers about building detection using Faster RCNN were available whereas as much literature for object detection using Yolo v5 was not available on aerial and satellite imagery datasets but was available on normal image datasets comprising of daily life objects.

5.3 FASTER RCNN:

As the model architecture that has been selected for the problem statement of the thesis, the following part of the report explains the basic faster RCNN architecture.

Traditional object detection algorithms comprise of three distinct steps depicted in [Figure 22] namely

- Generating Region Proposals (candidates that might have objects within them) using Selective search or edge box algorithms.
- Image descriptors like HOG are used for extracting a fixed length feature vector from these region proposals.
- This feature vector is then used to assign each region proposal to either the background class or to one of the object classes by a classification model like Supervised Vector Machine (SVM).

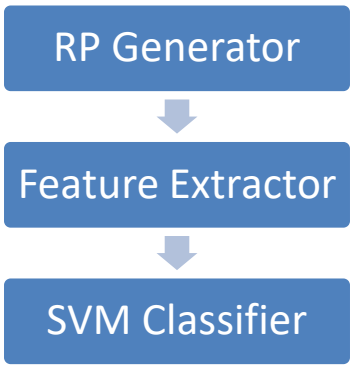


Figure 22 Object Detection Pipeline

The Region based Convolutional Neural Networks (RCNN) is a deep convolutional neural network that comes into picture in 2014, to replace this traditional pipeline with a CNN. It is quite similar to the traditional pipeline, but the difference is that the feature extraction part is solely

performed by CNNs instead of image descriptors like HOG. The RCNNs had a few limitations such as,

1. The region proposals still depended on Selective Search algorithm which is time consuming.
2. The detected features stored in disk cache for classification consumed memory.
3. Each Region Proposal from selective search is fed independently to CNN for feature extraction which is time consuming as there are close to 2000 region proposals generated by selective search per image.

These limitations are overcome by the introduction of Fast RCNN.

Fast RCNN proposes an ROI Pooling layer that extracts the feature vectors from all the 2000 region proposals of a single image simultaneously thus significantly cutting down the processing time. The multiple steps in the object detection pipeline is replaced by a single stage CNN. This removes the need to cache the detected features for classification.

As explained in [Girshick, 2015] and depicted in [Figure 23], a Fast R-CNN network takes as input an entire image and a set of object proposals. The network first processes the whole image with several convolutional (CONV) and max pooling layers to produce a convolutional feature map. The proposed ROI pooling layer uses max pooling to convert the features inside any valid region of interest into a small feature map with a fixed spatial extent of H × W (e.g., 7 × 7), where Height, H and Width, W are layer hyper-parameters that are independent of any ROI. In this paper, an ROI is a rectangular window pooled into a convolutional feature map. Each ROI is defined by a four-tuple (r, c, h, w) that specifies its top-left corner (r, c) and its height and width (h, w). ROI max pooling works by dividing the h × w ROI window into an H × W grid of sub-windows of approximate size h/H × w/W and then max pooling the values in each sub-window into the corresponding output grid cell. Pooling is applied independently to each feature map channel, as in standard max pooling. The ROI layer is simply the special case of the spatial pyramid pooling layer used in SPPnets in which there is only one pyramid level.

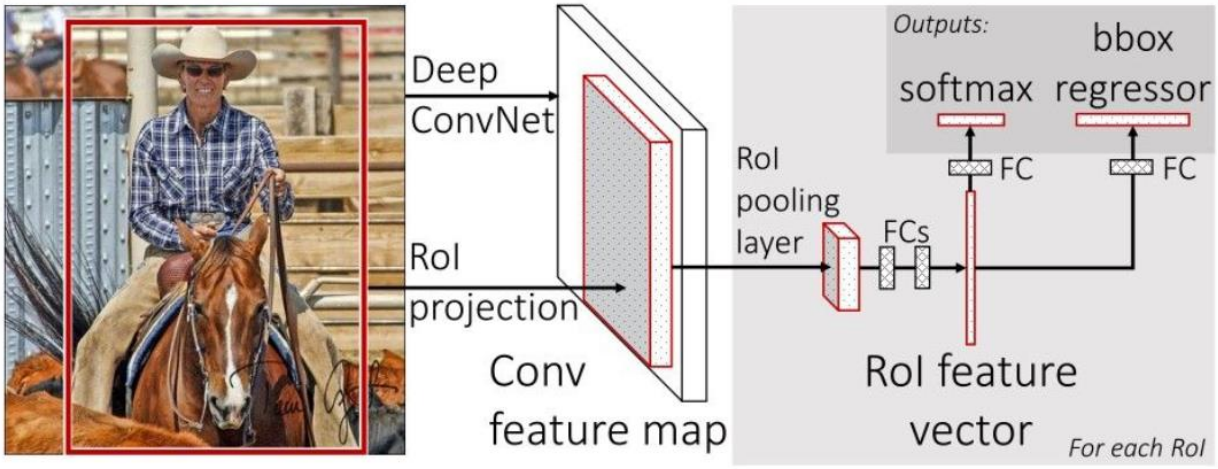


Figure 23 Fast RCNN architecture [Girshick, 2015.]

As observed in [Girshick, 2015],

- An input image and multiple regions of interest (ROIs) are input into a fully convolutional network.
- Each ROI is pooled into a fixed-size feature map and then mapped to a feature vector by fully connected (FC) layers.
- The network has two output vectors per ROI: softmax probabilities and per-class bounding-box regression offsets.
- The architecture is trained end-to-end with a multi-task loss

The Faster RCNN architecture was built as an extension to the existing architecture of Fast RCNN. The RCNN and Fast RCNN algorithms, still depend on the selective search algorithm for region proposal, which then gets fed to a CNN for classification. Faster RCNN exploits the use of Region Proposal Networks (RPN) replacing selective search. An RPN is a is a fully convolutional network that generates proposals with various scales and aspect ratios using a concept called anchor boxes.

As explained in [Ren et al., 2015], the RPN processes the image using the same convolutional layers used in the Fast R-CNN detection network. Thus, the RPN does not take extra time to produce the proposals compared to the algorithms like Selective Search. Due to sharing the same convolutional layers, the RPN and the Fast R-CNN can be merged/unified into a single network. Thus, training is done only once.

[Figure 24] depicts the workflow of faster RCNN,

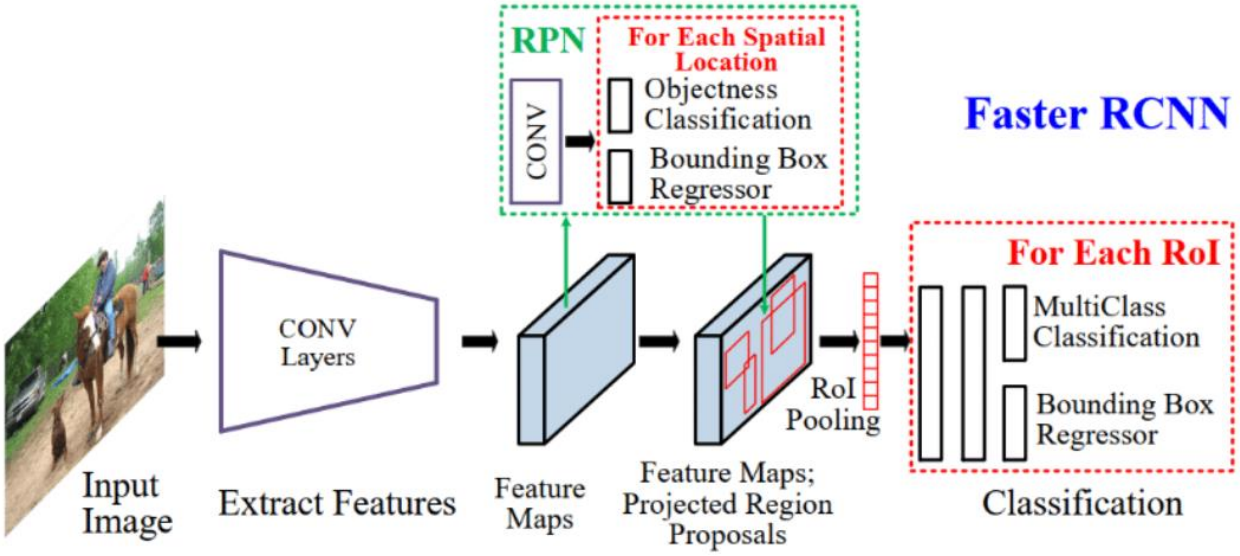


Figure 24 Workflow of Faster RCNN [Ghoury et al., 2019]

The Convolutional (CONV) layers of the network are shared between RPN and Fast RCNN. The final output of the last CONV layer is a feature map and is the input to RPN. It uses a sliding window approach to provide multiple ROIs which are then passed through two FC layers. Each ROI proposal is parametrized according to a reference box which is called an **anchor box**. The two parameters of the anchor boxes are Scale and Aspect Ratio. The default values used in the thesis are 3 scales and 3 aspect ratios, hence there are 9 anchor boxes per region proposal. Thus, a single image at a single scale is used while being able to offer scale-invariant object detectors, as the anchors exist at different scales. This avoids using multiple images or filters. The multi-scale anchors are key to share features across the RPN and the Fast R-CNN detection network. The first FC layer is named cls and represents a binary classifier that generates the **objectness score** for each region proposal (i.e., whether the region contains an object, or is part of the background). The second FC layer is named reg which returns a 4-Dimensional(4-D) vector defining the bounding box of the region proposal. These are provided as additional input to the Fast RCNN module along with the feature map from the last CONV layer. The ROI pooling layer in the Fast RCNN module takes over the feature extraction and the two FC layers of Fast RCNN module provide the class of the object detected and the bounding box 4-D vector the detected object.

5.4 METRICS:

The performance of the object detection models can be evaluated using a number of metrics, but the two important ones used in general are

- Intersection over Union (IOU)
- Mean Average Precision (mAP)

5.41 INTERSECTION OVER UNION:

Intersection over Union is computed for each predicted bounding box. The area of overlap between the predicted bounding box and its corresponding ground truth annotation divided by the area of union of these two bounding boxes gives the IOU. This can be easily understood from [Figure 25]. The higher the IOU, the better the prediction fits the ground truth annotation and hence better the performance of the model.

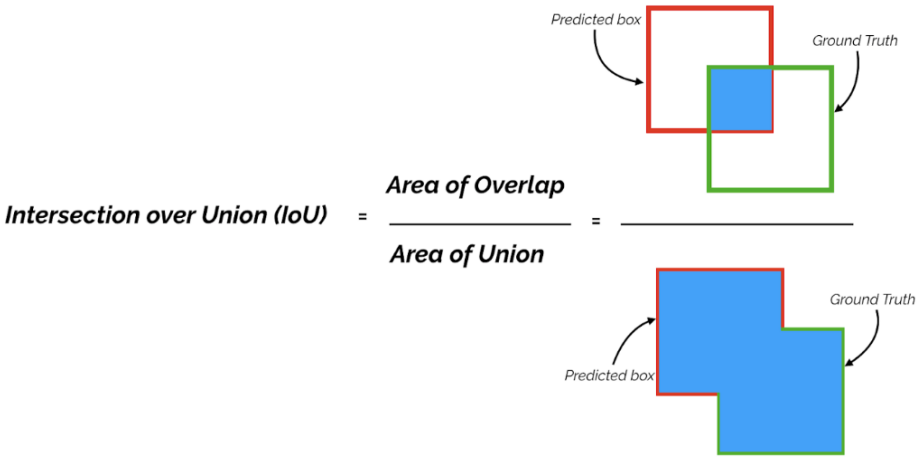


Figure 25 A figure depicting Intersection over Union [Padilla et al., 2020]

5.42 MEAN AVERAGE PRECISION:

Mean average precision is computed per image tile and has the following steps.

- Compute the number of True Positives (TP), False Positives (FP) and the False negatives (FN)
- A prediction is called true positive or false positive based on the IOU threshold. For example, if the IOU threshold is set to 0.5, then all the predictions with IOU >= 0.5 are classified as TP or else as FP.
- Compute Precision and recall using the equations Eq 1,Eq 2 .

$$Precision = \frac{TP}{TP+FP} \quad \text{Eq 1}$$

$$Recall = \frac{TP}{TP+FN} \quad \text{Eq 2}$$

- The coco_mAP is used as the standard which computes mAP [0.5_1] which is the average of mAP over 10 IOU thresholds from 0.5 to 1 in successive steps of 0.05. Hence the precision recall pairs for 11 different IOU thresholds (0.5 to 1 in a step size of 0.05) are computed,
- Plot the 11-point interpolated precision recall curve as in [Figure 26].

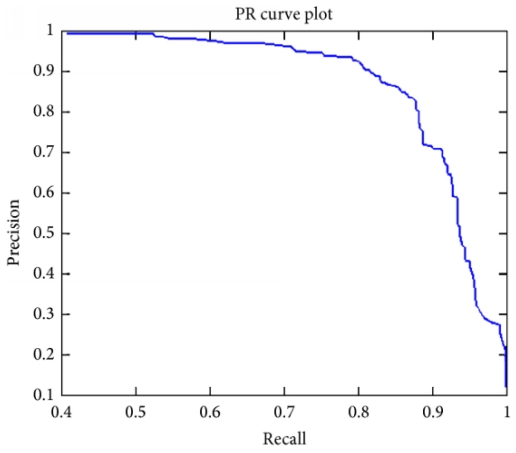


Figure 26 Sample Precision Recall Curve

- Compute the area under the curve for Average Precision (AP).
- If there are multiple classes, mean of AP over all the classes provides the Mean Average Precision(mAP)
- In our case, we have only one class(building) and hence average precision already gives the mAP.

Higher the mAP of an image, better is the model performance. Since test splits of datasets consist of multiple images, the average of mAP across the test split is used to quantify the performance of the split.

5.5 EXPERIMENTAL SETUP:

The first approach is to use a pretrained Faster RCNN network with the Resnet50 backbone. This model would be used as a baseline model from which it can be further improved. This improvement is measured by using the mAP metric.

The datasets are quite large but the resource available is an 8 Giga Bytes (GB) Graphics processing unit (GPU) which is not sufficient to train the entire dataset. Hence as a solution to this problem, a small fraction of the dataset was chosen which was in a supportable size for the GPU.

For sampling, the training and validation JSON files with the tile information, were divided into smaller splits each of size 2500 and one of the splits were randomly chosen. For BKG Dataset, the selected training split was 2500 sample tiles in size which was 5% of the total number of image tiles in the provided training split. In case of xView dataset, the chosen training split was 2500 sample tiles in size which was 10% of the total number of image tiles provided in its training split. A flow chart depicting the summary of the experimental setup which describes the research question answered in each experiment along with the procedure for the experiment can be seen in [Figure 27] and technical setup information is provided in [Table 1]

Table 1 Information on Technical Setup

TECHNICAL SETUP	
Programming language	Python version 3.7
ML Framework used	PyTorch version 10.2
Integrated Development Environment (IDE)	PyCharm version 2021.2
Geospatial software used	QGIS version 3.1

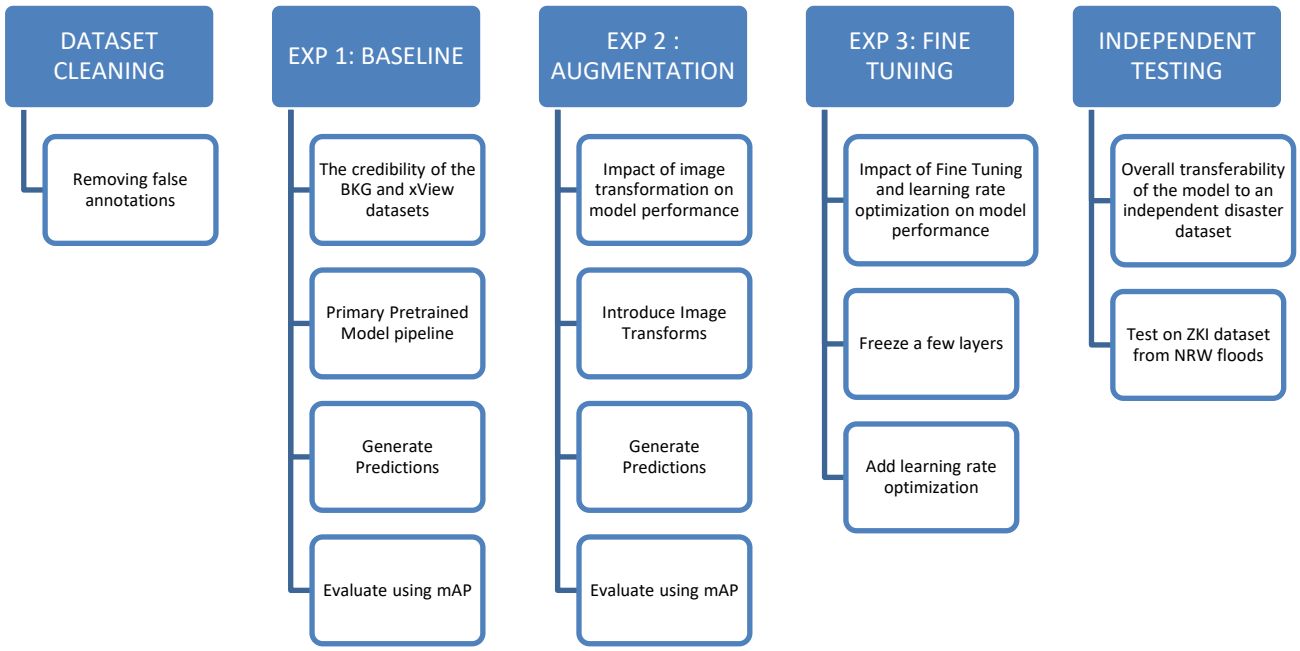


Figure 27 A Summary Workflow for experimental setup mentioning the research question answered in each experiment along with the key steps for each experiment.

5.6 PREDICTIONS:

To generate predictions, the saved models from their respective training phases are loaded again and the bounding box predictions for the image tiles in test split are generated as results. The idea is to train and test on independent test splits of the same datasets to see the model performance in a generic case. To evaluate how transferable the model is, predictions are generated for BKG test split on xView model weights and vice versa. These bounding box predictions are plotted by using OpenCV 2 (CV2) and Matplotlib modules in Python on the image tiles for visualization. The predictions are also saved as GEOJSON files for visualization on geospatial software like ArcGIS and QGIS. The generated predictions and the ground truth are used to compute the metrics Intersection over Union (IOU) for each predicted bounding box and Mean Average Precision(mAP) for the entire test split. The results which comprise these computed metrics and the visualizations will be presented in [Chapter 6].

5.7 EXPERIMENT 1: TILE SIZE INFLUENCE:

The influence of tile size on the performance of the models has been analyzed. The experiments were carried out for the baseline model in two image tile sizes namely 1024 x 1024 and 512 X 512 pixels and the results were documented. This experiment helps in understanding the impact of tile size on the mAP. This experiment helps us choose the right tile size for the rest of the experiments.

5.8 EXPERIMENT 2: BASELINE MODEL:

Research Question: where does the performance of the RCNN architecture in the form of a pretrained network on custom satellite and aerial imagery datasets stand?

The baseline model weights are obtained by training the Faster RCNN network with a ResNet-50 backbone on the BKG dataset and the xView datasets separately. This network is provided by the PyTorch platform. and is pretrained on the COCO dataset. Hence the model weights were initialized to the pretrained COCO weights. ResNet-50 backbone variant has 50 layers as indicated by the name. The training pipeline includes an early stopping module which stops the training and saves the weights for the epoch with the least validation loss. The patience parameter decides the number of epochs to wait before ending the training if the validation loss ceases to increase. The training and validation losses for Faster RCNN architecture is a summation of 4 losses which are RPN classification (Object foreground/background), RPN regression (Anchor → ROI), Fast RCNN Classification (object classes) and Fast RCNN Regression (ROI → Bounding Box) losses which is the output of the model in the training phase. A uniform value of 14 epochs has been used as the patience for the early stopping counter for all datasets.

BKG TRAINING:

- **Dataset:** BKG
- **Tile size:** 512 X 512 pixels
- **Architecture**: Fasterrcnn_resnet50
- **Learning Rate**: 0.00001
- **Optimizer**: Adam
- **No of Training Samples:** 2500
- **No of Validation Samples**: 2510
- **No of Test Samples**: 15400

The learning curve for training and validation on BKG dataset is seen in [Figure 28].

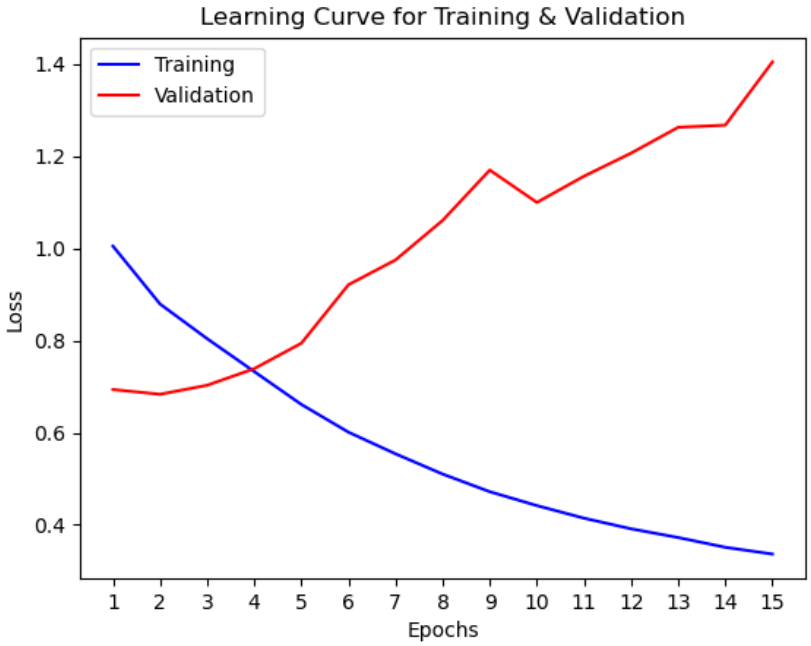


Figure 28 Learning Curve for BKG Dataset

The validation loss continues to increase and hence the training process is terminated at 15th epoch by the early stopping module.

XVIEW TRAINING:

- **Dataset:** XVIEW
- **Tile size:** 512 X 512 pixels
- **Architecture**: Fasterrcnn_resnet50
- **Learning Rate**: 0.00001
- **Optimizer**: Adam
- **No of Training Samples:** 2500
- **No of Validation Samples**: 2500
- **No of Test Samples**: 998

The learning curve for training and validation on xView dataset is seen in [Figure 29].

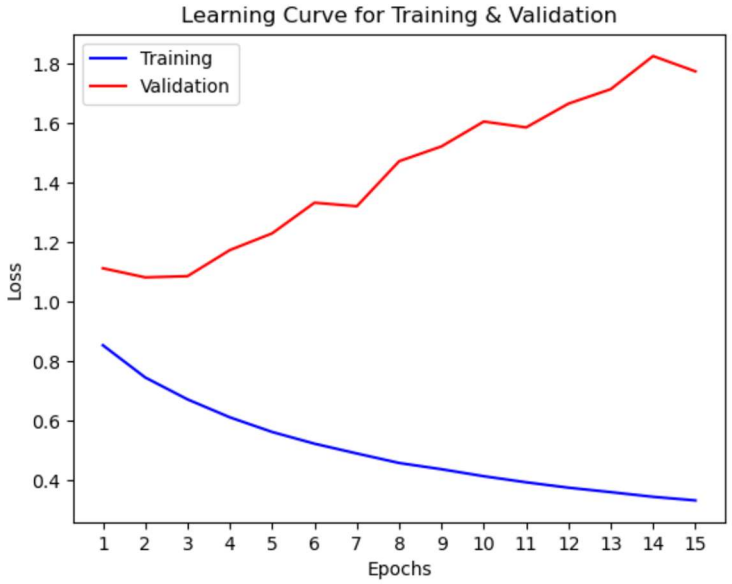


Figure 29 Learning Curve for xView Dataset Training

The validation loss in this case also ceases to decrease and continues to increase and hence the training gets terminated at 15th epoch. The model weights for the epoch with the lowest validation loss gets saved to be used for generating predictions.

5.9 EXPERIMENT 3: AUGMENTED MODEL:

Research Question: How does transforms induced on the imagery in the dataset affect the performance of a CNN model? Does it improve or deteriorate the performance of the existing model weights?

Once the mAP for the BKG dataset weights and the xView dataset weights are computed for their respective test splits, we infer where the model stands when compared to state-of-the-art mAP value for object detection. To improve the performance of the model, the first method implemented was Data Augmentation.

DATA AUGMENTATION:

Data augmentation is the process of increasing the versatility of the dataset by introducing a few image transforms to the image tiles. This increases the sample set size as well as changes the orientation and contrast of the contents of the image. Thus, when trained on this augmented dataset, the intuition is that the model learns better and improves the performance.

TRANSFORMS USED:

For applying transforms, the Albumentations module in Python has been used. The following transforms have been stitched together and applied to every image tile to create a transformed one.

RandomRotate90

This transform randomly rotate the input by 90 degrees zero or more times.

RandomScale

This transform randomly resizes the input such that the output image size is different from the input image size. It has a scaling factor range parameter that controls the amount of scaling and an interpolation flag that is used to specify the interpolation algorithm.

RandomCrop

This transform crops a random part of the image. It is associated with a height and width parameter that specifies the height and width for the crop.

Flip

This transform is used to flip the image horizontally or vertically.

RandomBrightnessContrast

This transform is used to randomly change brightness and contrast of the input image. It has the parameter brightness_limit to factor range for changing brightness, contrast_limit to factor range for changing contrast and a brightness_by_max fag which when set to true adjusts contrast by image data type maximum, else adjust contrast by image mean.

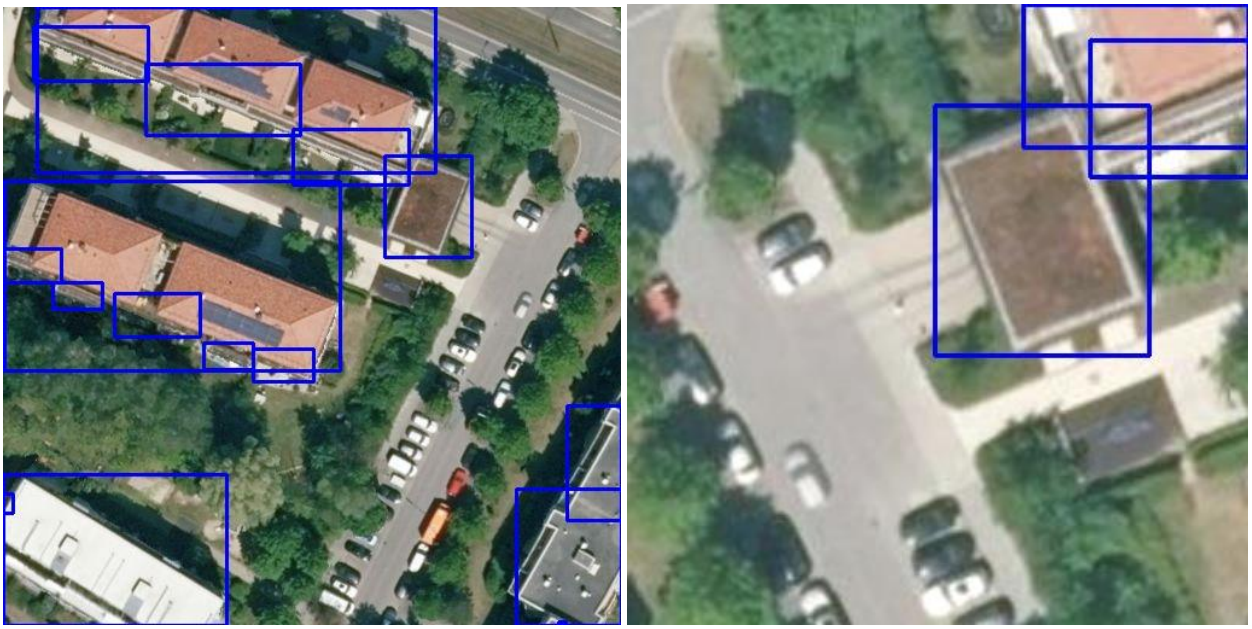
All the above transforms are also associated with a probability parameter p that decides probability of applying the transform which has a default value of 0.5. Below are a few image tiles of BKG dataset [Figure 30][Figure 31][Figure 32] and xView dataset [Figure 33][Figure 34][Figure 35] before and after the transforms. All the below figures are of scale 512 X 512. The ink blue annotations (bounding boxes) belong to the ground truth labels.

(a) *(b)*

Figure 30 Image tile dop20_rgb_32695_5339_1_r5_c6 (a) before augmentation (b) after augmentation. Ground Truth bounding boxes: Ink Blue annotations Scale: (102.4m X 102.4m)

(a) *(b)*

Figure 31 Image tile _aug_dop20_rgb_32695_5339_1_r6_c6 (a) before augmentation (b) after augmentation. Ground Truth bounding boxes: Ink Blue annotations Scale: (102.4m X 102.4m)

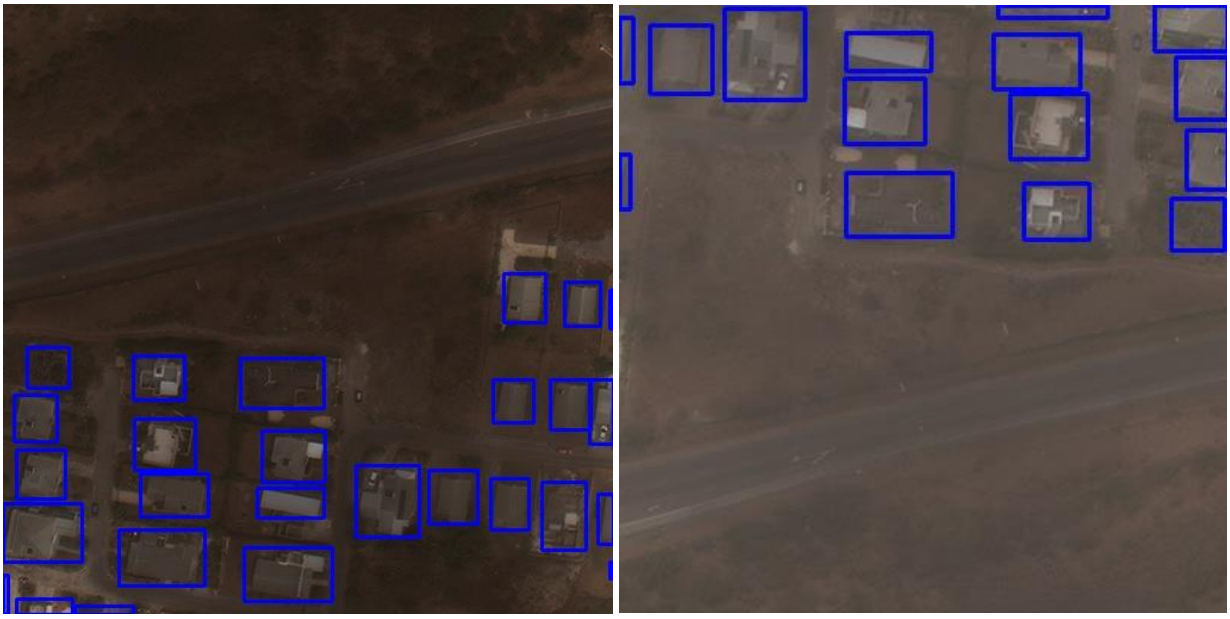


(a) *(b)*

Figure 32 Image tile dop20_rgb_32695_5339_1_r8_c7 (a) before augmentation (b) after augmentation. Ground Truth bounding boxes: Ink Blue annotations Scale: (102.4m X 102.4m)

(a) *(b)*

Figure 33 Image tile 1482_col_4_row_3 (a) before augmentation (b) after augmentation. Ground Truth bounding boxes: Ink Blue annotations Scale: (153.6m X 153.6m)



(a) *(b)*

Figure 34 Image tile 1482_col_4_row_1 (a) before augmentation (b) after augmentation. Ground Truth bounding boxes: Ink Blue annotations Scale: (153.6m X 153.6m)

(a) *(b)*

Figure 35 Image tile 1482_col_0_row_5 (a) before augmentation (b) after augmentation. Ground Truth bounding boxes: Ink Blue annotations Scale: (153.6m X 153.6m)

After having applied these transforms the training split and validation split of the dataset, they double in size as they contain the augmented data as well. Now the training process is repeated, with the new augmented dataset. The learning curves for the augmented BKG and augmented xView datasets can be seen in [Figure 36] and [Figure 37] respectively.

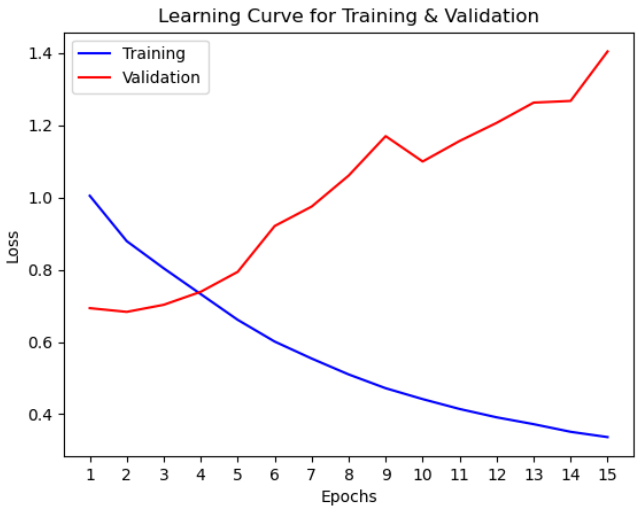


Figure 36 Learning Curve for BKG Augmented Dataset

The least validation loss 0.573 has decreased by 0.12 compared to the least validation loss of the baseline model which is 0.693.

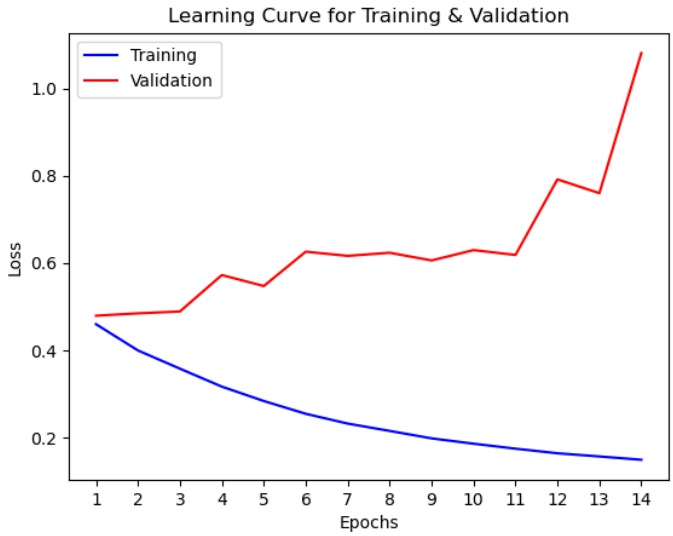


Figure 37 Learning Curve for xView Augmented Dataset

The least validation loss is 0.479 which is a huge decline from 0.596 of the baseline model for xView dataset.

5.10 EXPERIMENT 4: FINETUNED MODEL:

Research Question: What is the impact of fine tuning an RCNN model for object detection? Does it improve or deteriorate the performance of the existing model weights?

After having generated predictions on the augmented model and computing the mAP for the test splits, a further improved model was implemented using finetuning approach. In this approach, 2 changes were made in the training process.

1. The first 6 layers or three (conv + ReLu) layers of the four convolution and identity blocks of resnet50 backbone was frozen. This stops backpropagation through these layers.
2. A learning rate optimizer was used that keeps decreasing the learning rate after each epoch if the validation loss does not decrease.

After having implemented these changes the model weights were obtained by training again on the augmented dataset.

The learning curve obtained on BKG dataset is in [Figure 38] and the learning curve for finetuned model on xView dataset can be observed in [Figure 39].

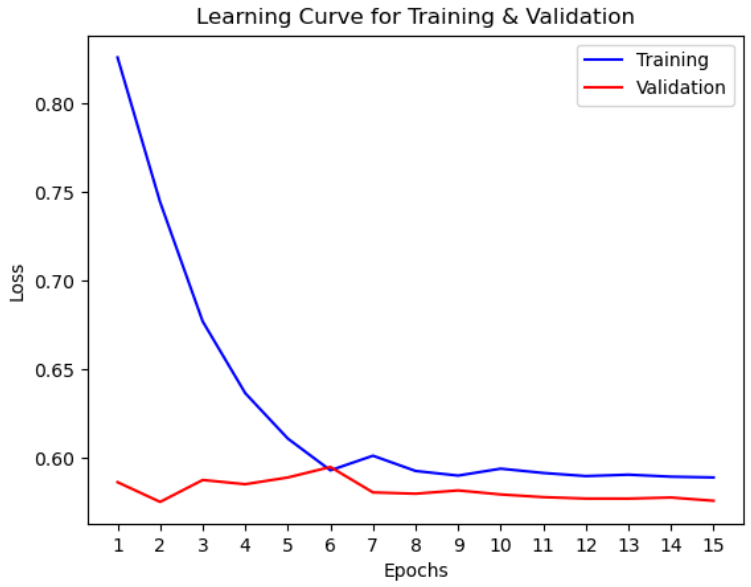


Figure 38 Learning curve for finetuned model on BKG dataset

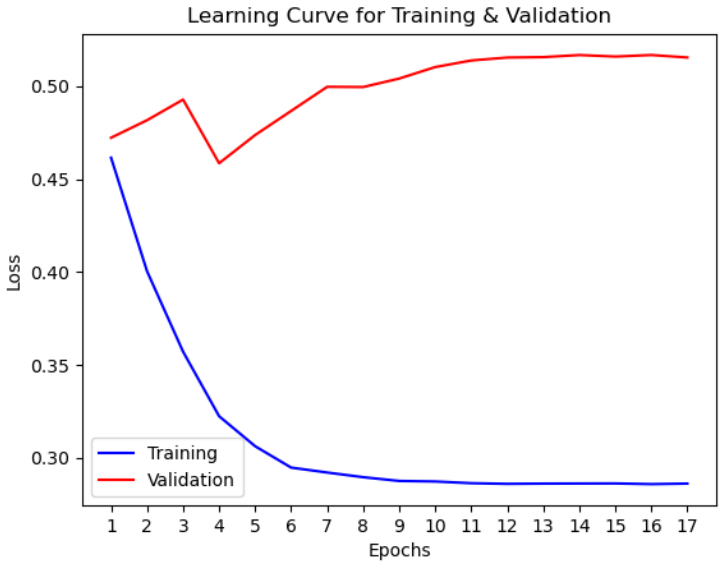


Figure 39 Learning curve for finetuned model on xView dataset

For BKG model, the validation loss is below the training loss but has further increased from the augmented model to 0.575 from 0.573 and for xView, the validation loss has decreased to 0.472 from 0.479 which is also a decrease from its corresponding augmented model.

5.11 EXPERIMENT 5: INDEPENDENT TEST SET:

Research Question: What is the overall transferability of the model to a dataset (with different resolution and acquisition geometry), which it has never seen before?

To test the transferability of the models in a completely independent setup, the ZKI dataset is used as an independent test set. The model has not seen this dataset acquired during the North Rhine Westphalia (NRW) floods. Both the BKG weights and xView weights are tested with this dataset to see which one performs better in a transferability scenario by evaluating using the mAP metric.

5.12 GEOSPATIAL VISUALIZATION ON A LARGER SCALE

The generated predictions are saved as GEOJSON files which are files that store these predictions converted and represented as simple geographical features such as lines and polygons to be visualized in different coordinate reference systems using a geographic information system application software like QGIS. The BKG and ZKI dataset predictions have been saved as multipolygons in EPSG:25832 spatial reference system. The xView dataset predictions are saved as polygons in EPSG: CRS84 coordinate reference system. All the

predictions for one image tile is saved as features in one GeoJSON. The predictions saved as GeoJSON files visualized in QGIS on the entire scene(comprising of multiple tiles) for BKG dataset can be seen in [Figure 40], [Figure 41] and for xView dataset can be seen in [Figure 4242], [Figure 4343].

Figure 40 Visualization of predictions as GEOJSON features of the scene dop20_rgb_32794_5824_1 of BKG dataset. (Each polygon represents a different building. The colours do not indicate any key difference)

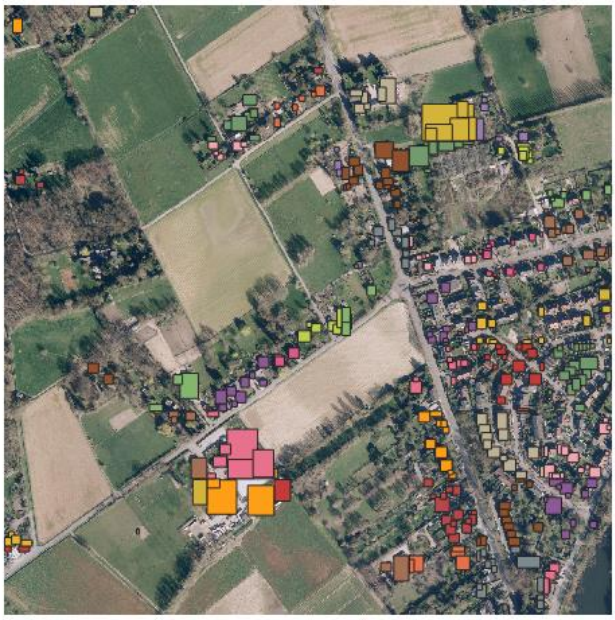


Figure 41 Visualization of predictions as GEOJSON features of the scene dop20_rgb_32330_5693_1 of BKG dataset. (Each polygon represents a different building. The colours do not indicate any key difference)

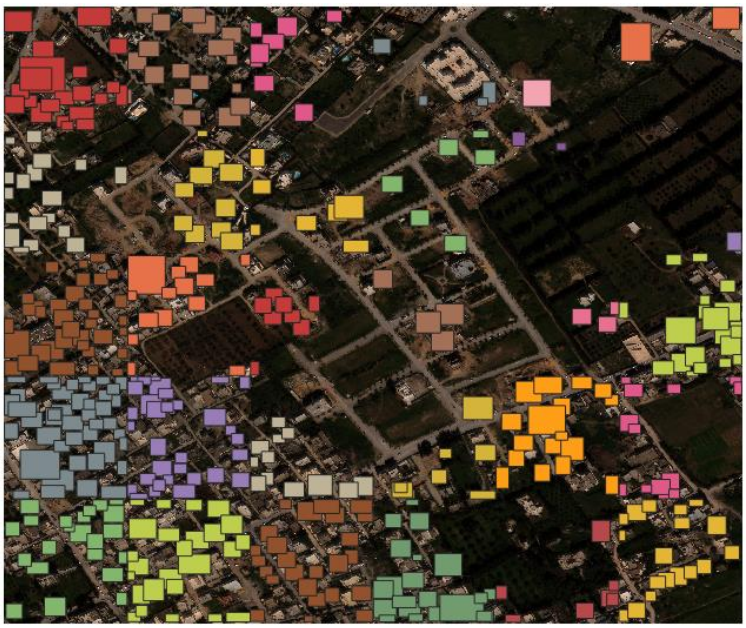


Figure 42 Visualization of predictions as GEOJSON features of the scene 104 of xView dataset. (Each polygon represents a different building. The colours do not indicate any key difference)

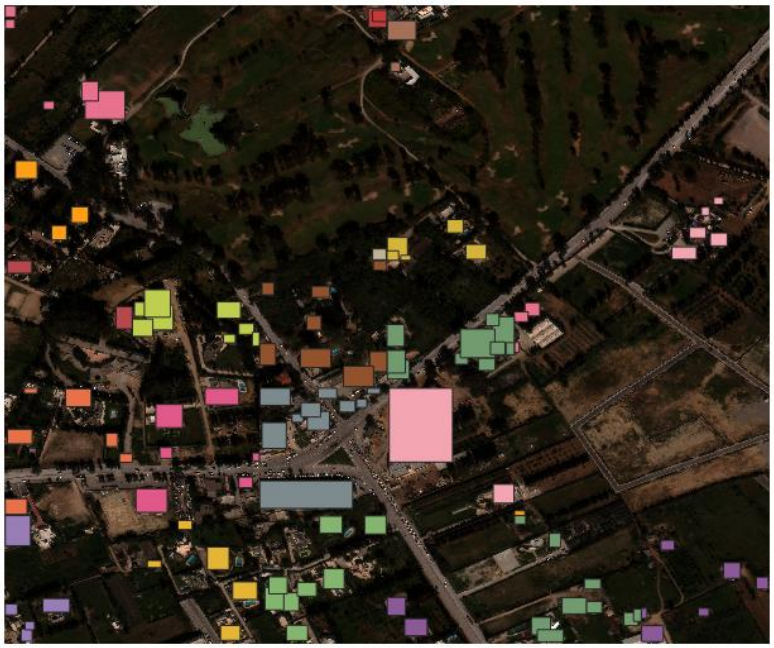


Figure 43 Visualization of predictions as GEOJSON features of the scene 106 of xView dataset. (Each polygon represents a different building. The colours do not indicate any key difference)

6. RESULTS

This chapter presents the results of the four experiments explained in chapter 5. It starts with the results for the influence of tile size experiment, that decides the suitable tile size for images to be used in all the 4 experiments. The performance of the 3 models (baseline, augmented and finetuned) from the first 3 experiments across the 4 scenarios explained below is grouped and presented in the form of a table. A chart [Chart 1] is added for better visualization of the table that helps in simplifying the task of analyzing the underlying patterns and identification of best models. The second half of the chapter consists of generated bounding box predictions and ground truth bounding boxes printed on the image tiles for all the 3 models across the 4 explained scenarios. This aids in understanding the performance of the models that is quantified by IOU and mAP metrics in [Table 3].

6.1 EXPERIMENT 1: INFLUENCE OF TILE SIZE ON PERFORMANCE:

The image tiles for both BKG and xView datasets are available in two tile sizes namely 1024 X 1024 and 512 X 512 pixels. An experiment on which tile size would better suit the object detection models and provide a better performance was conducted and the results have been summarized in [Table 2].

Table 2 Influence of Tile Size on mAP

TILE SIZE (Pixels)	TRAIN	TEST	mAP
512 x 512	XVIEW	XVIEW	**0.2637**
1024 x 1024	XVIEW	XVIEW	0.1913
512 x 512	BKG	BKG	**0.3685**
1024 x 1024	BKG	BKG	0.2414

The inference is that there is a steep increase in performance of the models when image tiles of size 512 X 512 pixels were used. So, all the below experimental results were obtained for models trained and tested on image tiles of size 512 X 512 pixels.

6.2 PERFORMANCE EVALUATION EXPLANATION:

The evaluation of the performance of the 3 models (experiment 1, experiment 2, experiment 3) is done in four different scenarios. The resultant model from

- experiment 1 is addressed as Baseline model,
- experiment 2 is addressed as Augmented model and
- experiment 3 is addressed as FT & LR Opt (Finetuned & Learning)

SCENARIOS:

GENERIC:

1) Training on BKG and testing on BKG which essentially uses the BKG model weights obtained during its training, on BKG training split to test on the BKG test split.
2) Training on xView and testing on xView which essentially uses the xView model weights obtained during its training, on BKG training split to test on the xView test split.

FLEXIBILITY:

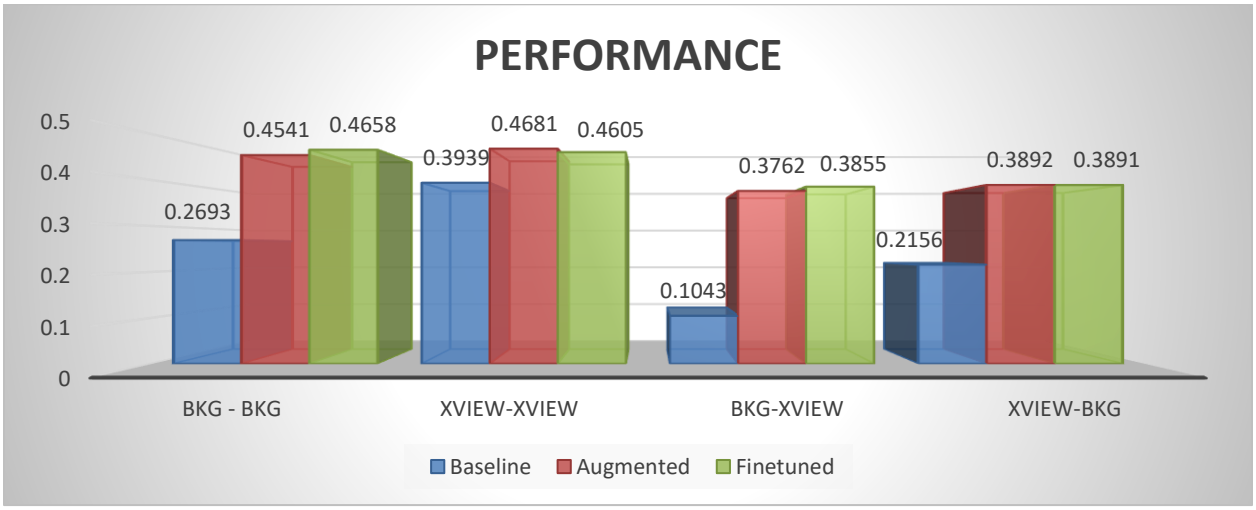
3) Training on BKG and testing on xView which essentially uses the BKG model weights obtained during its training to test on the xView test split
4) Training on xView and testing on BKG which essentially uses the xView model weights obtained during its training to test on the BKG test split

The first two scenarios help us infer the performance of the models in a general scenario while the next two scenarios help us evaluate the transferability and flexibility of the models. In each scenario evaluation has been performed on three different models namely Baseline, Augmented and FT & LR Opt (Finetuned & Learning Rate Optimized) as explained in chapter 4 and the results are summarized in [Table 3] and [Chart 1].

Table 3 Documentation of Performance of Models in 4 Scenarios

SCENARIOS	DATASETS		MODELS (EVALUATED IN mAP)		
	TRAIN	TEST	EXP 1 (BASELINE)	EXP 2 (AUGMENTED)	EXP 3(FT & LR OPT)
GENERIC (SCENARIOS 1 & 2)					
1	**BKG**	**BKG**	0.2693	0.4541	**0.4658**
2	**XVIEW**	**XVIEW**	0.3939	0.4681	0.4605
TRANSFERABILITY (SCENARIOS 3 & 4)					
3	**BKG**	**XVIEW**	0.1043	0.3762	0.3855
4	**XVIEW**	**BKG**	0.2156	**0.3892**	0.3891

Table 4 (Chart 1) A Column chart that represents Table 1 for better visualization



6.3 RESULTS FOR EXPERIMENTS 2,3,4 GROUPED BY SCENARIOS:

The inferences in [Table 3] and [Chart 1] are made based on the mAP obtained on the test splits of the three datasets. The prediction results on the image tiles have also been visualized to give us an understanding of the performance. The distinct improvement in performance between baseline, augmented and finetuned models can be observed in the following images. In the following images, the ink blue bounding boxes indicate the ground truth, and the cyan blue bounding boxes indicate a model prediction. The improvement in performance can be observed as missed predictions being identified, wrong predictions being eliminated or as an increase in the IOU metric of the same predicted bounding box between the compared models.

6.31 TRAINED & TESTED ON BKG:

In the first scenario, the models are trained on the training split and tested on the testing split of the same dataset BKG. We can observe that for the baseline model the mAP is 0.2693. When augmentation technique is applied, the mAP increases drastically to 0.4541 which is a significant increase. When the finetuning technique is applied, the performance still increases but not very steeply to 0.4658.

(a) *(b)* *(c)*

Figure 44 The predictions on dop20_rgb_32330_5693_1_r3_c7 for (a) Baseline model weights, (b) Augmented model weights and (c) Finetuned Model. Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (102.4m X 102.4m)

In [Figure 44], we can observe that the the baseline model has many predictions that are missing as we can observe a lot of ink blue bounding boxes but no cyan blue bounding boxes. But in the augmented model prediction, a few more cyan blue bounding boxes can be observed which

were missed by the baseline model with IOUs 0.67, 0.87 and 0.8 respectively. In the finetuned model, one bounding box missed by the augmented model seems to be identified with IOU 0.6 in the bottom right corner.

(b) *(b)* *(c)*

Figure 45 The predictions on dop20_rgb_32330_5693_1_r4_c4 for (a) Baseline model weights, (b) Augmented model weights and (c) Finetuned Model. Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (102.4m X 102.4m)

In [Figure 45], the baseline model has two missing predictions in the bottom rightside. But in the augmented model prediction, these two missed predictions are identified with IOUs 0.64 and 0.74 respectively. In the finetuned model, a building missed by augmented model is predicted with IOU 0.87, the IOU values of the predictions are comparatively better than the ones in augmented model's.

6.32 TRAINED & TESTED ON xVIEW:

In this scenario, the three models are trained on the training split and tested on the testing split of the same dataset xView. The baseline model starts with an mAP of 0.3939 which is greater than 0.2693 mAP of the BKG dataset's baseline model when trained and tested on it as discussed in the first scenario. When augmentation technique is introduced, the mAP increases steeply from 0.3939 to 0.4681. But when the finetuning approach is introduced, the mAP drops not very steeply but to 0.4605.

(a) *(b)* *(c)*

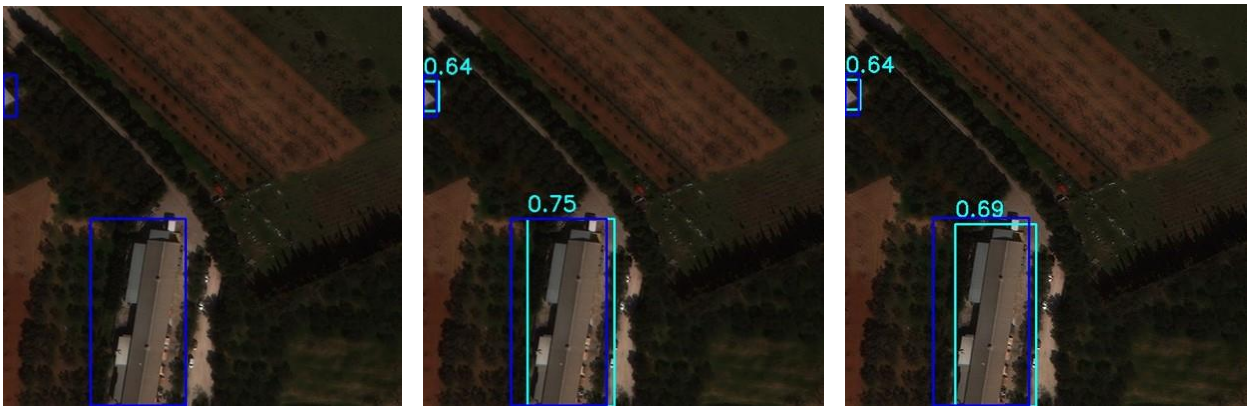
Figure 46 The predictions on 105_col_1_row_0 for (a) Baseline model weights, (b) Augmented model weights and (c) Finetuned Model. Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (153.6m X 153.6m)

In [Figure 46], a prediction missed by baseline model gets identified by augmented model with an IOU of 0.64. But when finetuned, building predicted by augmented model with an IOU of 0.77 is missed. Also multiple predictions for the large building with IOU 0.89 emerges.

(a) *(b)* *(c)*

Figure 47 The predictions on 180_col_2_row_1 for (a) Baseline model weights, (b) Augmented model weights and (c) Finetuned Model. Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (153.6m X 153.6m)

In [Figure 47], we can clearly observe that the missing prediction for the biggest building gets identified in the augmented model. The finetuned model performs almost to the same level but a few predictions in augmented model have a better IOU.



(a) *(b)* *(c)*

Figure 48 The predictions on 140_col_6_row_0 for (a) Baseline model weights, (b) Augmented model weights and (c) Finetuned Model. Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (153.6m X 153.6m)

In [Figure 48], the baseline model does not predict any of the two buildings. The augmented model predicts both of the buildings with an IOU of 0.64 and 0.75. The finetuned model also predicts both the buildings correctly but the IOU value for the larger building is better for the augmented model.

In the next two scenarios, we essentially evaluate how transferable the models are to a dataset with completely different resolution, regions of interest and sensor for acquisition and acquisition conditions.

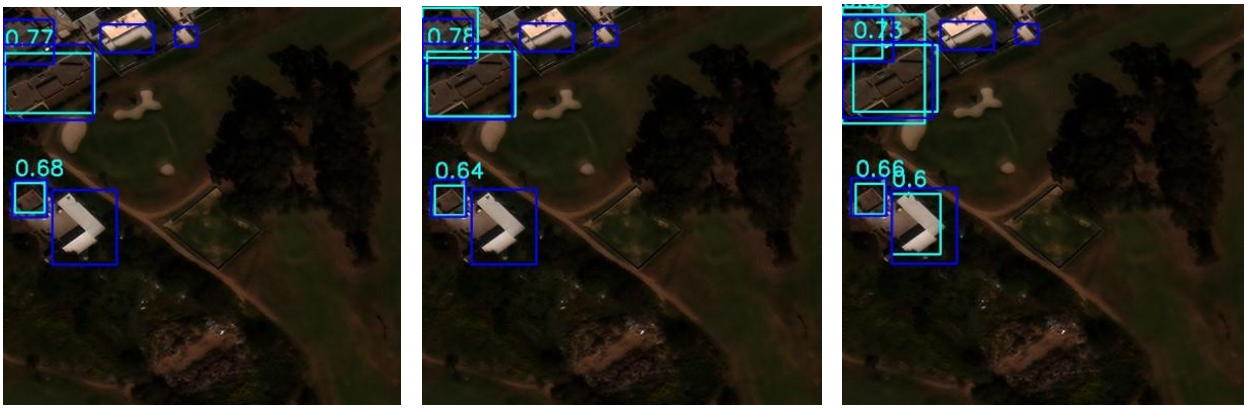
6.33 TRAINED ON BKG & TESTED ON xVIEW:

This scenario essentially evaluates the credibility of the models trained on BKG dataset when subjected to flexibility and transferability. The three models trained on BKG dataset which is restricted to Image tiles in Germany will be tested on xView dataset which has imagery from across the world, acquired with a different sensor and a slightly worse resolution. The baseline model starts with a very low mAP which is 0.1043. But when augmentation is applied, the model performance improves drastically to 0.3762. The finetuning approach also seems to improve the transferability slightly to 0.3855.

(a) *(b)* *(c)*

Figure 49 The predictions on 104_col_4_row_2 for (a) Baseline model weights, (b) Augmented model weights and (c) Finetuned Model. Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (153.6m X 153.6m)

In [Figure 49], the baseline model (a) predicts 3 out of the 6 buildings. The augmented model (b) makes one more prediction with an IOU of 0.65 thus visibly performing better than baseline model. The finetuned model (c) predicts one more building in the bottom with an IOU of 0.63 totally predicting 5 out of 6 buildings thus distinctly showing a better performance than the other two models.



(a) *(b)* *(c)*

Figure 50 The predictions on 106_col_3_row_0 for (a) Baseline model weights, (b) Augmented model weights and (c) Finetuned Model. Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (153.6m X 153.6m)

In [Figure 50], (a) predicts two of the six buildings with a decent IOU whereas (b) predicts one more building when compared to (a). The building missed by both (a) and (b) is predicted by (c) totally predicting four of the six buildings outperforming the other two models.

(a) *(b)* *(c)*

Figure 51 The predictions on 106_col_0_row_1 for (a) Baseline model weights, (b) Augmented model weights and (c) Finetuned Model. Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (153.6m X 153.6m)

In [Figure 51], the baseline model makes three decent predictions. The augmented model (b) predicts one more building on the left with an IOU of 0.75. But the finetuned model (c) predicts all the buildings predicted by the augmented model but with a better IOU and also an additional building in the bottom left corner with an IOU of 0.67.

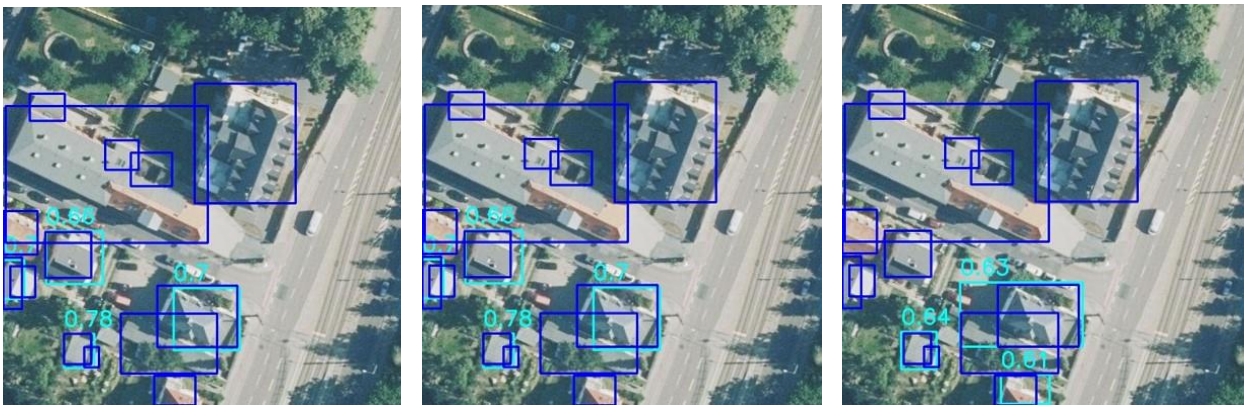
6.34 TRAINED ON xVIEW & TESTED ON BKG:

This is a scenario which essentially evaluates the models trained on xView dataset when subjected to transferability across datasets. In this case a model trained on xView dataset which has imagery from across the globe will be tested on BKG dataset with a slightly better resolution and restricted to images from Germany. The baseline model starts with a low mAP which is 0.2156 but increases to 0.3892 when augmentation is applied to it. But when finetuning is applied the model does not perform any better as the mAP is at the same value or can be technically considered very slightly decreasing by 0.0001 as it amounts to 0.3891.

(a) *(b)* *(c)*

Figure 52 The predictions on dop20_rgb_32622_5633_1_r7_c3 for (a) Baseline model weights, (b) Augmented model weights and (c) Finetuned Model. Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (102.4m X 102.4m)

In [Figure 52], the augmented model (b) does perform better by identifying many missing buildings in the predictions generated by baseline model (a). The finetuned model (c) seems to iden.tify the one prediction missed by (b) in the bottom.



(a) *(b)* *(c)*

Figure 53 The predictions on dop20_rgb_32834_5672_1_r2_c0 for (a) Baseline model weights, (b) Augmented model weights and (c) Finetuned Model. Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (102.4m X 102.4m)

In [Figure 53], the augmented model (b) predicts the same number of predictions as the baseline model but the finetuned model (c) makes only 3 predictions and performs not as good as the augmented model (b).

These images justify the observations made from the average mAP values on the test splits and presented in [Table 3] and [Chart 1] do hold well as for models trained on BKG the

finetuned model (c) performs better and for models trained on xView dataset, the augmented model outperforms the finetuned model in both the generic and transferability scenarios. In scenario 4, for models trained on xView and tested on BKG, both augmented and finetuned models perform the same way. But the overall performance is better for model trained on xView dataset.

6.4 EXPERIMENT 5 RESULTS AFTER TESTING ON INDEPENDENT TEST SET:

To confirm if our interpretations are valid, the above models are tested on the independent test set ZKI obtained during the NRW floods. This dataset has not been used in training of the models so that it can be used as an independent validation set. The best model when trained on BKG seems to be the finetuned model in both generic and transferability scenarios and the best model when trained on xView seems to be the augmented model. The predictions made by these two models on the same images of the ZKI dataset can be seen in [Figure 54],[Figure 55] and [Figure 56].

(a) (b)

Figure 54 The Finetuned BKG Model (a) and Augmented xView Model (b) on ZKI image 112. Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (61.44m X 61.44m)

(a) (b)

Figure 55 The Finetuned BKG Model (a) and Augmented xView Model (b) on ZKI image 383 Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (61.44m X 61.44m)

(a) (b)

Figure 56 The Finetuned BKG Model (a) and Augmented xView Model (b) on ZKI image 383 Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (61.44m X 61.44m)

Table 5 The BKG and xView Model performance on ZKI Dataset

WEIGHTS	TEST DATASET	CONFIDENCE	AUGMENTED	FINETUNED
BKG	ZKI	0.1	0.2939	0.2945
XVIEW	ZKI	0.1	0.2986	0.2986

[Table 5] summarizes that the model trained on xView has a better performance than the model trained on BKG which is justified by figures [Figure 54], [Figure 55] and [Figure 56]. [Figure 57] shows the geospatial visualization of all the generated predictions on ZKI dataset in QGIS for the scene dlr_luftbild_16_07_2021_rheinland_pfalz_1. The different colours do not represent any key difference.

Figure 57 Visualization of all the predictions as GEOJSON features of Ahrweiler. (Each polygon represents a different building. The colours do not indicate any key difference)

.

6.5 MISSING & MULTIPLE PREDICTIONS:

➢ The results chapter show that there are many missed predictions, where we observe an ink blue bounding box indicating the ground truth label and no corresponding cyan blue bounding box indicating a model prediction.

➢ The other distinct issue that can be noticed is multiple predictions for the same building instance. This can be observed as multiple cyan blue bounding boxes in place of a single ink blue bounding box. Multiple prediction issue is not as frequent as missed predictions.

This issue is caused by two parameters

- Prediction Score (PS) or Confidence Score,
- IOU Threshold

The confidence score and the correct IOU threshold varies from one data set to the other. The values of Confidence score and IOU thresholds have been computed by using mAP values as a measuring metric and plotted. The best mAP was observed for an IOU threshold of 0.6 in [Figure 58] for all three datasets. The best mAP is for a confidence score of 0.3 for xView and BKG datasets as observed in [Figure 59] and [Figure 60] and a value of 0.1 for ZKI in [Figure 61].

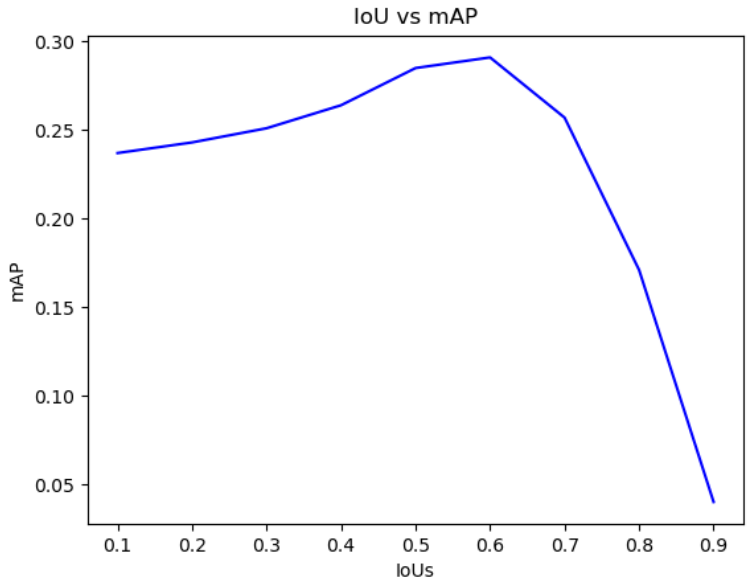


Figure 58 IOU vs mAP for BKG, xView and ZKI datasets

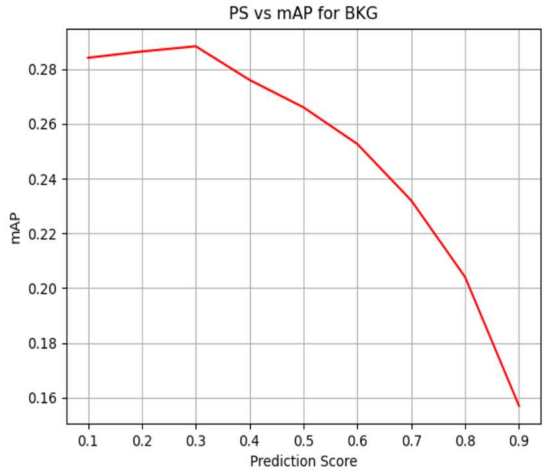


Figure 59 PS vs mAP for BKG Dataset

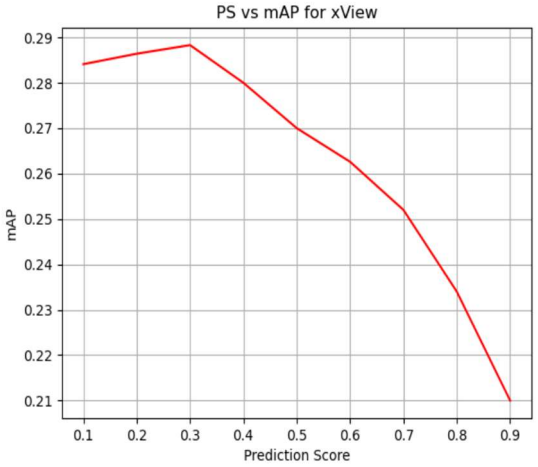


Figure 60 PS vs mAP for xView Dataset

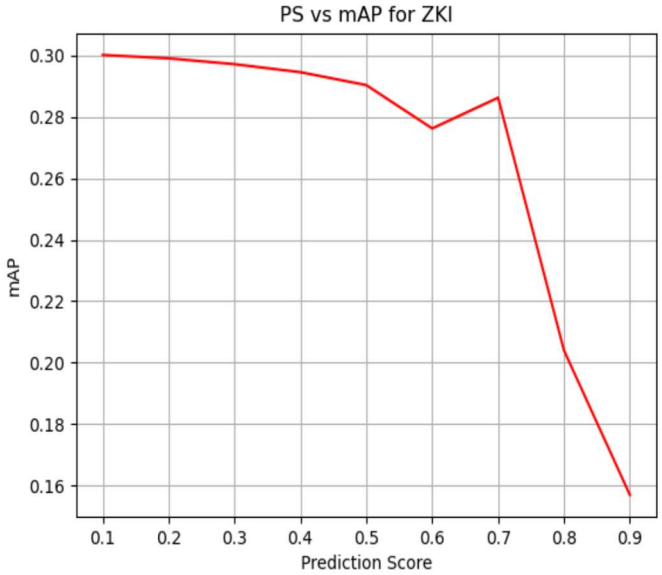


Figure 61 PS vs mAP for ZKI Dataset

The plots can be interpreted as lower the Prediction Score, lesser the number of missing predictions but higher the number of multiple predictions. Higher the IOU threshold, lesser the number of multiple predictions and higher the number of missing predictions. Hence the Prediction Score and IOU threshold has a tradeoff between them to provide the best performance as observed in [Figure 62] and [Figure 63].

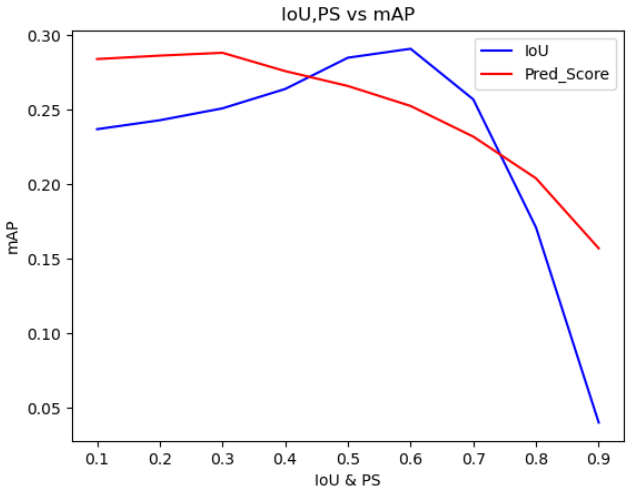


Figure 62 Tradeoff between PS and IOU for BKG, xView

The best value for PS and IOU is found to be 0.4 for BKG and xView datasets.

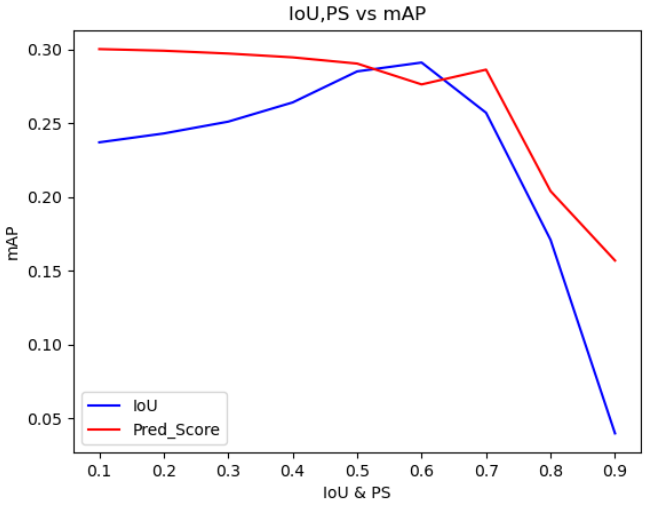


Figure 63 Tradeoff between PS and IOU for ZKI

The best value for PS and IOU is found to be 0.5 for ZKI dataset.

The predictions were generated again with these values and a significant reduction in the number of missing and multiple predictions can be observed.

(a) (b)

Figure 64 The Augmented Model before fixing tradeoff value (a) and after fixing tradeoff value (b) on xView image 106_col_0_row_1. Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (153.6m X 153.6m)

We can observe that in [Figure 64](b), a missed prediction in [Figure 64](a) by the best model on xView (augmented model) gets identified with IOU 0.54 after fixing PS and IOU to 0.4.

(a) (b)

Figure 65 The Finetuned Model before fixing tradeoff value (a) and after fixing tradeoff value (b) on BKG image dop20_rgb_32330_5693_1_r3_c7. Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (102.4m X 102.4m)

We can observe multiple missed predictions with the best model on BKG which is the finetuned model in [Figure 65] (a). Many of those missed predictions can be observed to be correctly identified by the same model after having rerun with a PS and IOU of 0.4 in [Figure 65] (b).

(a) (b)

Figure 66 The Augmented Model before fixing tradeoff value (a) and after fixing tradeoff value (b) on BKG image dop20_rgb_32834_5672_1_r2_c0. Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (102.4m X 102.4m)

We can observe that multiple missed predictions with the augmented model on BKG [Figure 66] (a), can be seen to be to be identified by the same model after having rerun with a PS and IOU of 0.4 in [Figure 66] (b).

(a) (b)

Figure 67 The Augmented Model before fixing tradeoff value (a) and after fixing tradeoff value (b) on ZKI image 476. Ground Truth bounding boxes: Ink Blue annotations, Prediction Bounding Boxes: Cyan Blue annotation, Scale: (61.44m X 61.44m)

We can observe multiple predictions for the same building with the xView augmented model on BKG [Figure 67] (a), which is observed to be replaced by a single prediction (0.66) by the same model after having rerun with a PS and IOU of 0.51 for ZKI dataset in[Figure 67](b).

7. DISCUSSION

Considering just the baseline models, the performance in the generic scenarios where we trained and tested on independent training and testing splits of the same reference data (same sensor) is better than the transferability scenarios where we train on one dataset and test on the other dataset (different sensors). This falls right on the intuition that the performance of a model when subjected to data not seen during the training phase is bound to decrease.

Again, when we focus on the baseline model across different scenarios, we can observe that the performance of the xView dataset is better than BKG quantitatively, in both the generic scenarios (0.3939 > 0.2693) and in the transferability scenarios (0.2156>0.1043).

The impact that is easiest to observe is of data augmentation in all the 4 scenarios. Augmentation improves the performance of all the 3 models in all the 4 scenarios. The improvement in performance from baseline to augmented model can be observed across the four scenarios in the decreasing order of Trained on BKG & Tested on xView, Trained & Tested on BKG, Trained on xView and Tested on BKG, Trained & Tested on xView.

When it comes to finetuned model, the observation is that models trained on BKG seem to benefit from the finetuning approach whereas for models trained on xView, the finetuned model seems to perform either on the same level as the augmented model or even slightly less robust when compared to augmented model. This can be seen as in Trained & Tested on BKG scenario and Trained on BKG & Tested on xView scenario, the finetuned model performance is quantifiably better than augmented model

- Trained & Tested on BKG → 0.4658 > 0.4541
- Trained on BKG & Tested on xView → 0.3855 > 0.3762

The finetuned model in both general and transferability scenarios wins over the others for BKG dataset. The augmented model wins over the other models in both generic and transferability scenarios for xView dataset. In both the generic and transferability scenarios, when a comparison between models trained on BKG and xView is done, the augmented model trained on xView seems to perform better and can be chosen as the ultimate winner.

The influence of tile size on performance is clearly evident from [Table 2]. A smaller tile size of 512 X 512 pixels has a steep increase in performance compared to the tile size 1024 X 1024 pixels irrespective of the kind of dataset. This is in line with the results of [Lee et al., 2022] where various tile sizes ranging between 500 X 500 to 1000 X 1000 pixels were used to

perform instance segmentation on MRI images and the tile size 500 X 500 pixels had the best performance.

Due to a restriction in the availability of GPU resource, a very small fraction of the totally available dataset was used for the entire experimental setup. The models trained on the complete train split of all the datasets might provide a better mAP value setting new benchmarks for object detection especially for transferability, but the pattern observed in performance between different scenarios might remain the same.

This finetuned model was generated by freezing the first six layers in each layer block of the resnet50 backbone. Since there is no hard rule on which layers should be frozen for a better performance, freezing more or lesser layers might be an interesting point to explore further in the finetuning area of the experiment.

A good transferable model could be developed if it is trained on many different datasets in a single epoch and hence fusing xView and BKG datasets together for training and testing them on ZKI dataset would also be an interesting focus area.

The core area of the thesis was on building detection. But since the labels for different object classes are available in these datasets, it would be interesting to explore the detection of other object classes like cars, in the event of a natural calamity that might help locate them easily. It would also be interesting to move on to testing the model on imagery acquired from not just a different sensor but also from different platforms like UAVs and drones. The models can be used and tested for performance of on-board processing by focusing on the real time detection while acquiring images. The experimental setup could be easily extended from object detection using bounding boxes to instance segmentation algorithm by using masked labels which would also be an interesting focus area.

8. CONCLUSIONS & OUTLOOK

8.1 SUMMARY:

The motivation of the thesis is to develop and test an object detection model using CNN that provides a robust performance when subjected to different optical datasets acquired with different sensors, with different acquisition geometry, in different acquisition conditions, with different areas of acquisition and processed to different spatial resolutions. Three different datasets BKG, xView and ZKI tiled to the same image tile size were provided for this purpose. Two broader scenarios, generic and transferability were tested for the purpose. In a generic scenario, the models were trained and tested on independent training and testing splits of the same reference data (same sensor) and for the transferability scenario, the models were trained with one dataset and tested with another (different sensors). The first model was developed using a pretrained network provided by PyTorch and was termed as baseline model. The baseline model performance was improved by implementing data augmentation which was further improved by using finetuning technique. After multiple testing and evaluation in many different scenarios and especiallly on the ZKI dataset, which is an independent test set, the augmented model trained on xView data is observed to perform better in both generic and transferability scenarios.

8.2 CONCLUSIONS:

The individually tailored experiments for each research question, seem to have been fruitful in providing some interesting answers.

8.21 Where do two – stage methods stand?

The literature survey clearly indicates the area of dominance for one stage and two stage methods in object detection. One stage methods are preferred in a situation where processing time is critical whereas two stage methods are preferred when accuracy is critical. The Faster RCNN architecture is a two-stage method, chosen, as this thesis prefers accuracy of predictions over processing time. One stage methods can be preferred for generating live predictions on the fly in the on – board processing unit of the sensor apparatus. The results for Experiment 2 provides us the baseline model which is a simple pretrained (COCO) Faster RCNN architecture with a ResNet50 backbone trained on training splits of BKG and xView datasets. Predictions generated on the baseline models and the mAP generated for those predictions show that without much tweaking of the architecture, the model provides a decent performance.

8.22 What is the effect of image transforms on the performance of the model?

The introduction of data augmentation is the first attempt at improving the performance of the baseline models. An augmented model is obtained by training the baseline model on augmented data (contains tiles with image transforms) along with the images from original training split. From the results of Experiment 3, it can be observed that there is a very steep increase in mAP, when predictions are generated on the augmented model. When the results of baseline and augmented models are compared, an interesting observation is that larger buildings which were missed by baseline model are correctly identified by augmented models. This can be attributed to the fact that random scaling and random cropping image transforms, creates a few tiles with a zoomed in scale that might help the model learn features of larger buildings better.

8.23 What is the impact of fine tuning an RCNN model for object detection?

Fine tuning has been implemented by introducing two changes. A learning rate optimizer was added that decreases the learning rate after each epoch, if the validation loss does not decrease. The first 6 layers of each layer block of the backbone was frozen during training. The result of Experiment 4 shows that fine tuning does not necessarily increase the performance of the model. This stands in line with the general idea that freezing layers might sometime increase the processing speed but decrease the accuracy, since it is always a tradeoff between the two of them. The impact of fine tuning, especially freezing of layers, is observed to be dataset oriented. The finetuned model does perform better than the augmented model when trained on BKG dataset but does not have the same impact when trained on xView dataset. Given that BKG dataset has a higher spatial resolution compared to xView, the conclusion is that finetuning might have a better impact on very high-resolution data.

8.24 How good is the transferability of the model?

The answer to this question is the main objective and goal of this thesis. It also answers the question “which model is the ultimate winner in terms of transferability”? The pattern observed while training and testing on BKG and xView datasets in scenarios 3 and 4 for checking transferability, is retained while experimenting on the independent flood dataset. The finetuned model trained on BKG and augmented model trained on xView perform best on this independent dataset. The one model that performs the best among these two is the augmented model trained on xView dataset. This can be attributed to the fact that even though, xView dataset has a comparatively lower resolution to BKG, it is undeniable that it contains imagery with buildings from across the globe unlike BKG dataset which contains buildings only from Germany. The uniformity of building structures and geometry across Germany and the versatility of building

structures and geometry globally has played a factor in the performance. The versatility of building geometries has helped the model learn better, the features of different building types from xView and hence a better performance on an independent test set, despite the poor spatial resolution of xView.

8.2 NEXT STEPS:

Given the availability of zero to very sparse literature on successful implementation of model transferability in CNN, as discussed in Chapter 7, it would be interesting to train a model on a dataset fused with xView, BKG and many other datasets from different sensors with different GSD and evaluate it on an independent test set. It would also be interesting to use much powerful machines and train the model on the complete dataset to observe the improvement in mAP. The prediction score and IOU threshold values can be treated as hyper parameters for this learning problem, and it would be interesting to understand the tradeoff between them for such a combined dataset. With respect to fine tuning, this thesis work can be extended into answering a new research question as to which layers of the model affect the performance after fine tuning and how. This can be analyzed by generating feature maps. They are the outputs of each layer in the CNN model that can be visualized. Feature map visualization will provide insight into which layer of the model has actually impacted the model performance. It would also be interesting to extend this experimental setup to other class labels. The experiment can also be extended to instance segmentation problem. Another inquisitive area would be to develop and train a one stage model like Yolov5 on the same dataset splits and evaluate the performance. A comparison of the results between Yolov5 and Faster RCNN would help understand which model has better processing time and accuracy and by what factor is there a difference in accuracy and processing time between this one stage and two stage methods. The same comparison can be repeated in an on-board processing, live object detection situation, for example while using drones, and can be used to check if Yolo v5 is faster than Faster RCNN.

Thus, building detection in the context of emergency response using artificial intelligence currently provides promising areas of research and application in remote sensing.

9. BIBLIOGRAPHY

Bai T, Pang Y, Wang J, Han K, Luo J, Wang H, Zhang H (2020). An optimized faster R-CNN method based on DRNet and ROI align for building detection in remote sensing images. Remote Sensing, 12(5), 762.

Bamler R (2021) Estimation Theory and Machine Learning Chapter 10: Neural Networks lecture slides, Technical University of Munich

Bunker RP, Thabtah F (2019). A machine learning framework for sport result prediction. Applied computing and informatics, 15(1), 27-33.

Camps-Valls G (2009) Machine learning in remote sensing data processing. In 2009 IEEE international workshop on machine learning for signal processing (pp. 1-6) IEEE.

Chen C, Gong W, Hu Y, Chen Y, Ding Y (2017). Learning oriented region-based convolutional neural networks for building detection in satellite remote sensing images. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 42, 461.

Ghoury S, Sungur C, Durdu A (2019). Real-time diseases detection of grape and grape leaves using faster r-cnn and ssd mobilenet architectures. In International conference on advanced technologies, computer engineering and science (ICATCES 2019) (pp. 39-44).

Girshick R (2015). Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (pp. 1440-1448).

Krizhevsky A, Sutskever I, Hinton GE (2012). ImageNet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

Lam D, Kuzma R, McGee K, Dooley S, Laielli M, Klaric M, Bulatov Y, McCord B (2018) xView: Objects in Context in Overhead Imagery. arXiv preprint arXiv:1802.07856.

Lary DJ, Alavi AH, Gandomi AH, Walker AL (2016). Machine learning in geosciences and remote sensing. Geoscience Frontiers, 7(1), 3-10.

Lee ALS, To CCK, Lee ALH, Li JJX, Chan RCK (2022). Model architecture and tile size selection for convolutional neural network training for non-small cell lung cancer detection on whole slide images. Informatics in Medicine Unlocked, 28, 100850.

Li M, Zhang Z, Lei L, Wang X, Guo X (2020). Agricultural greenhouses detection in high-resolution satellite images based on convolutional neural networks: Comparison of faster R-CNN, YOLO v3 and SSD. Sensors, 20(17), 4938.

Lin TY, Maire M, Belongie S, Bourdev L, Girshick R, Hays J, Perona P, Ramanan D, Zitnick CL, Dollar P (2014). Microsoft coco: Common objects in context. In European conference on computer vision (pp. 740-755). Springer, Cham

Majd RD, Momeni M, Moallem P (2019). Transferable object-based framework based on deep convolutional neural networks for building extraction. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 12(8), 2627-2635.

O'Shea K, Nash R (2015). An introduction to convolutional neural networks. arXiv preprint arXiv:1511.08458.

Padilla R, Netto SL, Da Silva EA (2020). A survey on performance metrics for object-detection algorithms. In 2020 international conference on systems, signals and image processing (IWSSIP) (pp. 237-242). IEEE

Ren S, He K, Girshick R, Sun J (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28.

Saeedi P, Zwick H (2008). Automatic building detection in aerial and satellite images. In 2008 10th International Conference on Control, Automation, Robotics and Vision (pp. 623-629). IEEE.

Simonyan K, Zisserman A (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Sirmacek B, Unsalan C (2008). Building detection from aerial images using invariant color features and shadow information. In 2008 23rd international symposium on computer and information sciences (pp. 1-5). IEEE.

Voigt S, Giulio-Tonolo F, Lyons J, Kučera J, Jones B, Schneiderhan T, Platzeck G, Kaku K, Hazarika MK, Czaran L, Li S (2016) Global trends in satellite-based emergency mapping. Science, 353(6296), pp.247-252

Westen VCJ (2000). Remote sensing for natural disaster management. International archives of photogrammetry and remote sensing, 33(B7/4; PART 7), 1609-1617.

Wong MS, Zhu X, Abbas S, Kwok CYT, Wang M (2021). Optical remote sensing. In Urban informatics (pp. 315-344). Springer, Singapore.

Yang W, Zhang X, Luo P (2021). Transferability of convolutional neural network models for identifying damaged buildings due to earthquake. Remote Sensing, 13(3), 504.

Yuan X, Azimi S, Henry C, Gstaiger V, Codastefano M, Manalili M, Cairo S, Modungo S, Wieland M, Schneibel A, Merkle N (2021). Automated building segmentation and damage assessment from satellite images for disaster relief. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 43, 741-748.

Zhao ZQ, Zheng P, Xu ST, Wu X (2019). Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11), 3212-3232.

Zhu X (2021) Remote Sensing – Advanced Method, Chapter 1: Introduction lecture slides, Technical University of Munich

Zou Z, Shi Z, Guo Y, Ye J (2019). Object detection in 20 years: A survey. arXiv preprint arXiv:1905.05055.

10. APPENDIX

Access to implementation is through DLR GitLab repository in obj_det_deivasihamani_dharani and is strictly request based.