Deep Learning Approach for Complex Image Processing Problems: Case Study with Explosion Detection and Satellite Image Segmentation

A thesis Report

Submitted in partial fulfillment of the requirements for the Degree of

Bachelor of Science in Computer Science and Engineering

Submitted by

Shafayet Ul Islam	**170104029**
Rishadul Islam Khan	**170104047**
Shams Al Ahsan	**170104005**
Md Sakib Sarwar	**160204002**

Supervised by

Dr. Mohammad Shafiul Alam

Professor

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Department of Computer Science and Engineering

Ahsanullah University of Science and Technology

Dhaka, Bangladesh

January, 2022

CANDIDATES' DECLARATION

We, hereby, declare that the thesis presented in this report is the outcome of the investigation performed by us under the supervision of Dr. Mohammad Shafiul Alam, Professor, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology, Dhaka, Bangladesh. The work was spread over two final year courses, CSE4100: Project and Thesis I and CSE4250: Project and Thesis II, in accordance with the course curriculum of the Department for the Bachelor of Science in Computer Science and Engineering program.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Shafayet Ul Islam
170104029

Rishadul Islam Khan
170104047

Shams Al Ahsan
170104005

Md Sakib Sarwar
160204002

CERTIFICATION

This thesis titled, **“Deep Learning Approach for Complex Image Processing Problems: Case Study with Explosion Detection and Satellite Image Segmentation”**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for the degree B.Sc. in Computer Science and Engineering in January, 2022.

Group Members:

Shafayet Ul Islam	**170104029**
Rishadul Islam Khan	**170104047**
Shams Al Ahsan	**170104005**
Md Sakib Sarwar	**160204002**

Dr. Mohammad Shafiul Alam
Supervisor
Professor and Head
Department of Computer Science and Engineering
Ahsanullah University of Science and Technology

ACKNOWLEDGEMENT

All praise to the almighty ALLAH for his blessing that helped us to finish this academic thesis with sound health and mind. In the beginning, we would like to show our gratitude to our supervisor Dr. Mohammad Shafiul Alam, Professor and Head, Department of Computer Science and Engineering, Ahsanullah University of Science and Technology for his constant support and encouragement. His engagement, constant monitoring, guidance, and constructive criticism helped us to understand the value of our project and the impact it can create in social life. We also want to thank all the faculty members, classmates, and staff of the Computer Science and Engineering department for their support and encouragement. Finally, we also would like to show our deepest gratitude to our parents and family members for their support and inspiration.

Dhaka
January, 2022

Shafayet Ul Islam

Rishadul Islam Khan

Shams Al Ahsan

Md Sakib Sarwar

ABSTRACT

Image processing is mostly used for the modification of images or extracting important information from images. Several techniques have been introduced for performing such feature extraction operations, but most of them do not decently help machine learning models to identify objects, events or pattern from the images properly. On the other hand, both machine learning and deep learning models that are used for computer vision application requires a large number of data for training and testing purposes. In this thesis, we have implemented and compared several deep learning models along with a generative model where we used no previous feature extraction method and a small number of data samples. The experiments were done in two major computer vision domain — real-time image classification and image segmentation. For the classification task, we detected and classified explosion type from CCTV footage in real-time. To solve the problem, both supervised and semi-supervised approaches have been adopted. In the supervised approach, deep learning models such as Convolutional Neural Network (CNN), Hybrid CNN-LSTM, CNN-GRU have been used. In the semi-supervised approach, Generative Adversarial Networks (GANs) have been implemented for the classification purpose and the model was able to obtain accuracy similar to the other deep learning models even with reduced datasets. For the segmentation task, we segmented river paths from satellite images with Generative Adversarial Networks (GANs). For both case studies, the dataset has been generated from the scratch and the performance of the models have been compared with each other.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A crucial step in every computer vision system is extracting features. Extracted features are then fed into several machine learning models that analyze and provide solutions or decisions based on the application domain. Mostly the feature extraction portion is done by performing hand-engineering methods or algorithms like Principal Component Analysis (PCA) [11], Linear Discriminant Analysis (LDA) [12], Independent Component Analysis (ICA) [13], etc. But the problem of such methods could be a high chance of overfitting, independent consideration of each feature, ignoring the useful feature or relation between feature and classifier. In this thesis, Deep learning techniques were applied in several real-life applications where all the feature extraction or handling was implicitly done by Convolutional Neural Network, Hybrid CNN-RNN, and Generative Adversarial Networks (GANs) model.

The problem that we tried to solve, falls under the domain of binary and multi-class classification and image segmentation. As a developing country with an enormous amount of population, Bangladesh often has to deal with incidents that lead to serious injuries of day-to-day people. Among those occurrences over the year, several explosions such as gas stations, factories, chemical explosions not only injured people badly but also took a decent amount of life. Conclusion after the investigation of these sorrowful events direct towards the delay of detection and notification of the incident. Thus in our thesis, the first case study we worked with is classifying explosion and non-explosion in real-time using CCTV footage. In addition to that, we also categorized the explosion into three major classes which are — House Explosion, Electrical Explosion, and Vehicle Explosion. After classifying the explosion type we used the YOLOv5 algorithm that detects the position of the fire in real-time. In classification, we used raw CCTV footage as a dataset collected from several social platforms like Youtube, DailyMotion, and models such as CNN, CNN-LSTM, CNN-GRU, and pre-trained VGG16 was used for performing the classification task. A semi-supervised approach was also applied where we reduced the dataset and obtained similar accuracy for classifying explosion and non-explosion footage that indicates that a small dataset can also be used for the classification task.

For our second case study, we detected a river path from the satellite image. This task falls under image segmentation and our approach to solve the problem was by using Generative Adversarial Networks (GANs). The real-life application of river path segmentation can be understanding the pathway of the river by analyzing several years of the satellite image, behavior of river paths during or after the flood, finding fertile land by analyzing river path directions, etc. To train the GANs model we manually collected around 500 high-resolution satellite images from EarthExplorer and marked the river path using software like ArcGIS. Then by using adversarial loss the generative model was able to segment river path from the satellite image.

All the feature extraction and understanding of the image pattern in these two major applications was done by the deep learning model. We provided the raw image and the model was able to analyze the pattern and take proper decision. In explosion detection, the model obtained 99.89% of text accuracy and for classifying the explosion class the model achieved more than 99.00% of test accuracy. In the segmentation, the achieved adversarial loss was 2.738 and the L1 loss was reduced to 0.090, which indicates that how good the generative model was, in terms of generating the segmented river path image.

1.1 Motivation

According to fortune [14], "The founder of google brain once built a computer vision system with 350 million images. But when he asked the audience at a recent manufacturing conference how much data they used in their systems, the majority of attendees replied around 50 images or fewer."

The advancement of machine learning and deep learning has shaped technology and opened opportunities that once were unimaginable. But to train a machine learning model, the features of the data have to be properly extracted first by using pattern recognition algorithms. On the other side, deep learning models require a large amount of data. Our motivation lies in designing a deep learning model that requires no hand-engineering feature extraction and uses less data to train the model.

We took two important domains of computer vision — Classification (binary and multi-class) and segmentation and tried to implement our vision. For the classification task, we classified explosion (along with the three major categories) and non-explosion in real-time from CCTV footage, and for segmentation, we segmented river paths from the satellite image.

To implement our vision we tested several deep learning models such as Convolutional

Neural Network (CNN), Hybrid CNN-RNN models, GANs for semi-supervised learning, and pix2pix GANs for image segmentation. For the supervised approach, we used a dataset that has over 7,000 samples whereas for the semi-supervised model we used only 1,000 explosion or non-explosion images. In segmentation, we trained the pix2pix model with only 468 high-resolution images and found the expected result.

Image processing has a vital role to play in real-life applications. Our motive is to enhance, generalize and optimize the conventional process by adding deep neural networks and generative models in digital image processing by performing them together to solve major problems like explosion detection (multi-class classification) and river path segmentation.

1.2 Objectives and Expected Outcomes

As a developing country, Bangladesh often faces catastrophes that lead to causalities and property loss. Explosive materials used in industries, homes, vehicles often create hazards that injuries people and sometimes take a life of innocents. To address this issue, several sensor-based smoke detection system made their appearances. But the delay and limitations of these systems could not save people in appropriate time. In this thesis, we tried to predict and detect explosions from CCTV footage in real-time to guarantee the safety of day-to-day people. In addition, we also classified the explosion type and detected the position of fire to take necessary precautions.

For our second case study, we segmented river paths from satellite images using conditional generative adversarial networks (C-GANs). As in recent years, climate change has affected the natural cycle of weather, sudden storms, heatwave, severe drought has started taking place. These sorts of unexpected incidents affected the behavior of rivers such as unanticipated changes of the river path, sudden floods, etc. A lot of such information can be acquired from the proper segmentation of river paths from a satellite image. In recent times, segmented river paths from a satellite image require machine heavy software like ArcGIS. The result of our experiment demolished the use of such heavy software. The pix2pix model uses an image-to-image translation technique to segment the river path accurately enough to gain in-depth information about the behavior of the river path.

The objective behind our thesis experiment lies in two domains – technical advancement and social impact. For the technical advancement, our goal was to detect and classify (binary and multi-class) explosions without any pre-processing of the data and using a smaller dataset. The CNN, Hybrid CNN-LSTM, CNN-GRU models were trained with a larger dataset (data samples over 7,000) and obtained test accuracy of 99.82%. Now with a smaller dataset where the labeled data was only 500, the GANs model was able to classify explosion and non-explosion images with 95.81% of test accuracy. Even in explosion classification the general CNN models were trained with more than two thousand data and obtained accuracy over 97.99% test accuracy whereas only with 500 labeled data (each class having 125 samples) the GANs model (semi-supervised approach) was able to achieve 92.76% of test accuracy. Thus the semi-supervised approach can get accuracy similar to the supervised approach with lower data samples. In segmentation, the pix2pix model was able to segment the river with 0.090 L1 loss which points towards accurate segmentation of the satellite image. From the point of view of social impact, our model was able to detect, classify explosions from CCTV footage that can help people from getting injured. And the river path segmentation can help us to understand the change or river path or predict floods from satellite image only.

1.3 Scope of Research

In recent years, the use of explosive materials has increased as most of the countries now are in developing states. The use of such explosive items increased the danger of explosions that caused millions of lives around the world. To create a precaution, several ideas like sensor-based fire detection, machine learning-based approaches have been introduced. The limitation of these approaches are: in the sensor-based detection system, sensors sometimes delay to activate properly and in some cases, if the smoke of the fire does not spread out certain level, the sensor does not activate. Even in larger areas, sensors can not cover the whole place thus the proper alarming fails to appear. In a machine learning-based system, the images on which the model is trained to take a lot of pre-processing. Most of the time, these pre-processing steps take heavy computational power and time. In the machine learning approach, the training requires feature extraction that is often hand-engineered and algorithm-oriented. So, if the algorithm fails to extract the feature then the model can be trained with unwanted features which eventually can lead towards false detection of explosion. In this thesis, we tried to avoid both sensor-based and machine learning-based approaches. We created several deep learning models and trained the data without any preprocessing. The model itself finds a pattern of images during training. The deep learning model both the supervised and semi-supervised approach created a wide range of door in explosion detection and classification which can be fine-tuned and improved.

In the segmentation domain, most of the research was done in the medical area. Though in recent years some of the geographical places have been segmented for gathering in-depth information about the land. But in most of the segmentation-based tasks, the images that are used as data samples are synthetic which means most are computer-generated. As a real dataset is hard to create and most of the research has been done on land or land-object such as houses, forests, vehicle segmentation, our research is mostly focused on river-path segmentation. We used satellite images and created a dataset from the scratch by using ArcGIS. Segmentation of river path can help us to gain knowledge about the behavior of a river for several years along with detecting fertile land or flood detection.

1.4 Organization of thesis

The thesis book contains several chapters in total. Each chapter is a step in the stairway that leads towards the end destination. A short description of each chapter is stated below.

Chapter one describes the core idea of our thesis. It also states the both technical and social impact of our research and the motivation behind our experiment. We also showed the scope of research in our proposed idea in both classification and image segmentation which might help other researchers to walk in our path and improve the techniques that we have described.

For research purpose we have studied several previous work on event detection, accident prediction, fire detection using machine learning approaches. Even for image segmentation we went through research paper where authors used CNN or Generative models in several segmentation area. Chapter two mostly describes several research paper, their core concept, methodology and limitations.

For image classification and segmentation, we have used deep learning models like Convolutional Neural Network (CNN), hybrid models such as CNN-LSTM, CNN-GRU, Generative Adversarial Networks (GANs), pre-trained VGG-16. In chapter three, the in-depth description of the functionality of the models and the core concept of their architecture has been described.

To train the model, the most important step in deep learning is dataset and pre-processing. In chapter four, we started the procedure of collecting data (from youtube, daily motion) and extracting frames from videos and handly-picking important footage from them have been described. The classification process of the dataset and organizing them for feeding them into the model have also taken a portion of the chapter. For detecting fire from CCTV footage we created a separate dataset for fire detection using YOLOv5 which was done by the software "LabelImg". In this chapter, we also described step by step guide of EarthExplorer and ArcGIS which have been used for creating the dataset of satellite image segmentation for river path detection.

In chapter five, we described in-depth details of the architecture of the models we have implemented. It covers information about convolutional layer, activation function such as "ReLU", "Leaky ReLU", MaxPooling layer, LSTM, and GRU units along with U-Net architecture of generator in the GANs.

In chapter six, we have shown the comparative analysis between the supervised and semi-supervised approaches for explosion detection and classification. We elaborately explained the founded result with train-test accuracy, precision, recall, adversarial loss, and L1 loss. The number of epochs and iteration, the optimizer that was used for the training was also mentioned with details graphs and tables in this chapter.

Though we have created a dataset and implemented models, our experiments are far behind perfect. In chapter seven, We discussed issues that can be problematic for the model in both segmentation and classification and how we can make our model more generalized towards analyzing a different kind of classification or segmentation.

Chapter 2

Literature Review

The experiment of our thesis covers two different domain of computer vision area – Image classification and Image segmentation. To understand both of the area in-depth, we have studied several research papers about image classification, video classification, event detection, generative models, image segmentation. Details of some papers are stated below.

2.1 Fire detection in video sequences using a machine learning system and a clustered quantitative image mark

Author: Abolfazl Zargari Khuzani, Rakshit Agrawal, Najmeh Mashhadi

Paper Description:
As most of the traditional approach of detecting fire and smoke is sensor-based, using video footage to detect fire is a challenging one. The sensor takes time to activate and suitable for a bounded place, but in wide-area smoke detectors cannot provide much-expected output. To solve the fire detection problem from video footage, this paper [1] applied multiple feature extraction methods and a multi-layer neural network.

Methodology:
The major steps that are followed in this paper are:

i. Videos are converted into three second clip for both fire and non fire videos.

ii. Videos were converted into frames. Color space model was applied on these frames so that fire objects can be easily distinguished from the surroundings.

iii. Various feature extraction methods were applied to the frames and grouped together to generate 270 features.

iv. As the number of features is large, Kernel-PCA was applied to reduce the features.

As for feature extraction, Fast Fourier Transformation, Wavelet, GLCM, GLDM, Shape, and density were applied to generate a series of vectors containing 15 elements.

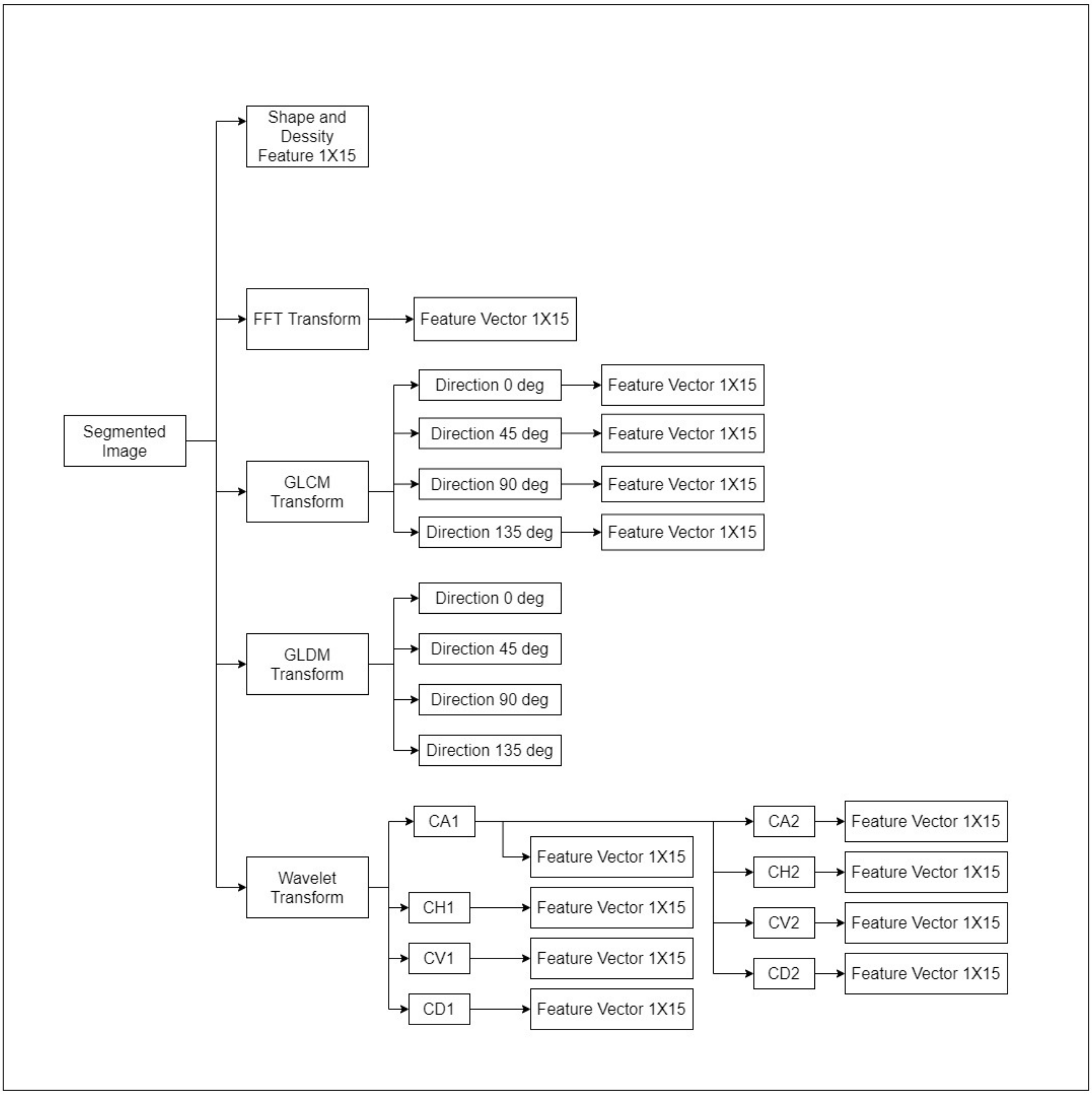


Figure 2.1: Feature Extraction Flowchart [1]

A total of 270 features were used as input for the multi-layer neural network.The proposed method of the paper detects fire with 96% Precision and 90% Recall.

Research Gap:

In this paper, the feature extraction process is done by hand-engineering. The pre-processing part does this huge calculation which is computationally expensive. Before sending to the neural network using this method for every frame of the footage will be expensive in terms of hardware processing.

2.2 Smoke Detection Based on Deep Convolutional Neural Networks

Author: Chongyuan Tao, Jian Zhang, Pan Wang

Paper Description:
Fire and smoke come one after another or vice versa. Effective smoke detection can prevent accidents related to fire or other chemical hazards. Detecting smoke of various shape textures and colors is extremely challenging. In this paper [15], an approach is proposed to detect smoke by applying a deep convolutional neural network.

Methodology:

(i). The convolutional model was fed with raw pixel values as input.

(ii). The model created a set of features automatically from the pixel values of the images.

(iii). The model is trained with 80,000 iterations of batch size 128.

(iv). The learning rate is set to 0.01 for the first 40,000 iterations and was decreased at the rate of 0.5 for every 10, 000 iterations.

This method of detecting smoke in images reaches an accuracy of 99.4% with 0.44% of false positives.

Research Gap:
In this paper, the dataset had 10,712 training images and 10,617 test images which were manually labeled. Labeling this huge amount of data could be hard for humans to manually label. A semi-supervised approach could be used to solve this manner. Another issue could be, smokes blocked by objects may not be detected by the model as the training data does not include any image that is related to this domain.

2.3 Machine Learning Algorithm for Fire Detection using Color Correlogram

Author: Jubeena. B. Maheen, Aneesh R P.

Paper Description:
As the sensor-based fire detection is limited to a certain area, using surveillance footage to detect fire might be able to solve the wide-area issue. In the paper, five steps are followed to determine the fire region.

Methodology:

(i). **Image Acquisition:** In this step, images are collected from the source in RGB format. These sources can be a High-resolution camera, web camera, or cc tv. Because of the different sources, captured images can be low-quality or high-quality images.

(ii). **Pre-Processing:** The filtering process is used to remove the unnecessary components. For this processing, a Gaussian filter is used for removing the unwanted interference from the acquired image.

(iii). **Image Segmentation:** The purpose of image segmentation is to clear and modify the images into more relevant and easier to evaluate. In this step, the whole image is divided into multiple segments in order to extract specific features from the image. The threshold method is used which can be performed by using a threshold value to convert a black and white image into a corresponding binary image.

(iv). **Feature Extraction:** In this step, large input images are altered into a diminished set of features. The problem occurs while operating complex data because a huge number of variables involved in it. Here, Color Correlogram is used to remove the characteristics of a picture.

(v). **Classifier:** In this process, Naïve Bayes Classifier is used for regrouping the segmented image. This classifier is useful for statistical classification.

In this paper [16], with the Naïve Bayes classifier, the proposed system could test Fire 2018 dataset and obtained 96.97% accuracy.

Research Gap:
In the preprocess segment of this paper, the edge detection algorithm has been used to differentiate the fire and non-fire segment, in case of an explosion if the whole scene of the camera is covered with fire this process would not help to detect properly. On the other hand, this paper does not offer any attention on solving the smoke detection issue which is a vital element of any explosion.

2.4 Real Time Object Detection and Tracking Using Deep Learning and OpenCV

Author: Chandan G, Ayush Jain, Harsh Jain, Mohana

Paper Description:
In this paper [17], Single Shot Algorithm (SSD) algorithm is used to perform object detection. SSD algorithm is based on VGG-16 [18] architecture. As the SSD algorithm uses VGG-16 as a base, it is very easy to implement. In this paper, SSD is used over YOLO [19](You Only Look Once) because YOLO performs better when accuracy is less important than speed. MobileNet is also used in this paper for optimizing the performance and latency for portable and embedded vision-based systems where process control is absent.

Methodology:

(i). For object detection Frames were captured from video. The difference is estimated from the consecutive frames. A local mean optical flow algorithm is used to enhance the frames. Background subtraction is used for removing background from the frames so that the detection model can box the object with great confidence.

(ii). For object tracking, frames that are consecutive are fed to the model in a fixed interval of time so that the rate of real-time detection is increased. The model is trained by feeding the converted frames and then for tracking the objects the videos are separated into frames and then directly fed to models.

(iii). The authors of the paper trained their model with the COCO dataset of object detection. The model is trained to detect 21 objects. The average confidence level of detecting objects was 99.68%.

Research Gap:
The disadvantage is maybe the way it has approached the object detection, small objects may not be detected by the model.

2.5 Multi-View Object Detection Based on Deep Learning

Author: Cong Tang, Yongshun Ling, Xing Yang, Wei Jin and Chao Zheng

Paper Description:
In this paper [20], a multi-view object detection approach is introduced. In traditional deep learning and regression models perform better detecting objects of large scale but often

fails to detect the object of small in scale. With the help of multi-view object detection based on the deep learning approach, the average F-measures were higher than traditional approaches. The multi-view approach is used along with YOLO, YOLOv2, and Single Shot Detection model. The mAP score was acceptable comparing with other deep learning models such as RNN, RFCN, SPP-net, RCNN, etc. Also, the run time per image is greatly reduced by the multi-view object detection approach.

Methodology:

(i). To get good performance in a small object detection case, Images were segmented into five sections as views and each individual view is sent as an input to the models. The final output image is made by combining the output of each view.

(ii). YOLO [19], YOLOv2 [21], and SSD are used as the base of the model and these are trained with VOC 2007 dataset.

(iii). multi-view YOLO, multi-view YOLOv2, and multi-view SSD results are compared with the traditional deep learning models like Faster RCNN, RFCN, SPP-net, etc.

(iv). The performance is acceptable comparing with other models. The average mAP score is very close to other models, sometimes beating them.

(v). The run time per image is lower than other models. Thus decreasing computational cost.

Research Gap:
Although the multi-view approach gives satisfactory results, applying this approach to other state-of-the-art models may provide better results.

2.6 Semi-Supervised Learning with Generative Adversarial Networks

Author: Augustus Odena

Paper Description:
In this paper [22], a semi-supervised approach is applied to GANs by forcing the discriminator to classify the data. MNIST [23] dataset of numbers is used to train the SGAN [24] to classify the images. The classification performance is of the SGAN is compared with the traditional CNN model to justify the correctness with fewer training examples.

Methodology:

(i). SGAN is developed with a modification on the discriminator model being able to train the generative model and also classify the data simultaneously.

(ii). The SGAN has trained on MNIST datasets of example 25, 50, 100, and 1000 examples. The classification accuracy of the discriminator is compared with a simple CNN model that is not trained with the GAN approach.

(iii). The accuracy of the classic CNN and SGAN is above 96% in the case of 1000 sample images and in all cases (25, 50, and 100 examples) the SGAN performs better (above 80%).

(iv). With small dataset, SGAN can be trained that can produce much better output than traditional classification models.

Research Gap:
GANs may not output the same result every time because the learning may not be the same every time the model is trained. Finding the best output may need fine-tuning of the model for both generative and discriminator models.

2.7 Semi supervised Hyperspectral Image Classification Based on Generative Adversarial Networks

Author: Ying Zhan , Dan Hu, Yuntao Wang, and Xianchuan Yu
Paper Description:
In this paper [25], a semi-supervised GAN is designed based on 1-D GAN (HSGAN) [26] to classify Hyperspectral images. The performance of the model is evaluated on Airborne Visible Infrared Imaging Spectrometer Image data. HSI images have more bands and continuous spectral features, so it has a significant advantage over remote sensing data. Many traditional machine learning algorithms have performed better in these types of data. But to gather HSI images is much more difficult than normal ones. With the majority of unlabeled data and little label ones, the semi-supervised approach is much more suitable than others.

Methodology:

(i). The HSGAN is first trained on unlabeled data. By the training, the number of epochs and batch size are first set to an optimum value.

(ii). After the training, the generator and discriminator model's weights are saved.

(iii). After that, the last layer of the discriminator is replaced with a softmax function for classification output.

(iv). Then, with the fine-tuned epoch number and batch size, the model was trained on the labeled data and the classifier cost was updated using the stochastic gradient method.

The HSGAN classifies the HSI dataset with satisfactory accuracy of 80% than other traditional models such as PCA KNN, CNN, etc. With a little amount of labeled data, the performance of the model is much better.

Research Gap:
The hyperspectral image is basically a processed image that is created from the electromagnetic spectrum. Getting or creating such images for various locations is difficult. As the sample in the dataset is small, using a deep neural net sometime creates a problem of overfitting and it creates a problem in fine-tuning the model during training.

2.8 A Deep Learning based Accident Detection System

Author: Gokul Rajesh, Amitha Rossy Benny, Harikrishnan A, James Jacob Abraham, and Nithin Prince John

Paper description:
In this paper [27], a system is introduced that can detect road accidents and also provide an alert message to the most proximate control room immediately. This system is based on deep learning techniques which can use a convolutional neural network.

Methodology:

(i). **Dataset:** The dataset consists of 5000 images, among them 2500 accident images and the rest of them are 2500 non-accident images. All of the images were taken from google images and these images are converted into the same format and size.

(ii). **Training and validation:** Among 5000 images, 4000 images were used for training. Then these training images were converting into two classes and also passing images through CNN(convolutional neural network). For validation, they use 1000 images belonging to two classes which are accident and non-accident. This dataset was trained for 30 epoch and this model is created by using sequential API. For classifier, Softmax is used here. And for data augmentation, an Image data-generator is used. The train batch size is fixed to 100 and the validation batch size to 10.

(iii). **Video Classification:** For video classification, deque was used. This deque is used to get rolling prediction averaging. After that mean subtraction was applied and this mean subtraction help to find the results of predictions of each frame. Deque saves this result. After that, the model is capable to detect it is an accident or not.

(iv). **Alert System:** By using the GSM module, an alert message is sent to officials about the accident. GSM module stored certain numbers. So, at the time an accident occurs, an alert will be sent to those numbers with an accident location.

Research Gap:
In this paper, the dataset is created by google images and images that are captured by mobile or CCTV, or camera. But in a real-life scenario, only a CCTV camera is used for capturing images. The model used in this system, correctly predicts eighty-five percent. So, there is a possibility of a false alarm.

2.9 Accident Detection using Convolutional Neural Networks

Author: Sreyan Ghosh , Sherwin Joseph Sunny, Rohan Roney

Paper description:
In this paper [28], a system is created which would detect an accident from CCTV. The dataset created for the purpose was trained using the CNN model. The model was then able to classify each frame from the CCTV footage as accident or non-accident. For components, Raspberry Pi 3 Model B+, GSM Module SIM800L, and for software Keras and OpenCV are used for detecting accidents and non-accident.

Methodology:

(i). In this paper, Inception v3 is used for building the model. Every frame from captured video was passed through Inception v3 and the output from the final pool layer of the network was saved. The purpose of the Inception v3 was to extract the features. After that, those vectors of the feature pass through to RNN.

(ii). A sequence of extracted features was generated using Inception v3. Several accident frames were gathered from video footage. These frames are used for training different

RNN models without passing continuous images to CNN.

(iii). Before training, the convolution layers of the Inception v3 were stalled. Then the images are trained with LSTM and fully connected layers. For final classification, sigmoid activation function have followed these layers.

(iv). For training the model, All the layers were trained using the Image Data Generator in Keras.

The model gave an accuracy ranging from 82% to a maximum accuracy of 98.76%. On an average the model gave an accuracy of more than 92.38%.

Research Gap:
In this system, a Raspberry Pi 3 B+ Model is required which is used here as a portable and remote computer to be set up on a CCTV camera. And it increases the implementation and maintenance cost.

2.10 An Automatic Car Accident Detection Method Based on Cooperative Vehicle Infrastructure Systems

Author: DAXIN TIAN, CHUANG ZHANG, XUTING DUAN, AND XIXIAN WANG

Paper description:
In this paper [29], a system is introduced which is an automatic car accident detection method based on Cooperative Vehicle Infrastructure Systems (CVIS), and here machine vision is also proposed. For the dataset, CAD-CVIS is used because this dataset contains various kinds of accident types, the location of the accident, and weather conditions at the time accident occurred. For the model, they developed a deep neural network model YOLO_CA based on CAD_CIVIS and a deep learning algorithm. To find small objects, Multi-Scale Feature Fusion was utilized here.

Methodology:

(i). For dataset, videos and images were captured from CCTV. This dataset contains 633 car accident scenes, 3255 accidental frames, and 2,25,206 non-accidental frames. For annotating the location of the accident in the image, LabelImg was used here.

(ii). In this paper, they not only find whether there is an accident or not but also identify the location of the accident. Later that location is used to broadcast RSU the emergency message. In this paper, classification and location algorithms were divided into two-stage. And these are:

(a) Two-stage models are R-CNN, Fast R-CNN, Faster R-CNN, and Faster R-CNN with FPN. These algorithms are used for identifying regions in an image. After detecting objects by extracting the features of CNN.

(b) One stage model is SSD and YOLO. These algorithms can give an end-to-end detection service.

(iii). YOLO-CA is built with 228 neural network layers. And these layers are different kinds of basic components of YOLO-CA. Here, the upsampling layer is used in YOLO-CA for detecting small objects.

2.11 Image-to-Image Translation with Conditional Adversarial Networks

Author: Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros

Paper description:
The authors have investigated a general-purpose solution for image-to-image translation using adversarial networks. General adversarial networks are used to generate results that are indistinguishable from reality. The authors used conditional GAN (cGAN) that is suitable for image-to-image translation tasks, as cGAN [30] can be conditioned on the input data to generate corresponding output results. Various forms of cGAN are used for discrete labels, prediction form maps, realistic image generation, annotation of images, etc.

Methodology:

(i). The objective of cGAN is to minimize the generator and maximize the discriminator loss.

(ii). cGAN is also compared with normal GANs, where cGAN performed better than others.

(iii). The generative network generates images with random noise and feedback from the discriminator.

(iv). For the generative model, patchGAN is used and for the discriminator model, U-net architecture is used.

(v). Both the generative and discriminative models used BatchNorm-Relu as the activation function.

(vi). To optimize the whole network, minibatch SGD and Adam solver are used alternatively between generator and discriminator.

Research Gap:
the idea is pretty straight forward and no lacking or other things are there as generative models are not bound to the model but to the hardware and training.

2.12 Satellite Image Segmentation based on Different Objective Functions using genetic algorithm: A Comparative Study

Author: S. Pare, A. K. Bhandari, A. Kumar, G. K. Singh, and S. Khare

Paper description:
As satellite image segmentation is usually required a lot of effort and time, the authors have used genetic algorithms to segment satellite images based on different objective functions. As image segmentation-related problems are vastly unoptimized and hard to solve, genetic algorithms are suitable for these problems [31].

Methodology:

(i). To optimize the genetic algorithm, the authors have proposed multilevel thresholding as thresholding differentiates target objects from background pixels.

(ii). Kapur's entropy [32], Otsu class variance [33], and Tsallis entropy [34] are used for the thresholding.

(iii). Later these optimizations techniques are compared with each other to find out the best technique.

(iv). Each of the techniques has been used to generate five thresholds to optimize the genetic algorithm

(v). Among all of them, Kapur's entropy with the genetic algorithm has provided better efficiency and feature measurement parameters compared to others.

Research Gap:
The authors have not used any kind of feature extraction technique for further optimizations.

2.13 Comparison of Different Convolutional Neural Network Architectures for Satellite Image Segmentation

Author: Vladimir Khryashchev, Leonid Ivanovsky, Vladimir Parlov, Anna Ostrovskaya, and Anton Rubtsov.

Paper description:
In this paper [35], the researchers used different CNN(Convolutional Neural Network) model architecture to detect geo-objects from a satellite image. For comparison of Satellite Image Segmentation, the researcher modified the model architecture and also used different satellite image-based databases such as PlanetScope, Landsat-8, etc. To calculate or analyze the result, the researcher compared their model output with an expert result.

Methodology:

(i). The researchers modified the CNN model by using three different model architectures. These three model architectures are LinkNet, U-Net and SegNet.

(ii). To compare these three new models, they calculated the model accuracies based on the DSTL Image Database.

(iii). As all three models have shown higher accuracies, they used Dice similarity coefficients(DSC) to find the best model.

(iv). Later the researchers simulated the results on PLANTSCOPE and LANDSAT datasets using CNN(Convolutional Neural Network) U-Net model architecture as this model architecture performed better than the other twos.

Research Gap:
The authors have not used any other models except CNN to compare Satellite Image segmentation.

2.14 Wildfire Segmentation on Satellite Images using Deep Learning

Author: Vladimir Khryashchev, and Roman Larionov

Paper description:
In this paper [36], the author segmented wildfire from satellite image using Convolutional Neural Networks. For the purpose, two different dataset was used — The Resurs dataset that contains 10 bit three channel high resolution images and The Planet database that has 10 bit three channel high resolution spatial images.

Methodology:

(i). As each satellite image was high resolution, 256x256 portion was cropped and corresponding mask image was generated.

(ii). For data augmentation, the cropped images were rotated in 90, 180 and 270 degree which helped to generate more images than the dataset itself contains.

(iii). All the pixel value was converted in between 0 to 255.

(iv). Random chromatic distortion (HSV color format) was applied on each image that helped the deep-learning algorithms to deal with noisy images such as clouds in the satellite image.

(v). U-ResNet34 was used for training and testing. The model has the U-Net architecture with encoder-decoder. The encoder is used for downsampling whereas the decoder upsamples the satellite image.

(vi). Several performance measurement metrics such as F1, IoU, DSE were used for measuring the performance.

2.15 Towards Accurate High Resolution Satellite Image Semantic Segmentation

Author: MING WU, CHUANG ZHANG, JIAMING LIU, LICHEN ZHOU, and XIAOQI LI

Paper description:
The paper [37] proposes AD-LinkNet, a neural network for semantic segmentation. The

application of this neural network was to extract roads, buildings etc from the satellite images. The AD-LinkNet was able to perform better than the D-Linknet which was introduced in CVPR 2018

Methodology:

(i). For the dataset, the paper used three different dataset — DeepGlobe's road extraction dataset, DeepGlobe's land classification dataset and Inner Mongolia's land classification dataset.

(ii). The Ad-LinkNet contains encoder–decoder structure, serial–parallel combination dilated convolution, attention mechanism.

(iii). The model also contains two major part — At first part, the model has ResNet that uses the skip connection. Secondly, The model has an attention module. Eventually the model combines both of them.

(iv). During training the Hue of the images, saturation of the images and the original color value of the images were changed. Data augmentation was applied before the training.

(v). Metrics like IoU, mIoU were used for the performance measurement and comparison.

Chapter 3

Proposed Deep Learning Approach

In terms of explosion detection, each frame of the CCTV footage will be pre-processed and sent to our deep neural network which will classify if the given frame contains any sign of explosion. For, satellite image segmentation we have used generative (pix2pix) model that uses image to image translation technique for image segmentation.

3.1 Deep Convolutional Neural Network (CNN)

Convolutional Neural Network is a class of deep neural net which takes an image along with learn-able weights and biases as input and can distinguish among pictures. In traditional machine learning algorithm the feature extraction is done by hand engineering method, but a CNN can understand and extract feature if large set of data has been fed to it.

As we know, an image is nothing but matrix of pixel values. If the image is RGB then it contains 3 color channel stacked one after another. A CNN reduce the image into a form where it is easier to process and it tries not to lose feature that are needed for prediction or classification.

If the image is in grey scale format, then it contains only one color channel. The element involved in carrying out the convolution operation in the first part of a Convolutional Layer is called the Kernel or Filter. Kernel could be of any size but most of the time it is of $3\times3\times1$.

Figure 3.1, contains a glimpse of convolutional computation where the Stride Length $= 1$ (Non-Stride). Matrix multiplication operation has been done between filter and the input portion of the image over which the kernel is hovering. The objective of the Convolution Operation is to extract the high-level features such as edges, texture from the input image. The initial convolution layer takes out the lower level feature of the image such as edges, color where the later layers adapts the high level feature.

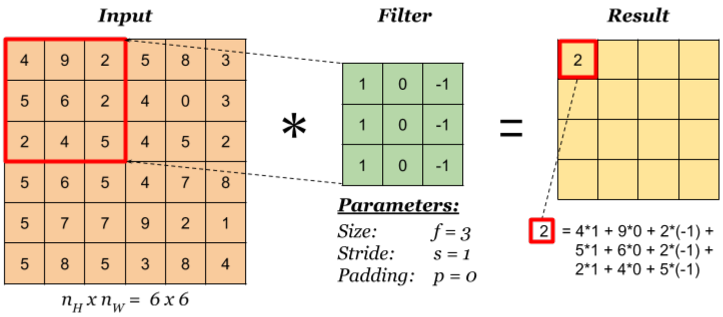


Figure 3.1: Convolutional Calculation [2]

Pooling Layer: Pooling layer reduce the spatial size of the convolved feature. This is to decrease the computational power required to process the data through dimensionality reduction. There are two kinds of pooling layer: max pooling and average pooling. Max pooling returns the max value of the portion which is enfold by the kernel whereas, Average pooling layer returns the average value of all the value of the portion enfolded by the kernel.

Fully Connected Layer: At the later portion of the CNN architecture fully connected layer is situated as they are cheap in terms of learning high level feature. The softmax activation function then used for multi-class classification which back propagates the error and adjust these weight and bias using algorithms like gradient descent.

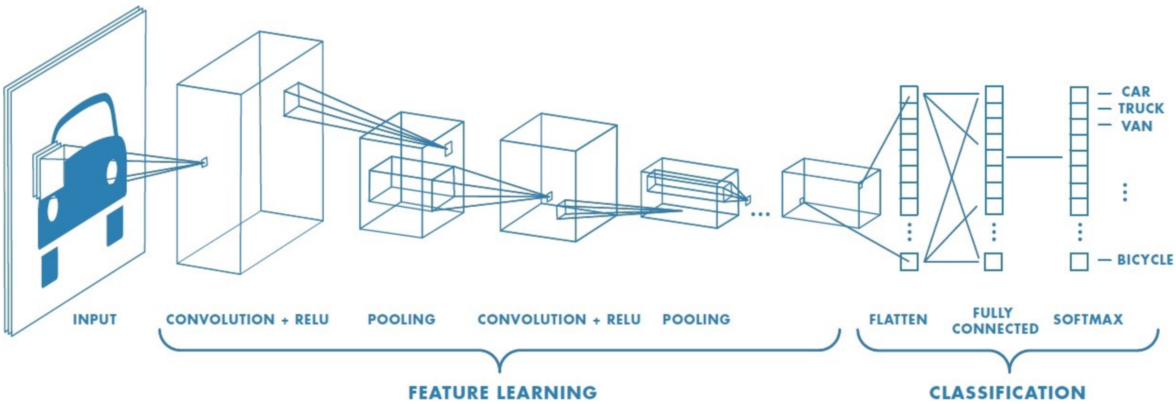


Figure 3.2: Convolutional Neural Network [3]

3.2 Recurrent Neural Network (RNN)

Recurrent neural network is a special form of artificial neural network. In traditional ANN, a fixed size of vector can be sent as an input, but in RNN, a series of input can be given with no predetermined size. In RNN, a single input can be sent to multiple layers, so that it can remember the past. As of it, the past decisions can influence the future knowledge. RNN tries to learn similarity while training and use the knowledge to generate output while testing or validating. As of the architecture of RNN, previous inputs and new inputs are given to generate one or more output, these output can also be sent as an input to one or more layers as inputs. There are multiple activation functions that can be used to determine whether a neuron of RNN would activate or not. Some examples are: Sigmoid, tanh, ReLU, etc. There are some types of RNN that are used widely. Some examples are: Bidirectional RNN, Long-Short Term Memory, Gated Recurrent units, etc.

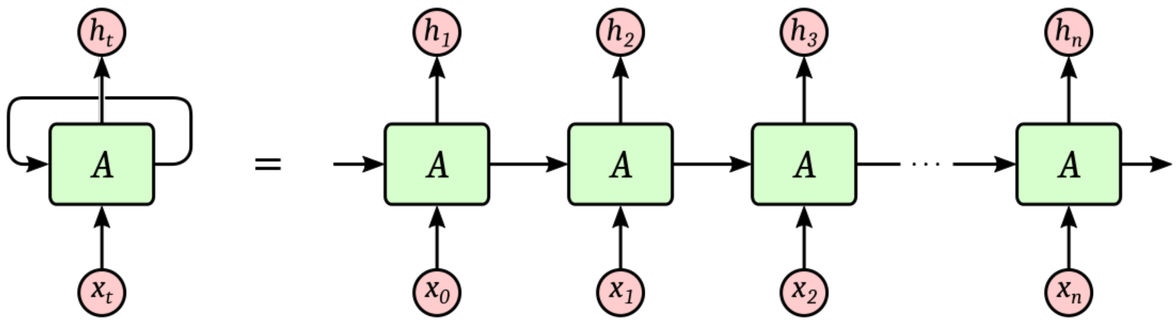


Figure 3.3: Architecture of RNN [4]

BRNN: In Bidirectional RNN, inputs can be sent to not only forward neurons but also sent to previous neurons. That means the current neuron can get knowledge not only about past decisions but also from future. These feature of BRNN improves accuracy by a lot.

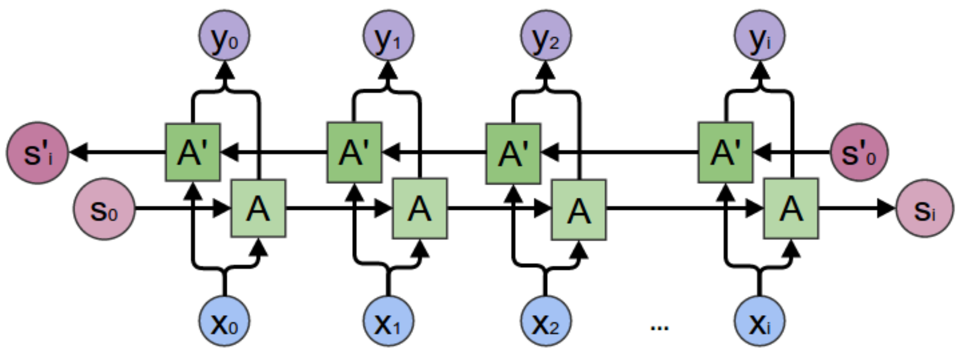


Figure 3.4: Architecture of BRNN [5]

LSTM: One of the most popular architecture of RNN is LSTM. In this architecture, a neuron can forget the previous knowledge at a random time interval. In some cases, decisions that are not coming from the recent past can decrease the accuracy of the model. In this architecture, a neuron has three gates, input, output and forget. With the forget gate, information that are decreasing the performance of the model can be erased and recent knowledge can be stored. In LSTM, neuron can forget these unnecessary decisions that are saved on the neuron. This architecture is widely used for classifying, processing, predictions and time series based data.

GRU: This variant of RNN is very similar to LSTM. This RNN architecture uses hidden layers instead of gate method to remember the previous decisions. In these hidden layers, the reset

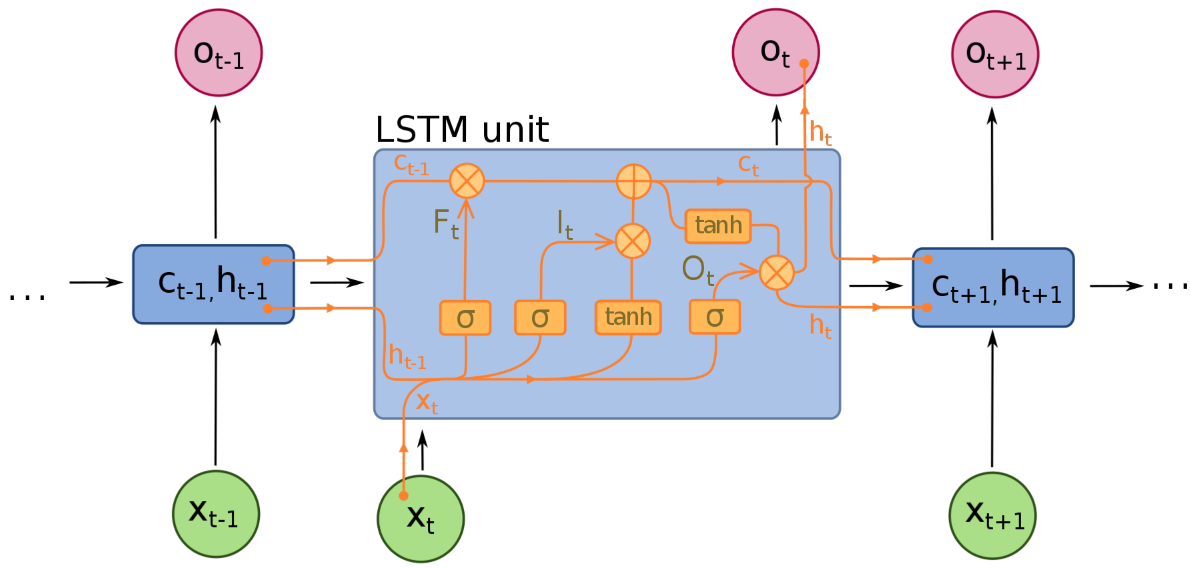


Figure 3.5: Architecture of LSTM [6]

and update of information is controlled.

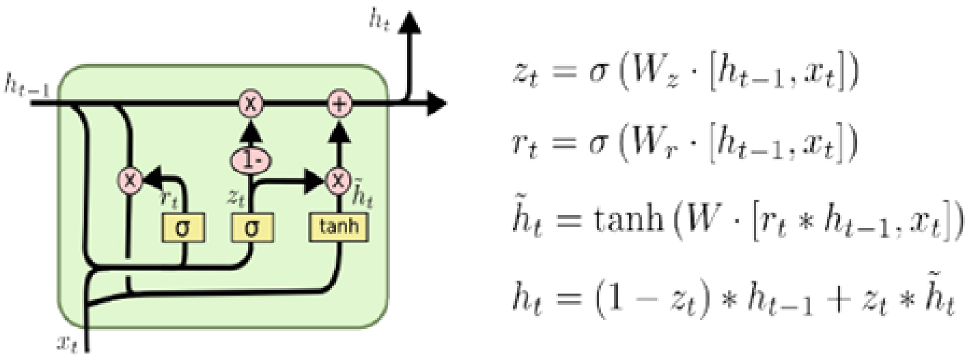


Figure 3.6: Architecture of GRU [7]

3.3 Pre-trained Model (VGG-16)

VGG stands for Visual Geometric Group. VGG-16 is a convolutional neural network with 16 layers containing trainable parameters. After every three layers there is a max pooling layer. These max pooling layers do not contain any trainable parameters.

VGG-16 takes 224x224 colored images as input. These images pass through a stack of convolution layers and max pooling layers. Max pooling layers are size of 2x2 with stride of 2. So, max pooling layers are non-overlapping windows. In the end the output of the last max pooling layer is flattened and pass through 3 dense layers. This is how output is generated in VGG-16.

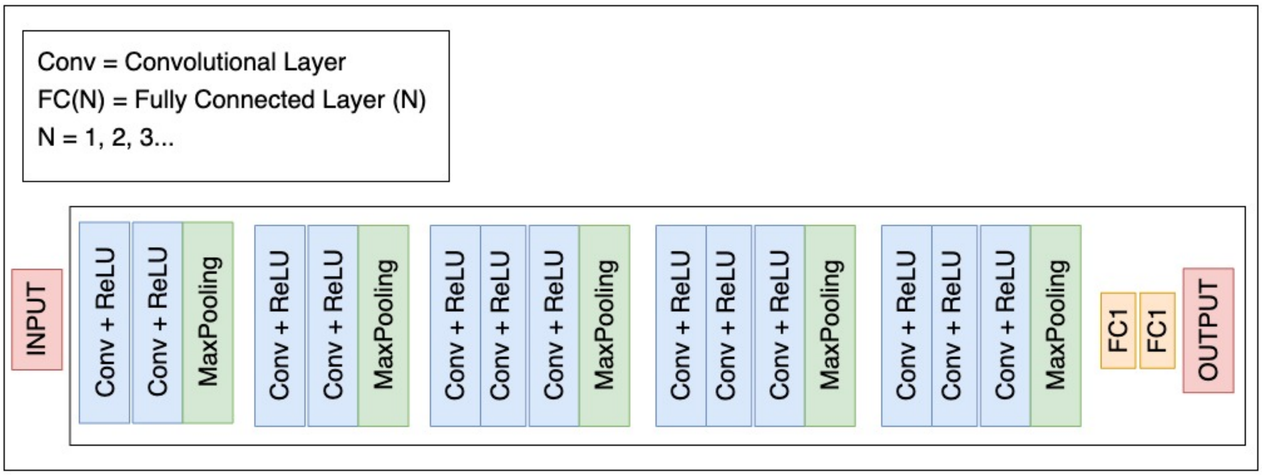


Figure 3.7: Architecture of VGG-16 [8]

3.4 Semi-Supervised Approach using Generative Adversarial Network

Generative Adversarial Networks (GANs) is a special machine learning framework which contains two different neural network inside of it.

The GAN is consists of:

(i). Generator Network

(ii). Discriminator Network

Functionality of GANs:

From a random noise distribution the generator tries to create an image with the help of the discriminator.

In the Figure 3.8, the discriminator takes both the real image (ground truth) and the generated image (considered as fake image) as input and classify if the generated image is fake or real using a feed-forward neural net or a CNN. While the discriminator consider the generated image as fake the error returns to the generator whose weights and biased get updated. It is important to keep in mind that, while the Discriminator is being trained the weight and bias of generator is held fixed and vice versa.

GANs for Classification: In our thesis, we will use the GANs architecture as a classification model. As explosion and non-explosion is hard to label manually, we will use a

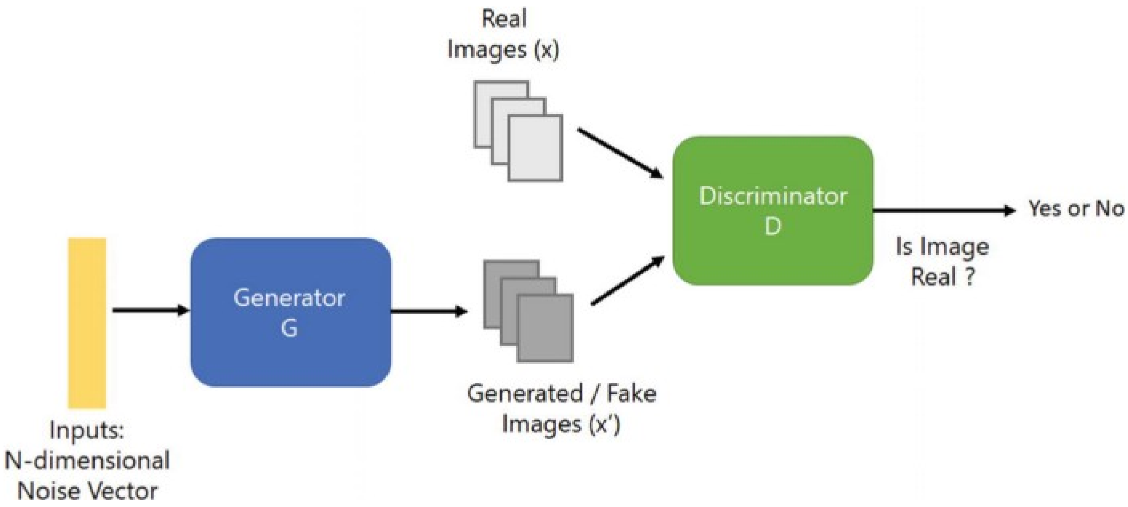


Figure 3.8: Architecture of GANs [9]

semi-supervised approach where some of the date will be labeled and most of them will not be.

Both the label and un-label data will go to the discriminator as input along with the fake generated image, the discriminator this time will output the probability of being the image of explosion, non-explosion and real/fake. Thus the model will be trained. By using a softmax activation function instead of using sigmoid, we can achieve such output. The error will then be back-propagated both in discriminator and generator at the same time.

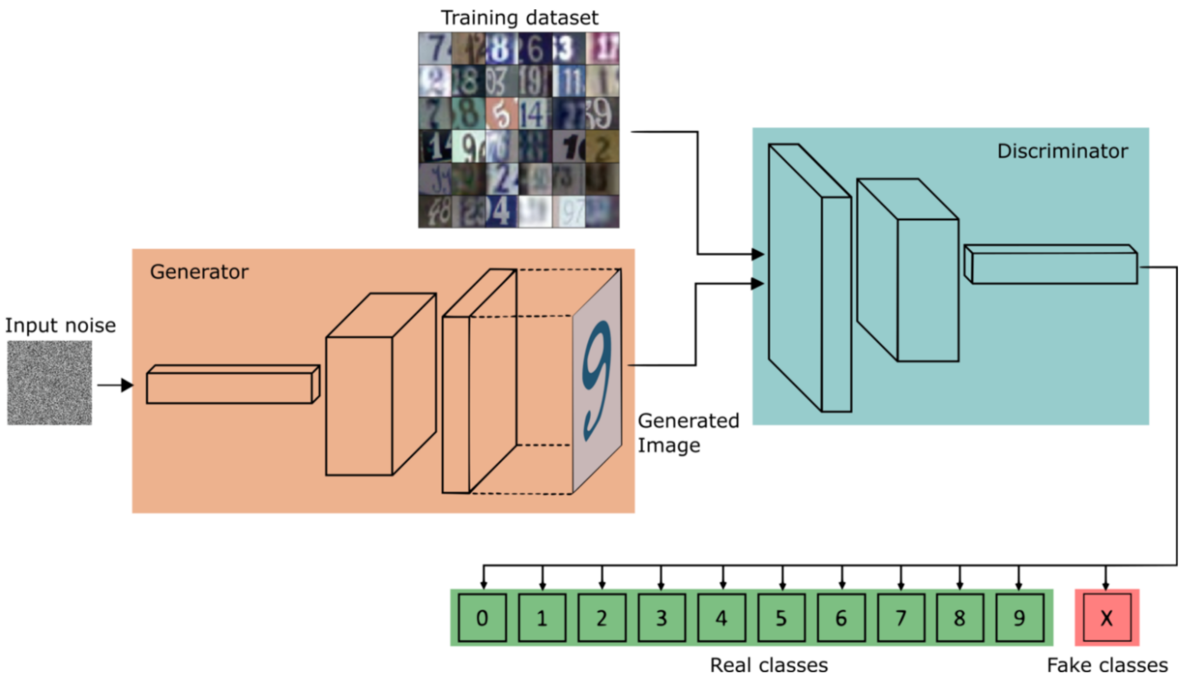


Figure 3.9: Semi-Supervised Classification using GANs [10]

In traditional GANs, we discard the discriminator after the generator is trained properly.

So that, we can generate image using the model, but in case of using this semi-supervised approach the classifier (discriminator) will be our final model.

3.5 Generative Adversarial Networks (GANs) for Image Segmentation:

A generative adversarial network contains two major neural networks — a Generator and a Discriminator. The functionality of GANs in image segmentation is a little bit different than other GANs models. The in-depth architecture and the functionality of GANs in terms of image segmentation are stated below.

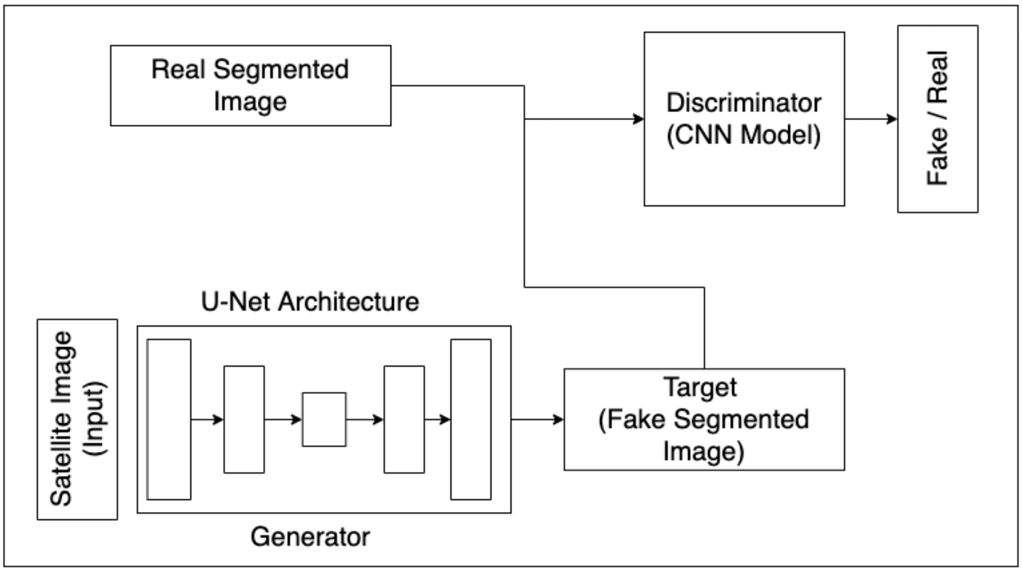


Figure 3.10: Architecture of GANs model for Image Segmentation

Discriminator: Discriminator in general GANs model distinguishes between real or fake image. The real image comes from the original dataset and the fake image is the image generated by the generator. Initially, the discriminator is trained on real and fake samples and while training the generator the parameter of the discriminator is untrainable. The architecture of the discriminator contains a convolutional neural network that uses the PatchGAN method for making a decision. In a general CNN model, the output of the neural networks is binary. But In PatchGAN, the model performs classification for each patch and generates a grid of decisions. By averaging each value the final output becomes a decision whether the image is real or fake.

Generator: The generator of the GANs model is not simple as the discriminator. The generator follows U-Net architecture which is created with encoder-decoder. The encoder does the downsampling of the input image whereas the decoder is used for upsampling the image. The model also contains a bottle-neck layer and uses skip connections. The sole purpose of the generator is to take a source image as input and generate a target image. The loss of the generator model contains two phases — the adversarial loss and the L1 loss. The adversarial loss is updated while the discriminator predicts the fake generated image as a real image. On the contrary, the L1 loss is the mean absolute error between the generated image and the targeted image. By averaging these two losses the generator is trained. After training for a decent amount of iterations the generator can output a segmented image for any particular object, which in our case is the river path from the satellite image.

Chapter 4

Data Collection and Preprocessing

In this section, we have described all steps regarding data collection and pre-processing. As we are working with two different case study, we have devided the workflow in six different subsection. First five subsection describes data collection and pre-processing for explosion detection, classification and fire object detection. And the last subsection contains detail guide about the use of EarthExplorer and ArcGIS that has been used for creating segmented satellite images.

4.1 Explosion Detection – Supervised Approach

In the supervised approach, to train the deep learning model in order to predict explosions, videos were collected from sources like Kaggle [38] (Theft detection dataset), Youtube and DailyMotion. Approximately, 205 explosions and 150 non-explosion videos were collected from these sources.

While collecting videos from youtube and daily motion, keywords like "explosion videos", "real life explosions", "fire on buildings", "gas station explosions" etc. were used.

Among these videos, some were of length five minutes, some were of one hour containing multiple explosion scenes, so after analyzing each video manually, important (which contains the explosion part) sections were selected and length of the videos was shortened in 5 seconds time duration. As each video taken was 30 fps (frame per seconds) and length was 5 seconds, a total of 150 frames were extracted from a single video. To avoid repetition of similar frames we hand-picked useful 15 to 25 frames that particularly contained explosions.

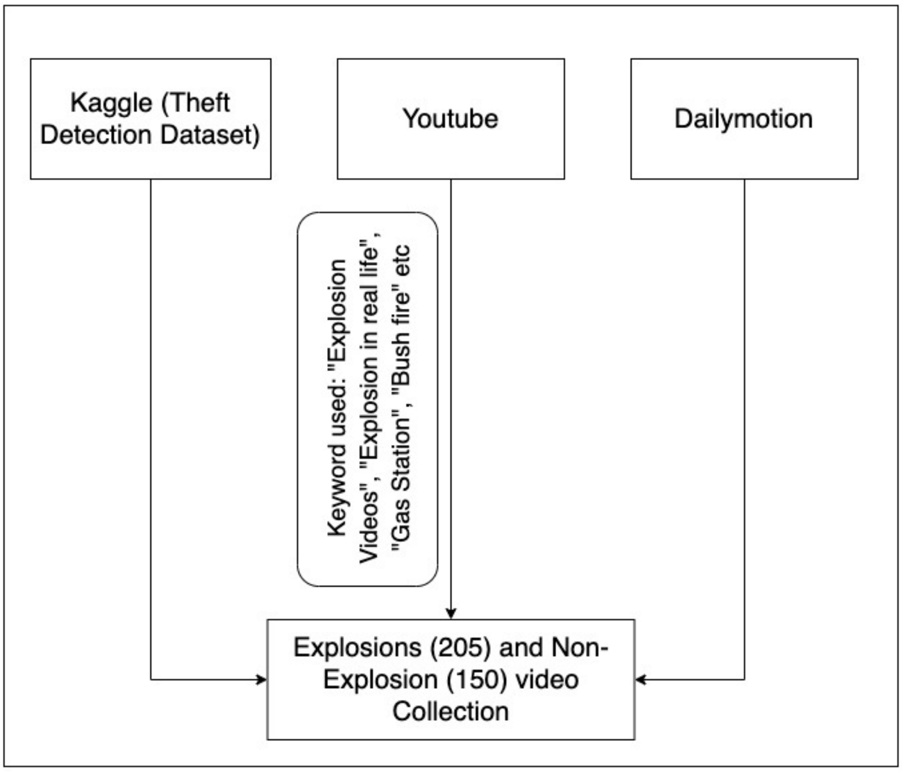


Figure 4.1: Video collection using several mediums

Approximately 3599 images of the explosion were collected and resized by using the mentioned procedure. 70% of these images were used as training images and 30% were used for testing.

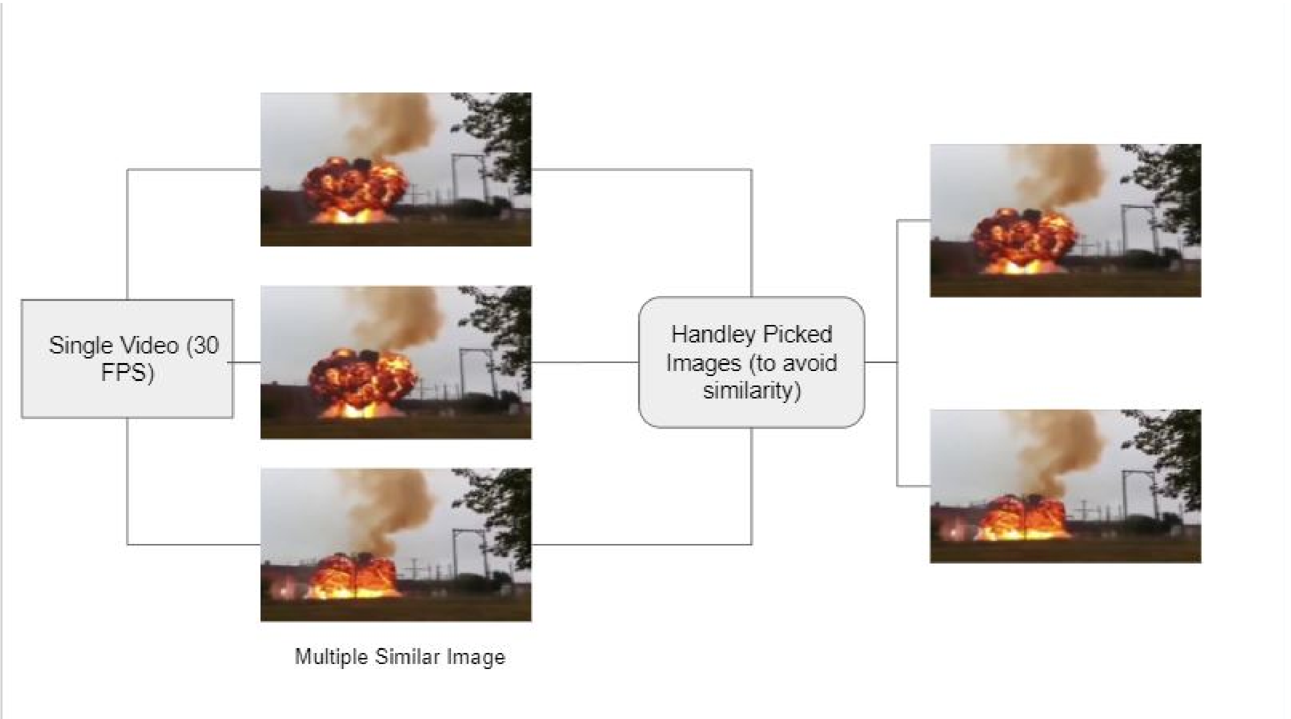


Figure 4.2: Extract useful frame to avoid similarity

For non-explosion, similar mediums were used to collect 150 videos and length were decreased in 5 seconds. As these were also 30 fps videos, from a single video nearly 150 frames were extracted and randomly 15 to 25 frames were selected as sample data. Eventually a total of 3715 images were selected as non-explosion dataset. Then 70% of the non-explosion images were used to train the model and 30% were used to test the model performance.

4.2 Explosion Detection – Semi-supervised Approach

In the semi-supervised approach, we trained Generative Adversarial Networks (GANs) for which labeled images(explosion, non-explosion) and unlabelled images were used.

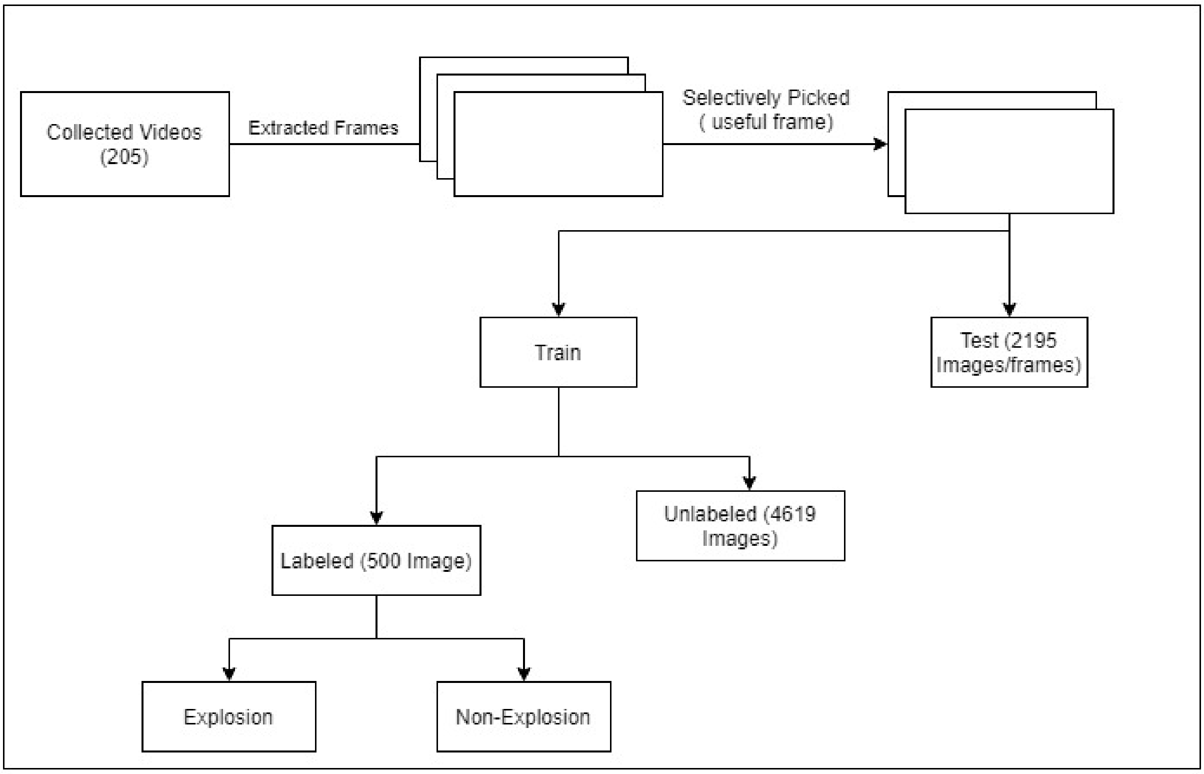


Figure 4.3: Data Collection (Semi-supervised approach)

In the process of creating dataset for generative model, after selecting useful frames from the videos, they were splitted in 70-30 formation which means, 5119 images were used in training and the rest of 2195 images were used for testing purposes. Among the training images, 500 were labeled which means 250 images were specified as explosion and other 250 were specified as non-explosion and 4619 were unlabeled where both explosion and non-explosion images were blended.

4.3 Explosion Classification – Supervised Approach

In the previous section, the dataset was created for detecting explosion in real-time. In addition to that, the challenge taken is whether the model can classifies the type of explosions. In order to do that, we created three major class - Electricity Explosion, House Explosion, Vehicle Explosion along with a “No explosion” class.

The training dataset contains around 1,987 images that has 458 Electrical Explosion Images, 323 House explosion Images, 506 Vehicle explosion Images and 700 non-explosion Images. The detailed number of images in the each class inn given in the table below.

Class Name (Explosion)	**Train**	**Test**
Electrical	458	115
House	323	81
Vehicle	506	127
Non-explosion	700	175

Table 4.1: Number of images used in each class for training and testing the model (Supervised Approach)

4.4 Explosion Classification – Semi-supervised Approach

For semi-supervised approach, the goal was to classify explosion in real-time using generative adversarial network. As the approach is semi-supervised, the number of labelled data is reduced. In the training only 125 images of each class (Electrical Explosion, House Explosion, Vehicle Explosion, Non-explosion) has been used. In total, only 500 labeled images has been used for training while the other 1239 images were unlabelled.

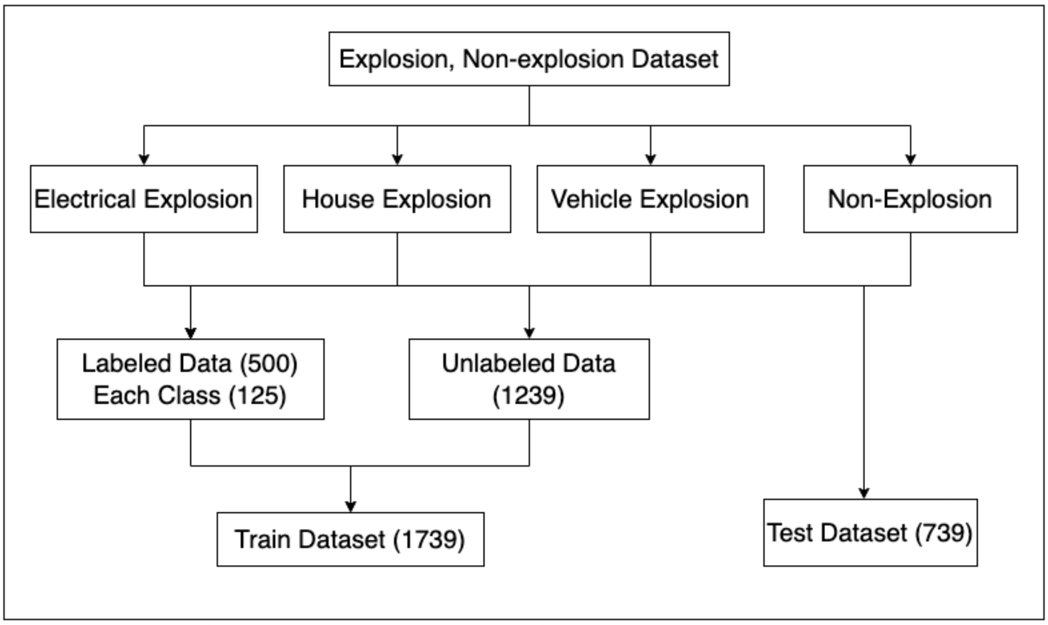


Figure 4.4: Data Collection for Explosion Classification(Semi-supervised approach)

4.5 Object Detection

To detect the exact position of fire after an explosion by analysing real time video, an object detection model was trained which required a different set of dataset. In order to prepare the dataset a total number of 335 images which contains fires and flames was taken and annotated using an annotation software named "LabelImg".

Figure 4.5: Data Annotation using LableImg (Fire Object Detection)- Sample-01

Figure 4.6: Data Annotation using LableImg (Fire Object Detection)- Sample-02

After creating the annotated dataset, YOLOv5 (You Only Look Once) models were used to train on the dataset and overall 70% accuracy were obtained.

4.6 Satellite Image Segmentation

In our second case study, our sole focus is to segment river paths from satellite images. To train the Generative model (pix2pix), we developed a dataset that contains both satellite images where river path is segmented and not segmented. To develop the dataset two software were used are — EarthExplorer and other is ArcGIS.

At first the EarthExplorer website was used to —

- Locate the river in the satellite image of the world,
- Make a boundary of a certain area where the river could be found,
- Extract the water body from the chosen area and
- Download the waterbody data with satellite image

EarthExplorer walkthrough for Image Segmenetation:

(i). Go to: https://earthexplorer.usgs.gov/ and find an area that contains rivers

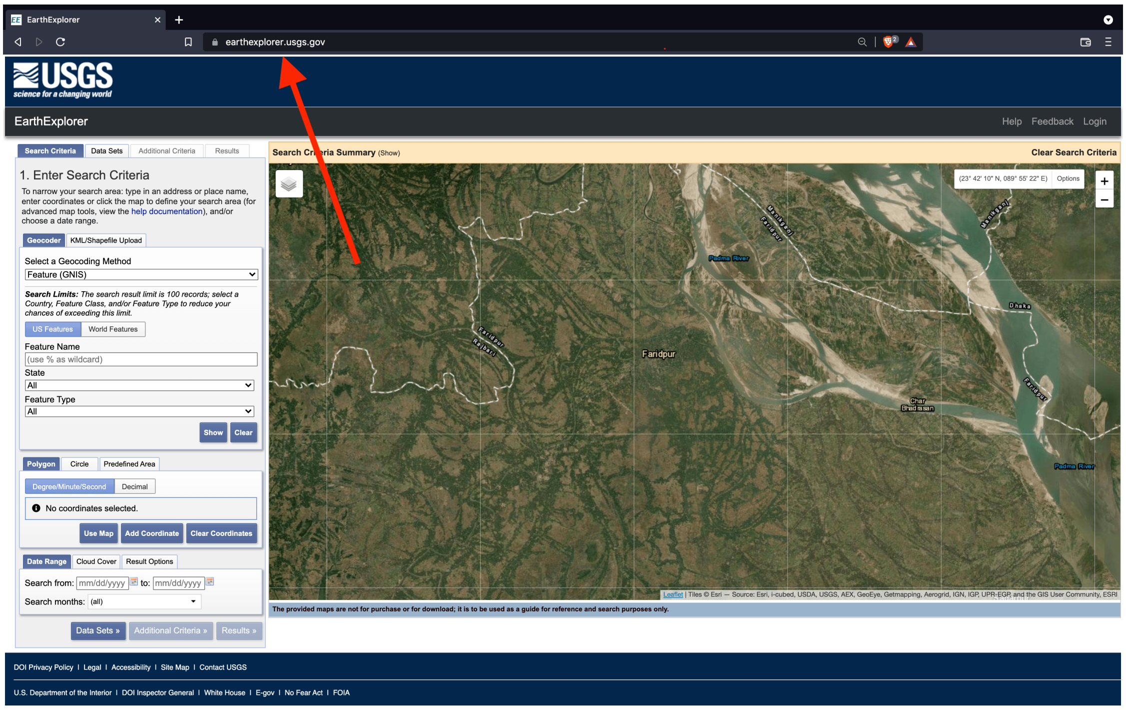


Figure 4.7: EarthExplorer Guide For Image Segmentation: Step-01

(ii). Right-click to put the boundary on the river portion

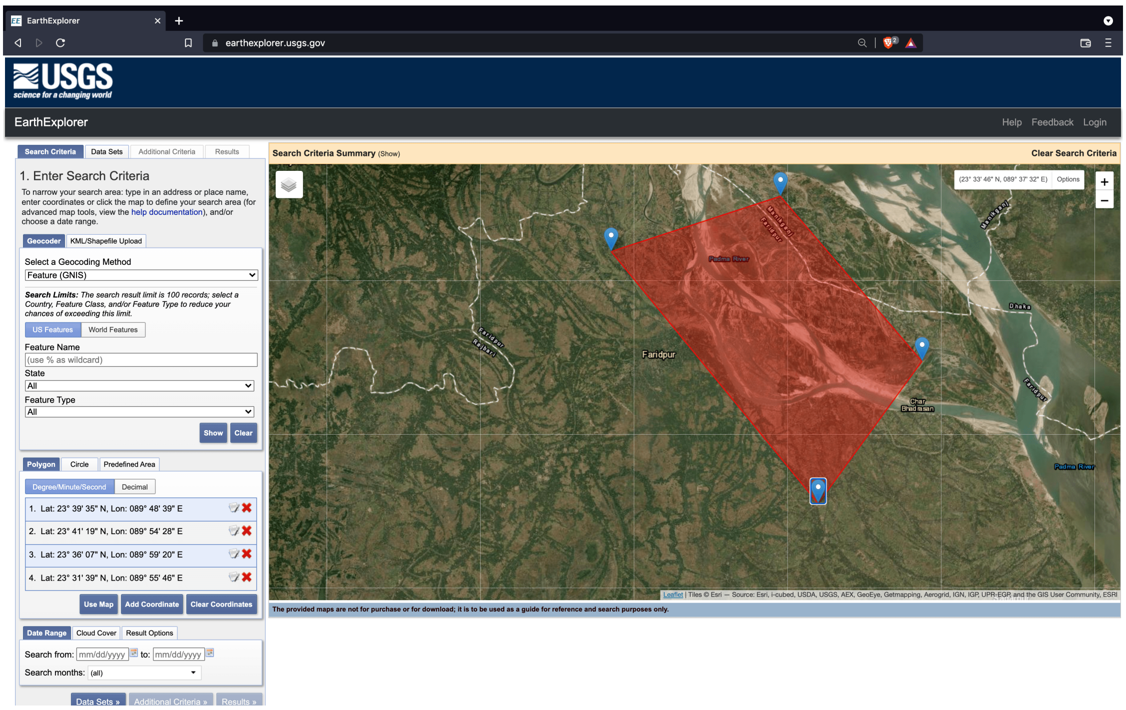


Figure 4.8: EarthExplorer Guide For Image Segmentation: Step-02

(iii). Click on the **Data Sets** tab, Expand **Digital Elevation**, Expand **SRTM** and check the "SRTM waterbody data" box

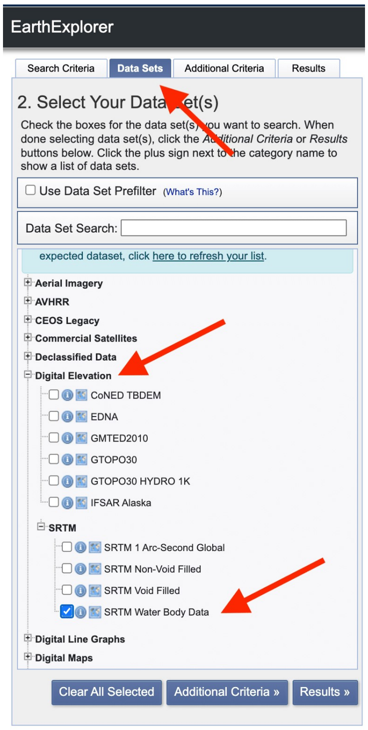


Figure 4.9: EarthExplorer Guide For Image Segmentation: Step-03

(iv). Click on the **Results** tab and click on the download indicator (Right side of show metadata)

Figure 4.10: EarthExplorer Guide For Image Segmentation: Step-04

(v). When the download box pops up. Click the **Download** button

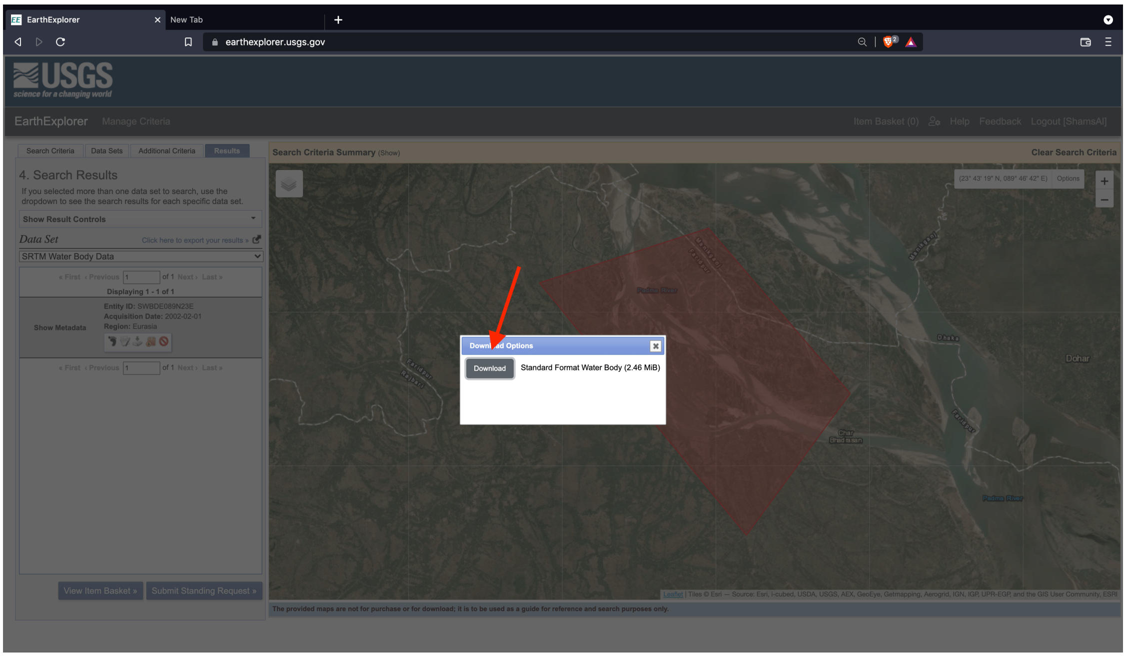


Figure 4.11: EarthExplorer Guide For Image Segmentation: Step-05

(vi). A zip file with a unique code name will be downloaded

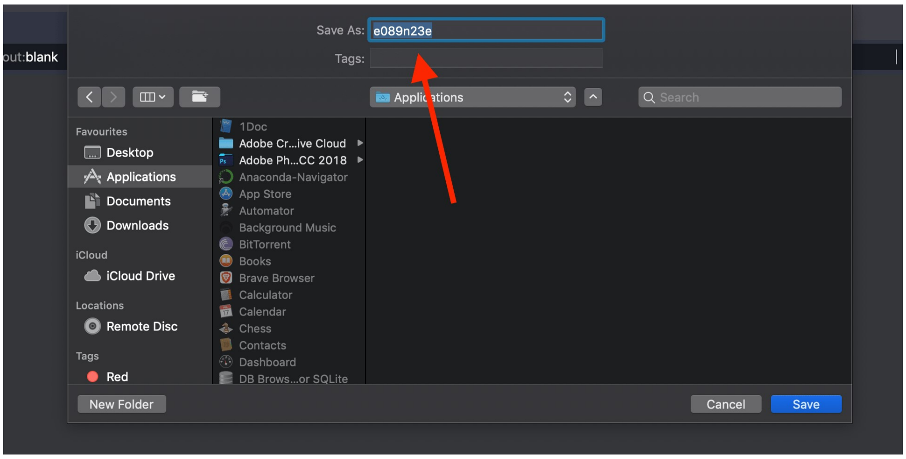


Figure 4.12: EarthExplorer Guide For Image Segmentation: Step-06

The downloaded zip file from EarthExploer contains a ".shp" file that only can be opened and visualized by the software "ArcGIS". So in the second major step, the ".shp" file was opened using ArcGIS.

ArcGIS walkthrough for Image Segmenetation:

(i). Open ArcGIS software and from the right panel click on “Connect to Folder” which will provide options to navigate to the “.shp” file

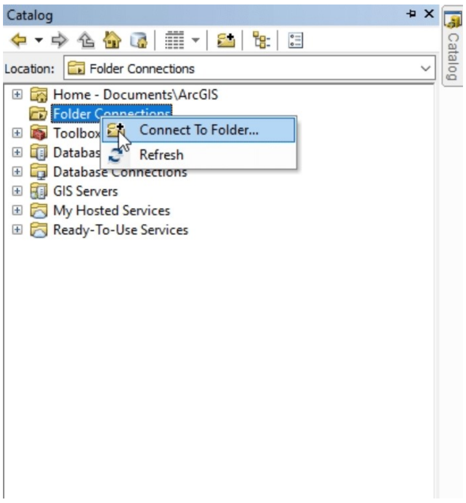


Figure 4.13: ArcGIS Guide For Image Segmentation: Step-01

(ii). From the upper panel, click on “Add Basemap”

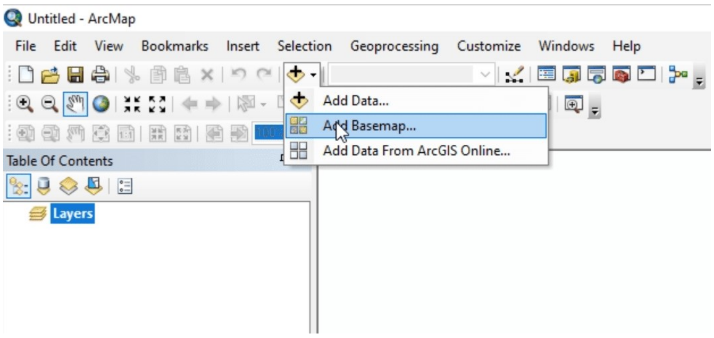


Figure 4.14: ArcGIS Guide For Image Segmentation: Step-02

(iii). Select "Imagery with Labels"

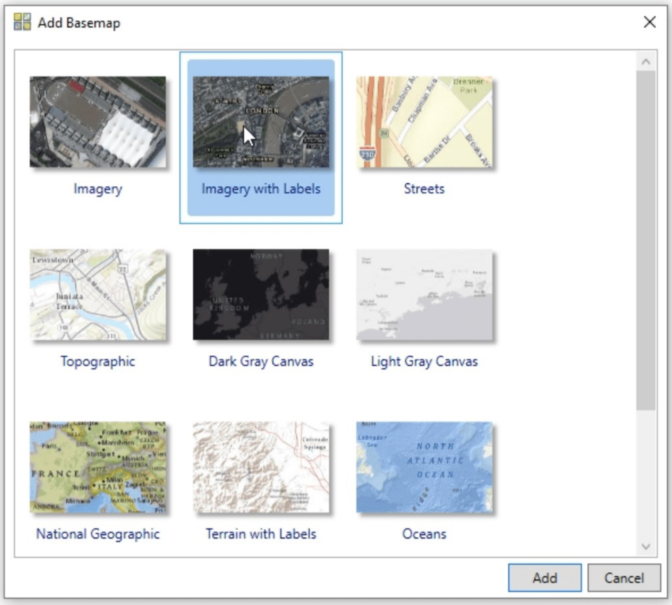


Figure 4.15: ArcGIS Guide For Image Segmentation: Step-03

(iv). By clicking on "Add" from the upper panel, navigate to the ".shp" file from the pop-up window and add the ".shp" file

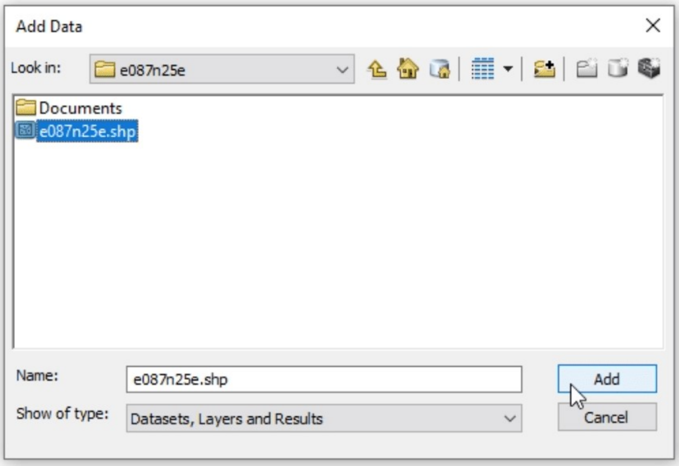


Figure 4.16: ArcGIS Guide For Image Segmentation: Step-04

(v). Go to "Arc Toolbox", Expand "Projection and Transformations" and select "Define Projection"

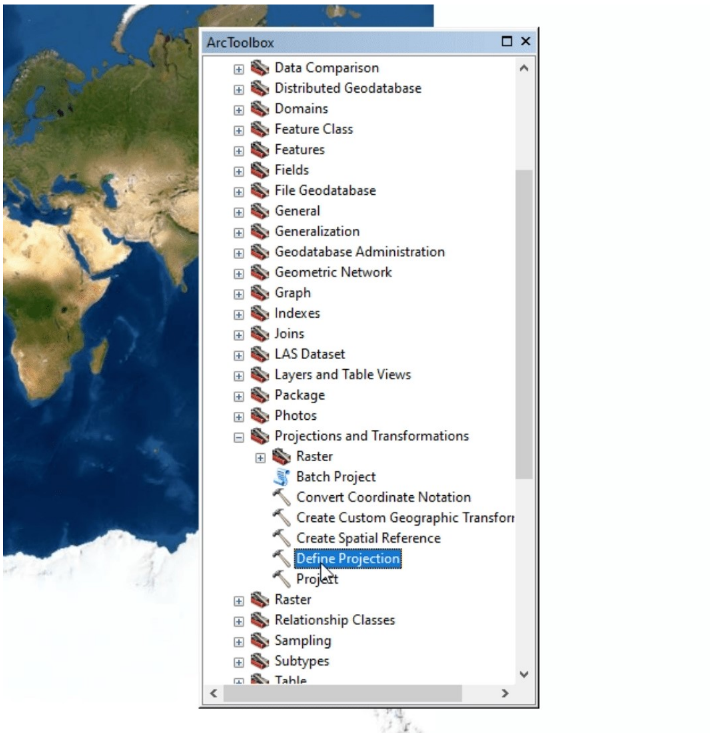


Figure 4.17: ArcGIS Guide For Image Segmentation: Step-05

(vi). When the pop-up window opens, click on the right button of "Coordinate System"

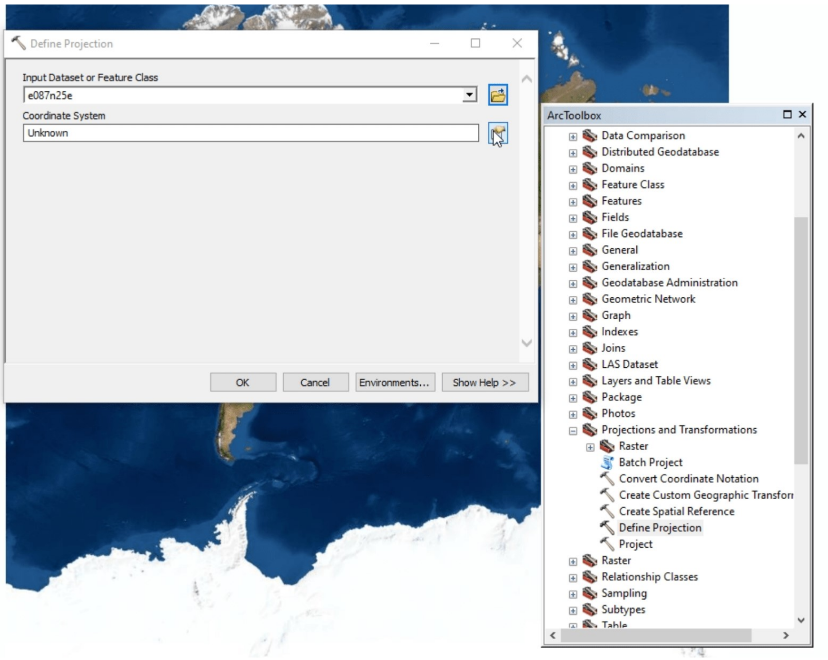


Figure 4.18: ArcGIS Guide For Image Segmentation: Step-06

(vii). On the follow-up pop-up, expand "Geographic Coordinate System"

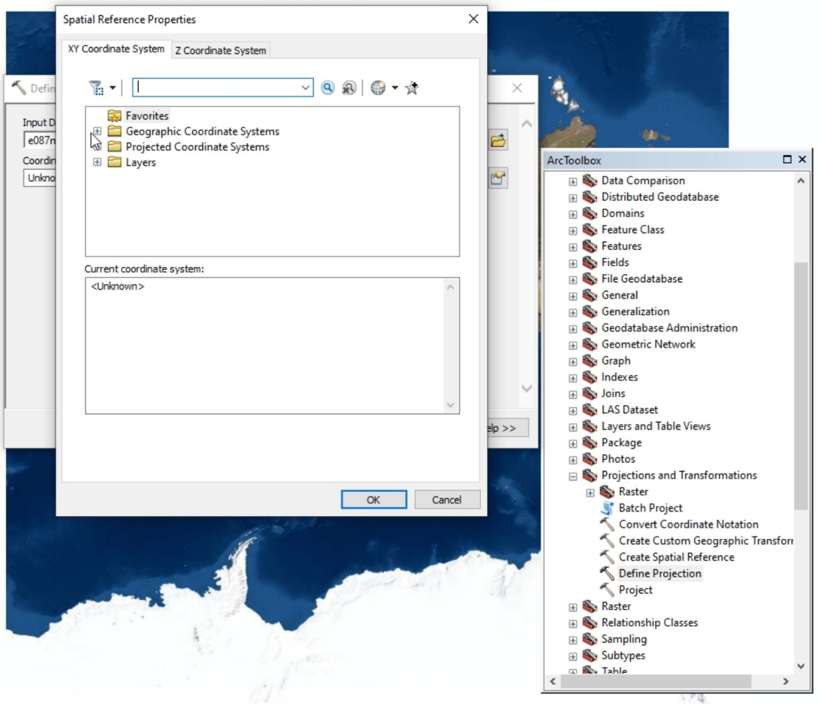


Figure 4.19: ArcGIS Guide For Image Segmentation: Step-07

(viii). Under the Asia region, Select "Everest Bangladesh" and press "ok"

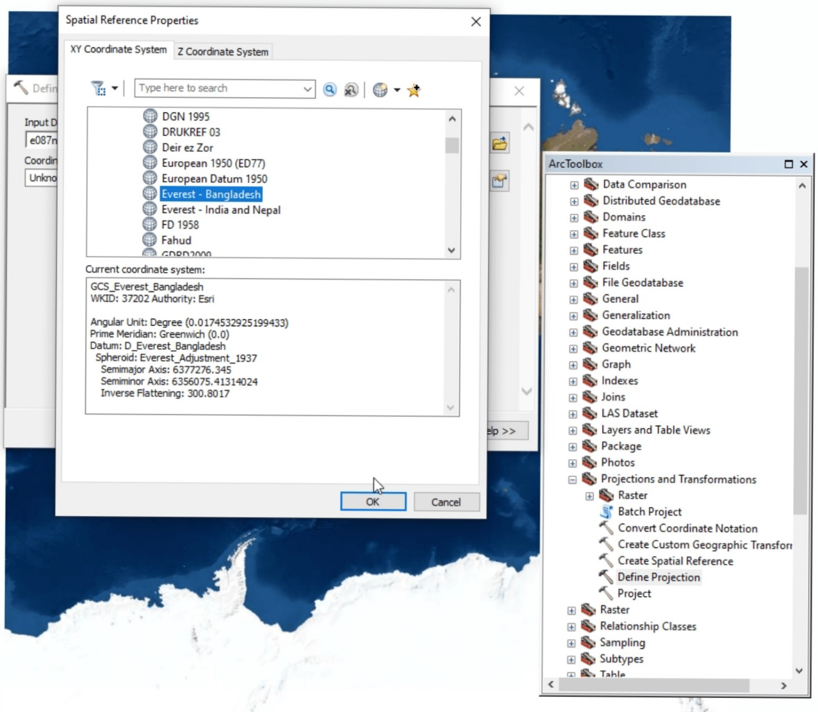


Figure 4.20: ArcGIS Guide For Image Segmentation: Step-08

(ix). When "Define Projection" pops up on the bottom right, choose the color from the color pallet

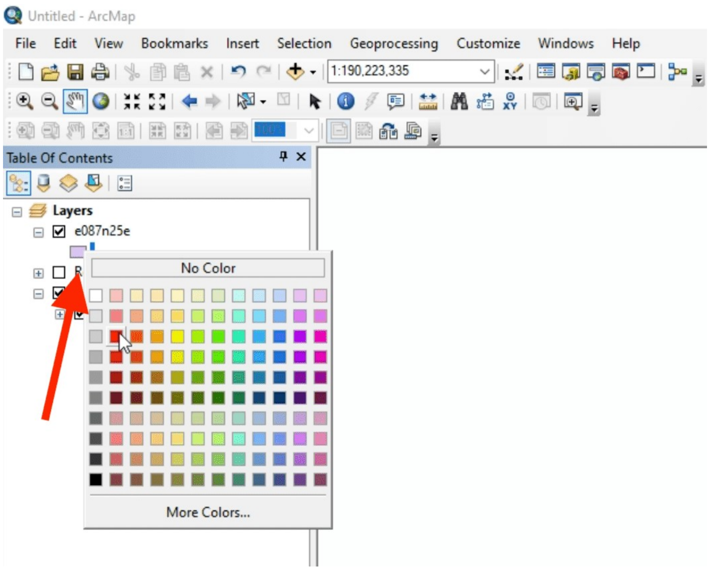


Figure 4.21: ArcGIS Guide For Image Segmentation: Step-09

(x). Right-click on the file name from the left panel and select "Zoom to Layer"

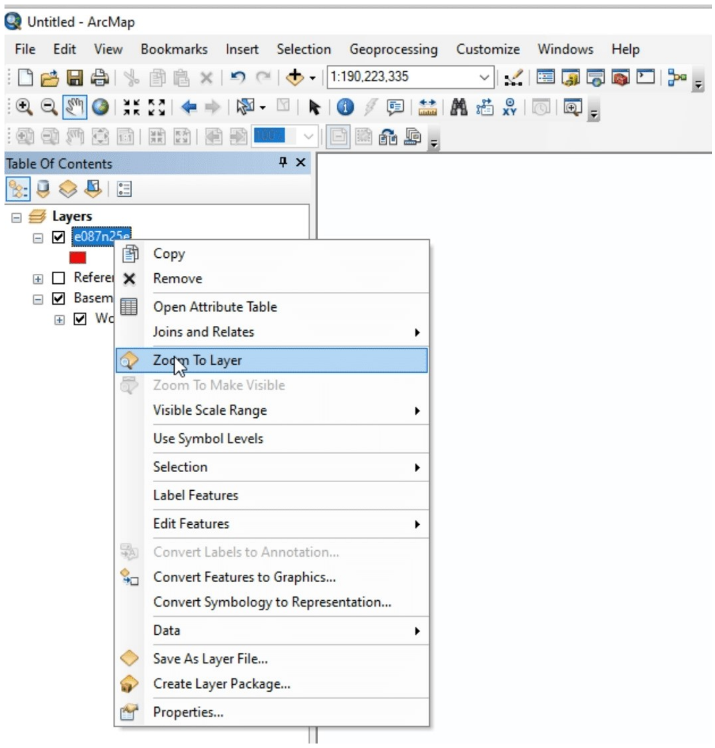


Figure 4.22: ArcGIS Guide For Image Segmentation: Step-10

(xi). Now choose the segmented area, export, and save the file in ".png" format

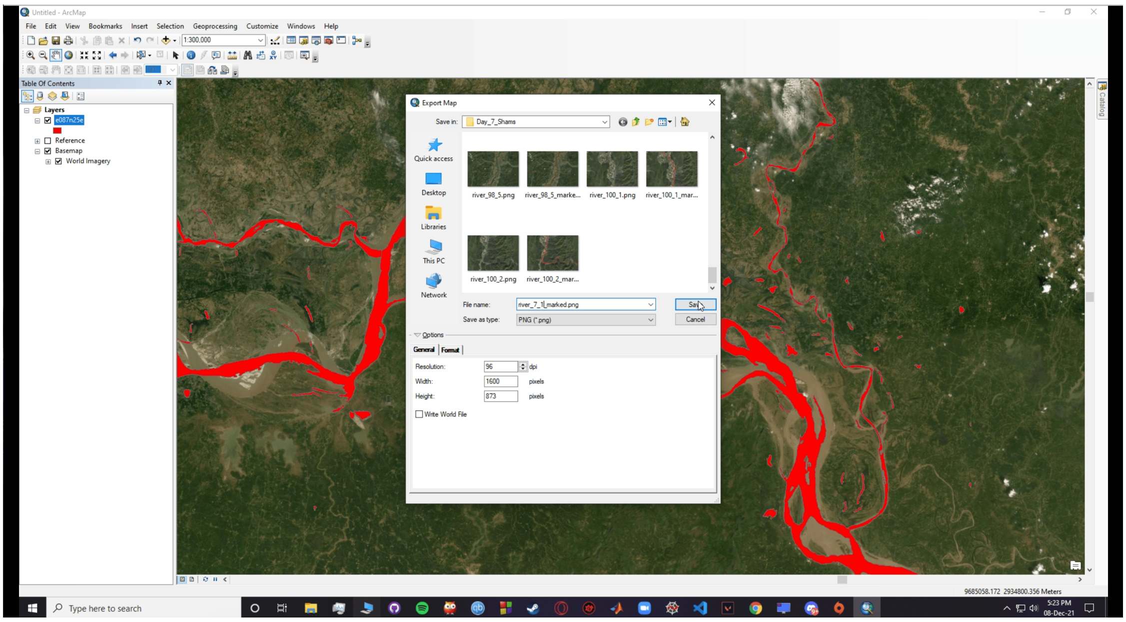


Figure 4.23: ArcGIS Guide For Image Segmentation: Step-11

(xii). By unchecking the file name box on the left panel, the unsegmented image will appear, which also need to be exported and save in ".png" format

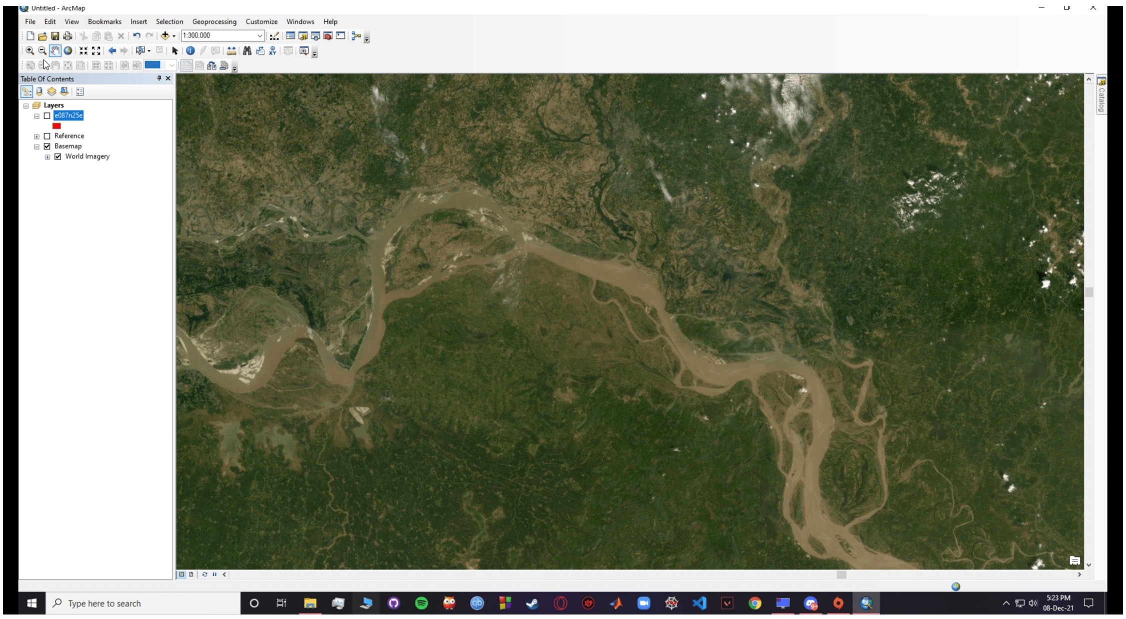


Figure 4.24: ArcGIS Guide For Image Segmentation: Step-12

By following these steps, an image of the original satellite image and segmented satellite image can be achieved. We kept all the images (both original satellite image and segmented image) in a folder and using python we merged the satellite image with the corresponding segmented satellite image. The dimension of the merged image was 600x1200. All the merged images were then used as the dataset for the pix2pix GANs model.

4.7 Data Sample for Classification and Segmentation

For the purpose of our experiment (both for explosion classification and satellite image segmentation), two separate dataset have been created. For explosion detection, we have collected explosion related videos from social platforms such as youtube, daily motion, etc. After collecting videos we have hand-picked 12-25 frames from each video and thus, the dataset has been created.

Figure 4.25: Data Sample for Explosion Detection from CCTV footage-01

Figure 4.26: Data Sample for Explosion Detection from CCTV footage-02

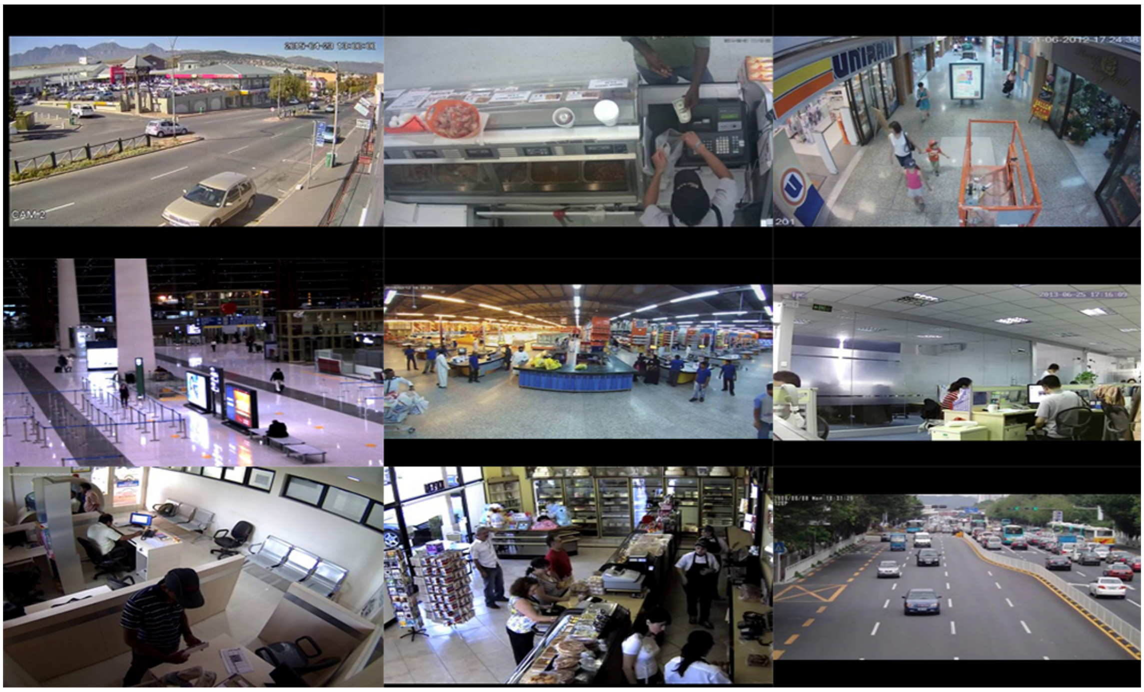


Figure 4.27: Sample of non-explosion images for Explosion Detection

In explosion classification, our goal was to predict the type of explosion. Thus the dataset was divided into three major classes. Electrical or Industrial Explosion, House Explosion (in-door and out-door) and Vehicle Explosion.

Figure 4.28: Data sample for explosion classification (top row - explosion caused for electrical wiring, middle row - vehicle explosion, bottom row - house explosion)

For segmenting river path from satellite image, we can create a dataset using the EarthExplorer and ArcGIS. We have taken the satellite image from EarthExplorer and then we segmented water-body using ArcGIS. We generated over four hundred and fifty samples and trained with pix2pix model (C-GANs).

Each image used for segmentation is concated version of both satellite image and segmented satellite image. the left portion of the image contains original satellite image and the right portion of the image contains river path segmented version of that particular satellite image.

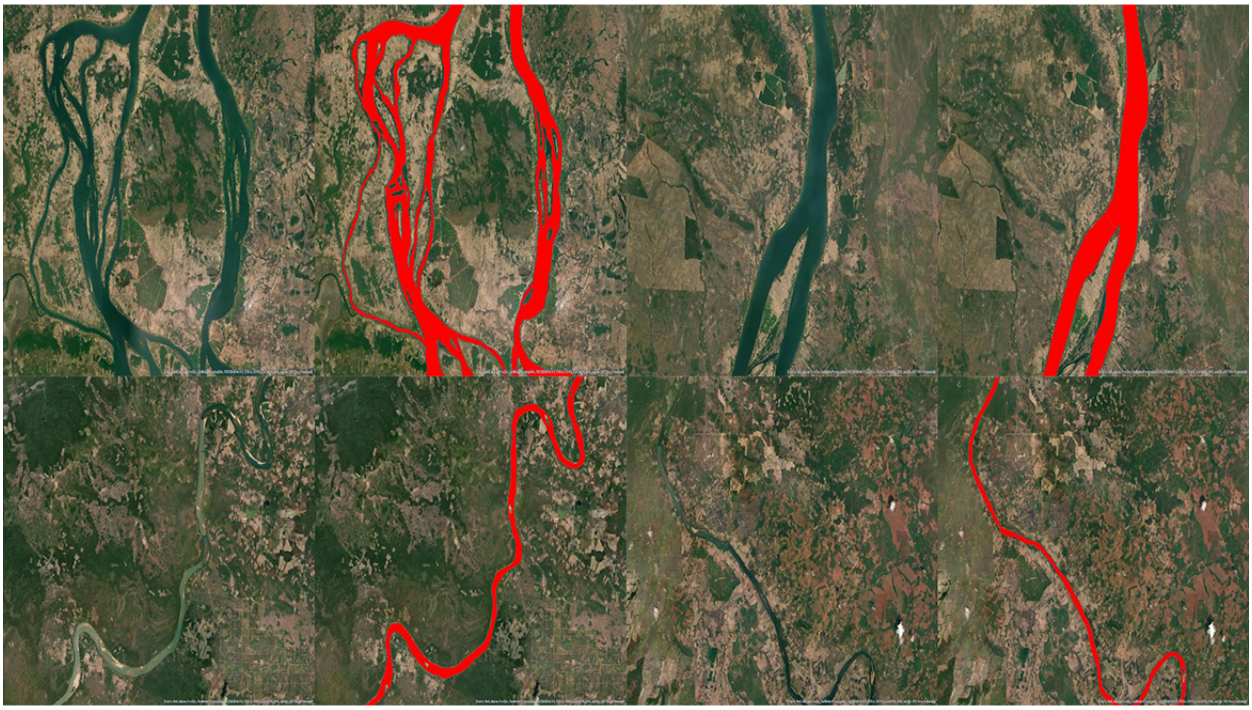


Figure 4.29: Data samples for river path segmentation from satellite images-01

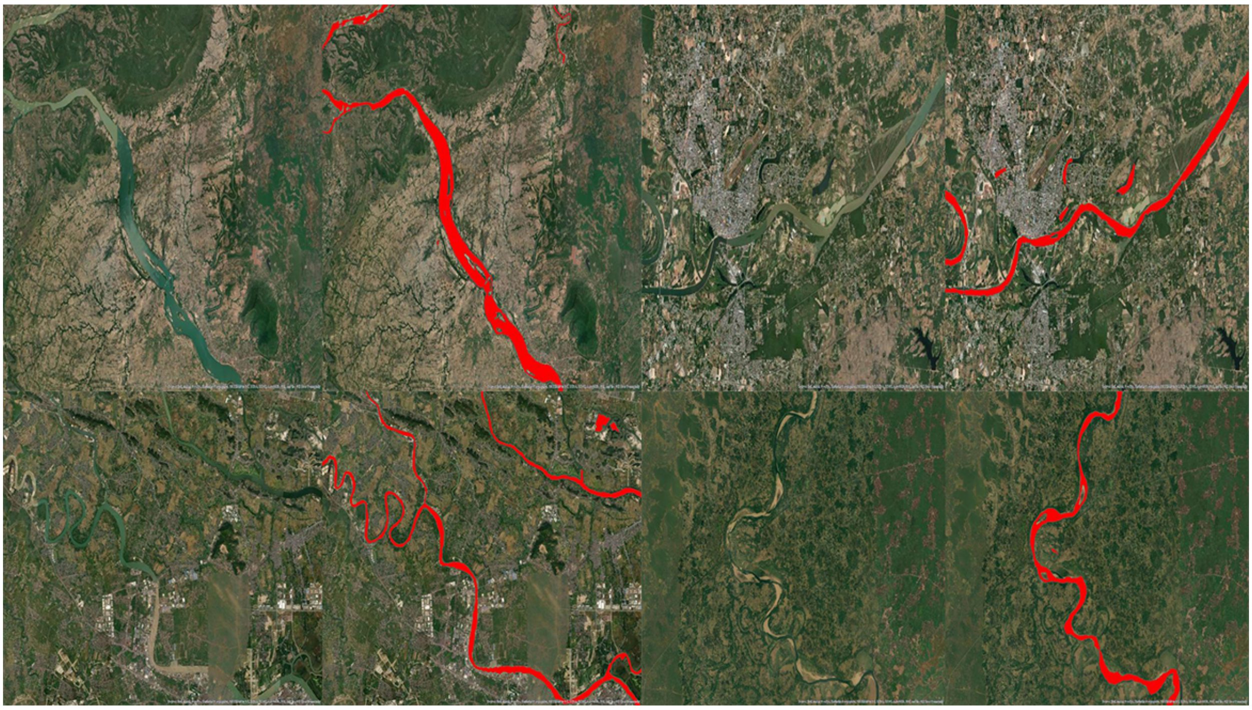


Figure 4.30: Data samples for river path segmentation from satellite images-02

Chapter 5

Model Training and Hyperparameter Tuning

In this section, the core architecture of each model has been described with proper figures. As both of our case studies are computer vision-oriented, deep learning models such as CNN, CNN-LSTM, CNN-GRU, Generative Adversarial Networks (GANs) have been used to train and test the dataset. We have also described the functionality of each model along with optimizers, loss function, and other parameters.

5.1 Convolutional Neural Network (CNN)

For image classification, convolutional neural nets are well known and effective as they extract and analyze the features in each layer. For the experiment, a deep convolutional model was designed that contains over 9.7 million trainable parameters. Four convolutional layers each followed by a 2x2 (two by two) max-pooling layer were used in the model along with three dense layers in which the last layer had two softmax outputs.

The model was trained several times using different optimisers such as “adam optimiser”, “RMS Prop”, “SGD” and “ReLU” activation function was used in each convolutional and FC layer. After training for 45 epochs the model obtained the highest accuracy of 99.82% using “Adam” optimiser.

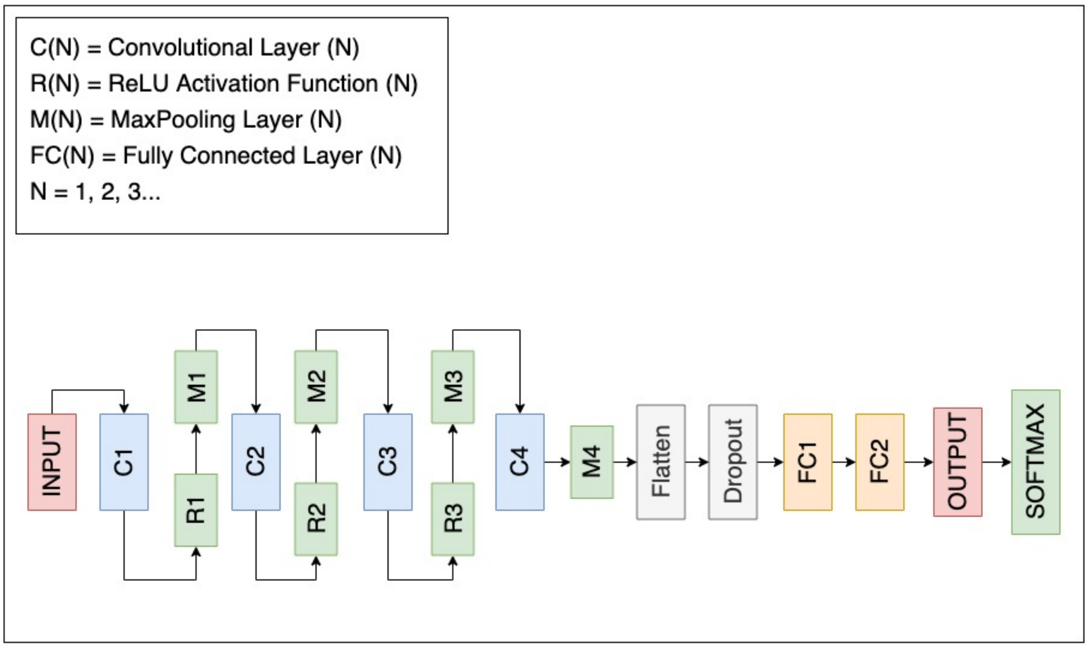


Figure 5.1: Customised CNN Layer

5.2 Pre-trained VGG16

VGG16 was first introduced in 2014 and became famous instantly because of its 92.7% accuracy on ImageNet [39]. This model contains 16 layers in which 13 layers are convolutional and 3 layers are fully connected. The model has over 14.7 million trainable parameters and was trained on ImageNet that has over 15 million high resolution images which belong to 22 thousands different categories.

In the experiment of classifying explosion and non-explosion images VGG16 pre-trained model was used where 13 convolution layers selected from the VGG16 model and 3 fully connected dense layers were customised according to the dataset.

As several optimisers were used to train the model for the explosion dataset, among them, 99.91% test accuracy was obtained by using "Adam" optimiser with 45 epochs of training.

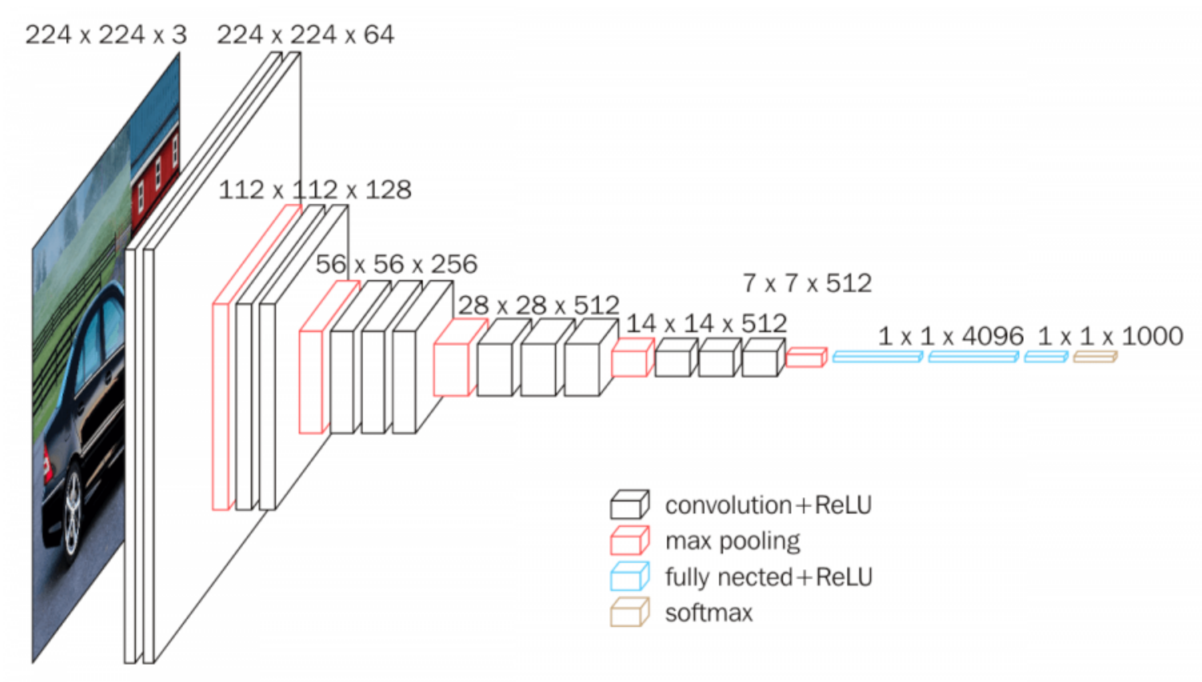


Figure 5.2: General VGG16 Architecture [8]

5.3 Recurrent Neural Network (LSTM)

Recurrent Neural Networks are suitable for working with sequential data, mostly used in terms of working with textual data. In the RNN, the output from the previous layer is also fed to the current layer and it can maintain its own memory or state which is used for later prediction.

In the experiment, a custom hybrid model was built which used both convolutional layers along with the LSTM (Long Short Term Memory) layer. The input image was 224 x 224 x 3 which then was passed by three convolutional layers each followed by a 2x2 max-pooling layer. Then the output was reshaped by using the reshape layer of keras and passed through a LSTM layer that had 64 output units. Three dense (Fully Connected) layers were used at the end where the later one had the softmax output that classified the image.

Total 1,12,322 parameters were trained for the classification and 91.20% test accuracy were obtained by using RMS Prop as optimiser and running the model for 45 epoch.

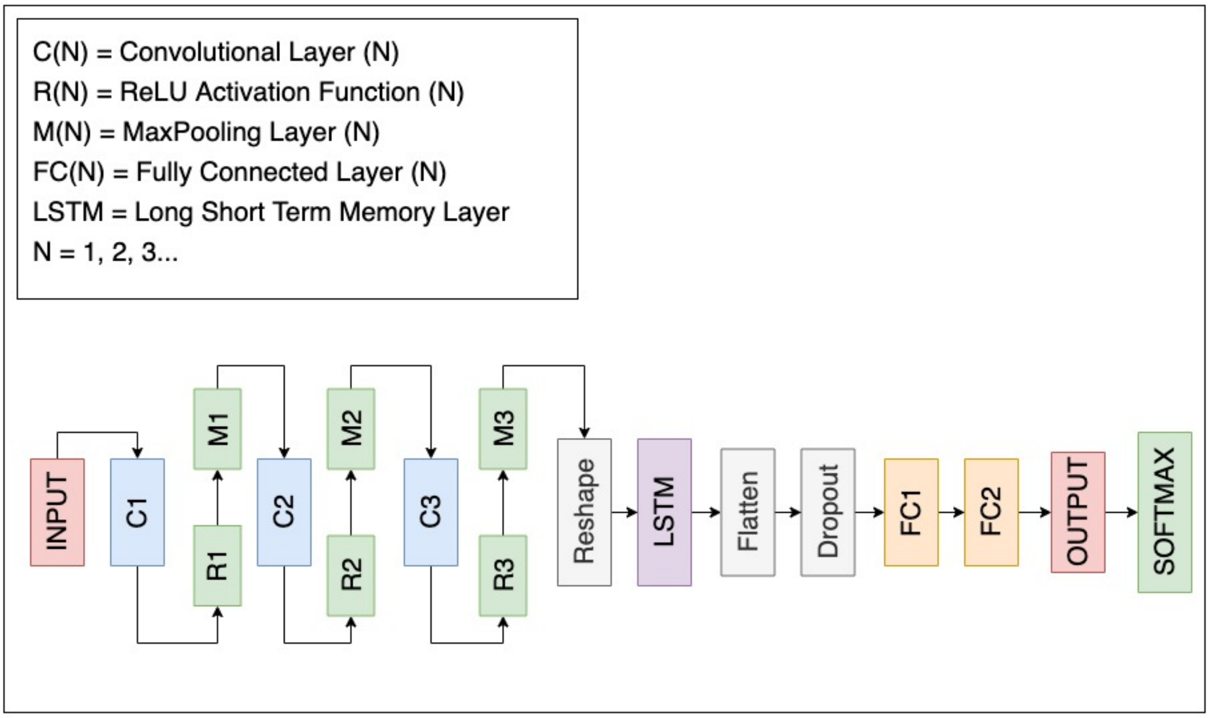


Figure 5.3: Hybrid CNN-LSTM Architecture

5.4 Recurrent Neural Network (GRU)

For further experimentation in classifying explosion and non-explosion images, another hybrid model of CNN and RNN was established which contains GRU units instead of LSTM units.

The model takes an 224 x 224 x 3 input size image and passes through three convolutional layers using the "ReLU" activation function. Then after reshaping the output of the last layer, it is sent as an input of the GRU layer. Three fully connected layers were used eventually where the later one outputs the label.

The highest test accuracy of 98.36% was found by using 45 epochs and "RMS Prop" optimiser which was far better than the CNN-LSTM model.

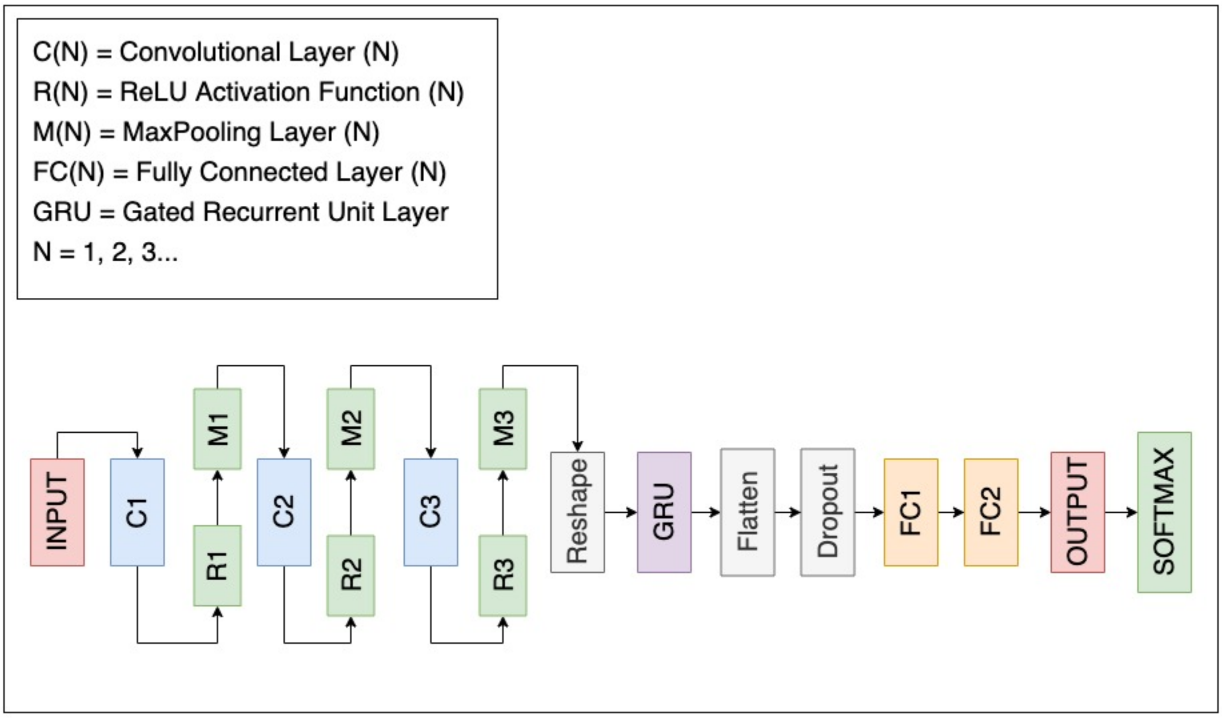


Figure 5.4: Hybrid CNN-GRU Architecture

5.5 Semi-Supervised Approach (GANs)

Generative Adversarial Network (GANs) contains two artificial neural nets inside of it. One is known as a generator and another is a discriminator. First, a random noise generator generates an image which is sent to the discriminator and on the other hand, labeled data (image) is sent to the discriminator. The discriminator then tries to differentiate between two images provided to it. The job of the generator is to create such an image so that the discriminator cannot distinguish between the two images given to it. Eventually when the discriminator is fooled by the generator which means when the discriminator cannot differentiate between generated and labeled image then the training is done and the generator model is used for generating similar images.

In the experiment of classifying explosion and non-explosion images, GANs were used differently than most other models. As the approach is semi-supervised, 500 labeled images were sent to the discriminator along with the unlabeled 4619 images which contain both explosion and non-explosion images. So, the job of the discriminator is not only to distinguish the real and fake (generated by generator) images but also to learn which image belongs to which class (explosion or non-explosion). After training for several epochs the discriminator model is used for classifying the desired classes.

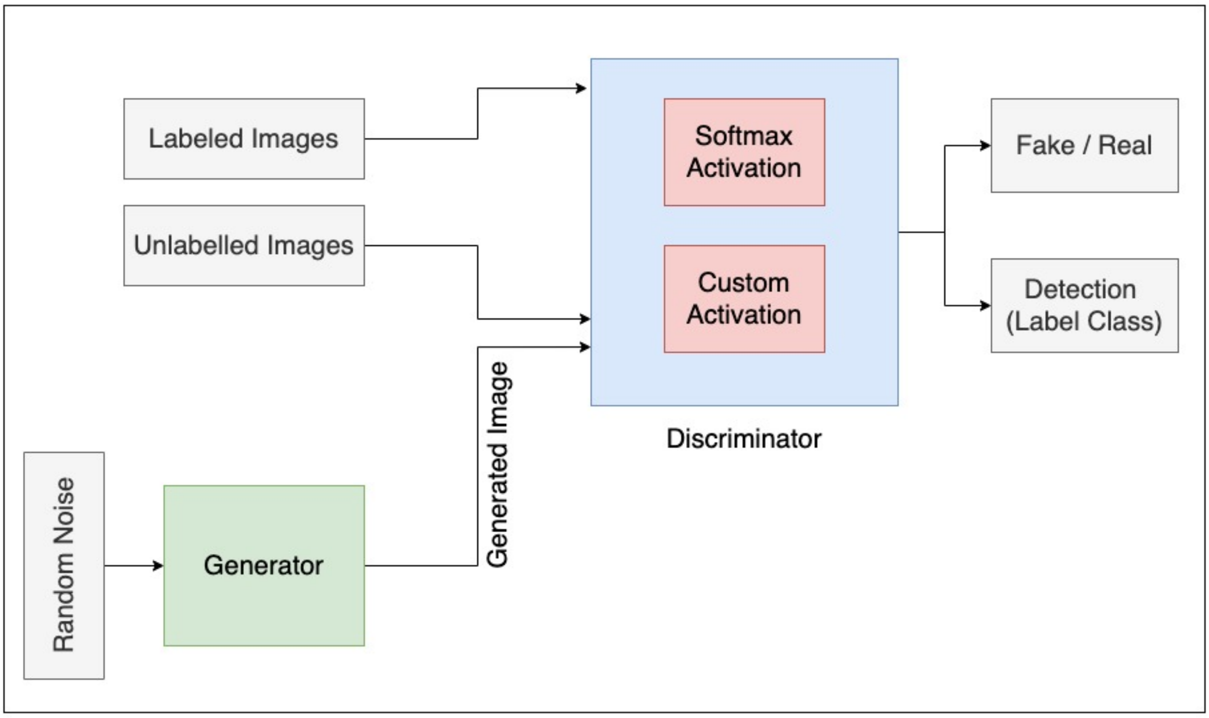


Figure 5.5: Architecture of Generative Adversarial Network for Explosion Detection

Generator:

The generator takes a random noise as input which is of dimension z (2400). Initially the random noise passes through a dense layer which is of dimension 256x7x7. Then using the reshape layer of keras the dimension is converted to 7x7x256. After reshaping, up-sampling is done by using Conv2DTranspose layer which converts the 7x7 image to 14x14 image. Three consecutive Conv2DTranspose layers each followed by BatchNormalization and LeakyReLU activation up-sampled the image and generated an image of dimension 112x112x3.

This generated image is then passed as an input of discriminator which then classifies if the given image is explosion or non-explosion and if the image generated is real or fake.

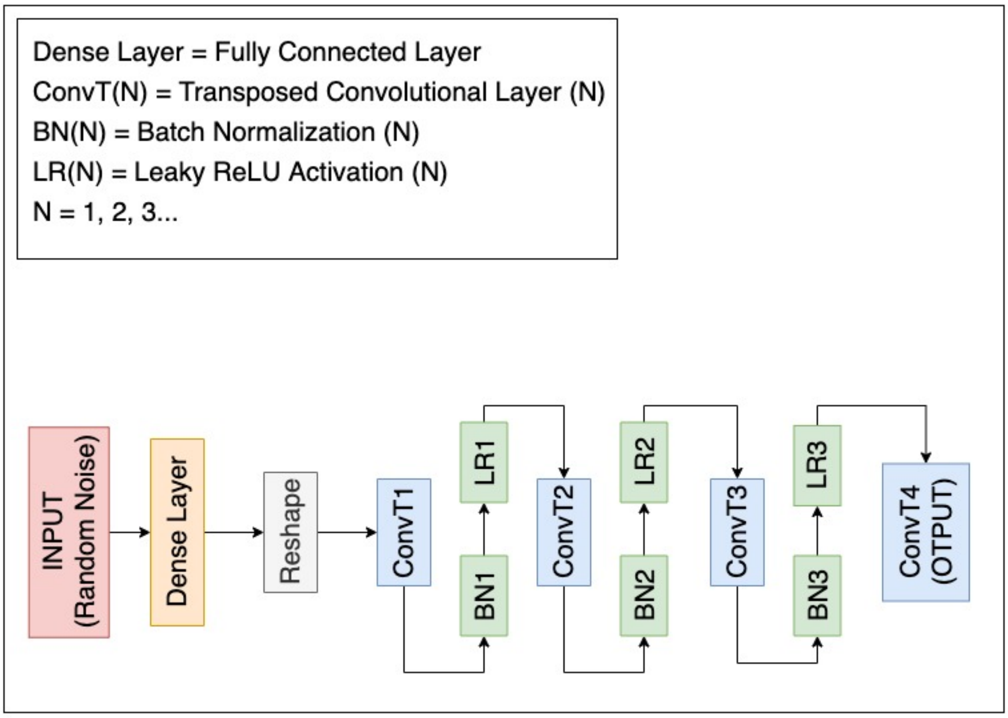


Figure 5.6: Architecture of Generator Model

Discriminator:

The discriminator takes the input generated by the generator along with the labeled and unlabelled images. Its job is not only to distinguish the real or fake image but also to classify them.

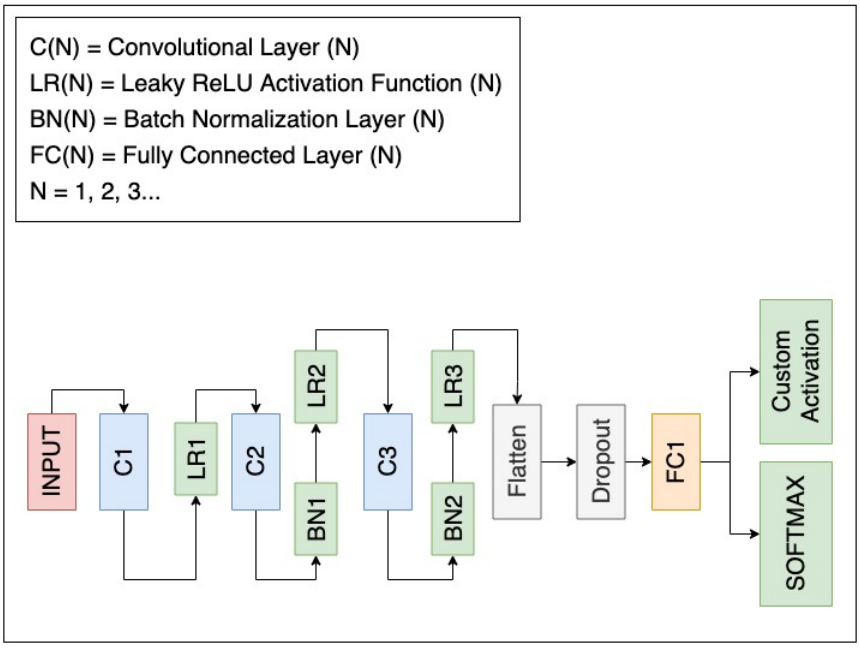


Figure 5.7: Architecture of Discriminator Model

The discriminator has three convolutional layers along with “LeakyReLU” activation and batch normalization. The end dense layer has two different kind of outputs, one is a softmax output which is for labelling the class properly and another uses a custom activation function that makes the decision if the image passed to the discriminator is real or fake.

5.6 GANs Architecture for Satellite Image Segmentation:

The pix2pix model contains two neural networks that are — generator and discriminator. It also has a composite model where the generator and the discriminator are connected. In segmentation, the satellite image is the source image and the segmented satellite image is the target image.

Initially, the generator takes the source image and produces a sample of the target domain. The discriminator is trained by using these fake segmented satellite images along with the true segmented image from the dataset. After the training, the discriminator learns to distinguish between the fake and real segmented image. While training the discriminator model, “binary_crossentropy” was used as the loss function.

After training the discriminator model, the generator model is trained by using the composite model. The real source image has been sent as an input of the generator model and the generator outputs a target image (segmented satellite image). This generated image is sent to the discriminator along with the true segmented satellite image by concatenating. The discriminator uses the PatchGAN [40] method to map both of the images and measures the loss between them. This loss is known as “L1 loss” or “Mean Absolute Error”. Each time the discriminator predicts “real” for the generated image the adversarial loss is also minimized. So, the generator is trained with the average of both “adversarial” loss and “L1 loss”.

Generator: The generator model is created with an encoder-decoder. It also uses the U-Net architecture for implementing skip connection. The encoder block contains a Conv2D layer followed by a Batch Normalisation layer (Conditional) and a LeakyReLU layer. The decoder contains a Conv2DTranspose layer followed by a Batch Normalisation layer and drop out layer (conditional). It uses ReLU as an activation function. The generator contains seven consequence encoder blocks, a bottleneck layer with no batch normalization layer, and seven consequence decoder blocks. Eventually, the generator model outputs a generated 256 x 256 x 3 image.

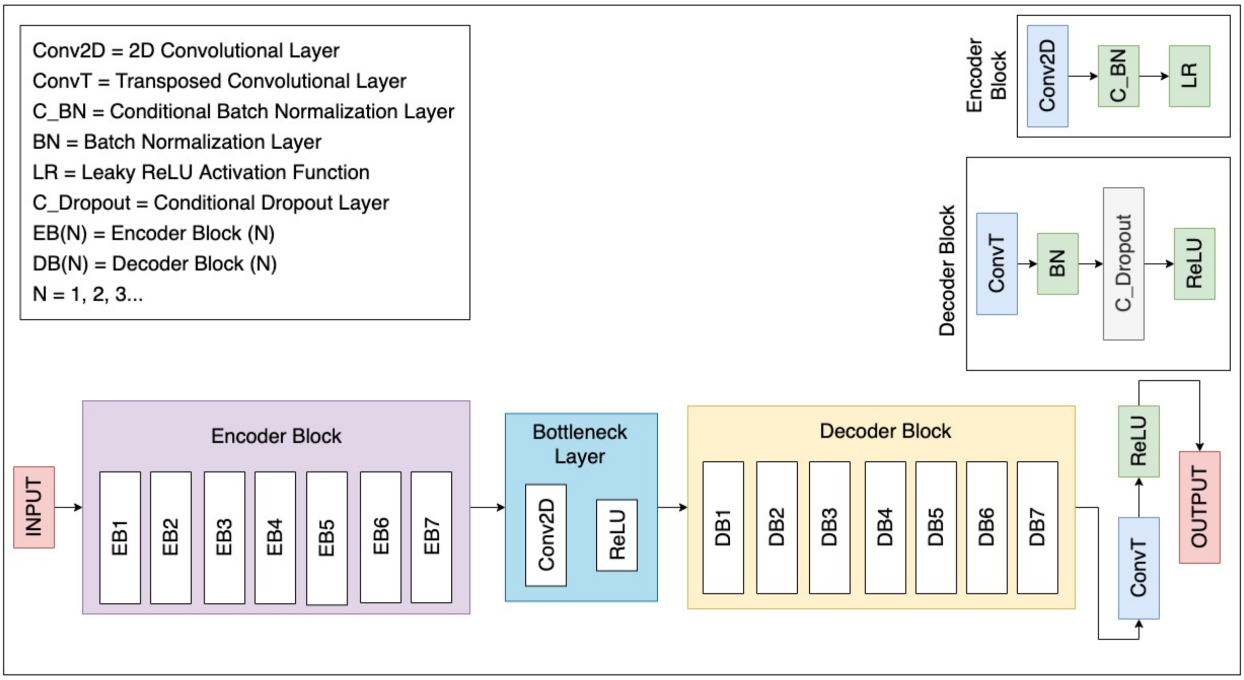


Figure 5.8: Architecture of Generator Model for Image Segmentation

Discriminator: The discriminator model has five Conv2D layers conditionally followed by the Batch Normalisation layer. It uses "LeakyReLU" as an activation layer. The final layer is a patch output layer with a sigmoid activation function. The model use "Adam" as an optimizer and uses "binary_crossentropy" as the loss function. The discriminator takes input two images, one is generated image (fake segmented image) and the other is the true segmented image from the dataset. It then concatenates the two images and tries to distinguish between the real and fake using the classification technique.

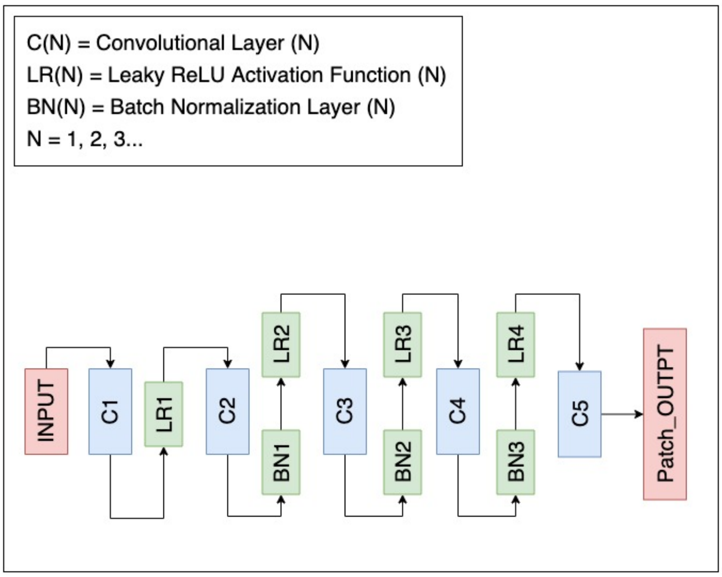


Figure 5.9: Architecture of Discriminator Model for Image Segmentation

5.7 Fire Detection (YOLOv5)

YOLO (You Only Look Once) is a powerful object detection algorithm that was pretrained on COCO [41] dataset which contains over 1.5 million instances. Yolov5 is an extension of YOLO that was written in pytorch.

In the experiment, separate fire detection dataset were generated which then is used for training the YOLOv5 model. The model was trained for 200 epochs using a batch size of 4.

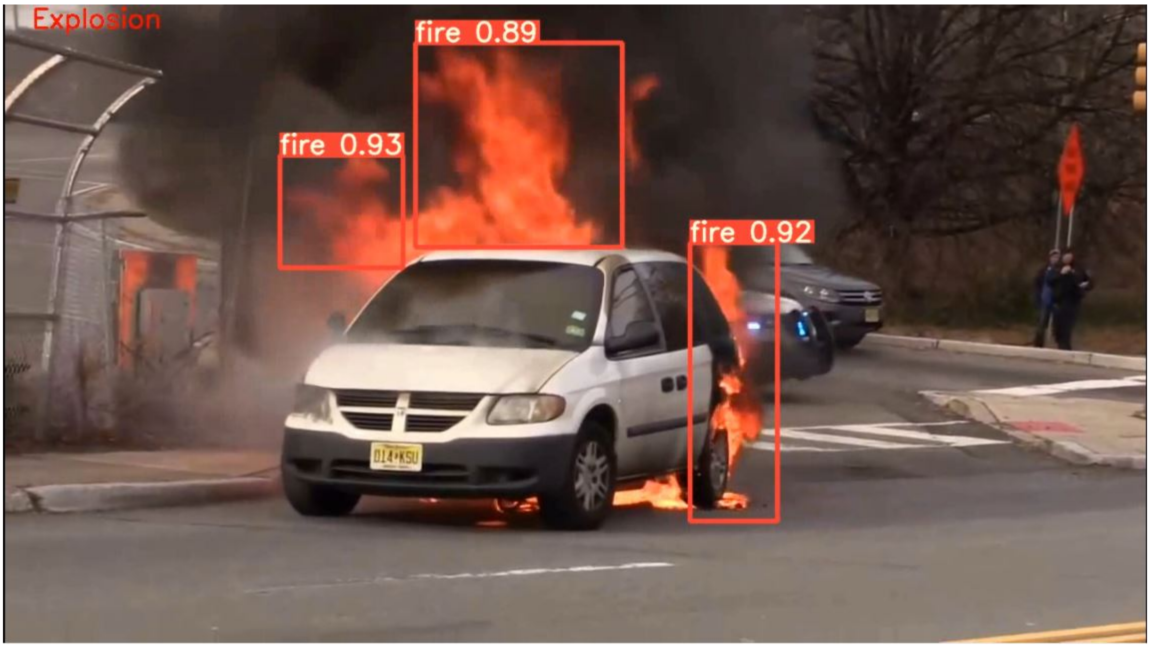


Figure 5.10: Sample Image of Fire Detection using YOLOv5

Chapter 6

Result Analysis

In the result analysis section, we have demonstrated the output of all the models. The section contains four different subsections. The first subsection contains a comparative analysis between deep learning models (supervised and semi-supervised approach) for explosion detection (binary classification). The second subsection contains model accuracies for explosion classification (multi-class). In the third section, we have described the adversarial and L1 loss of the pix2pix GANs model for river-path segmentation from the satellite image, and finally, in the fourth section information about the GPUs that are used for training purposes have been described.

6.1 Explosion Detection — Supervised and Semi-Supervised

Our thesis contains two different experiments — classification and image segmentation. Thus the result analysis have two different segments. In the classification, we classified explosion or non-explosion footage from real time CCTV footage. Addition to that, we classified the categories of explosion. Currently we only categorised three major classes — House Explosion, Vehicle Explosion, Electrical Explosion.

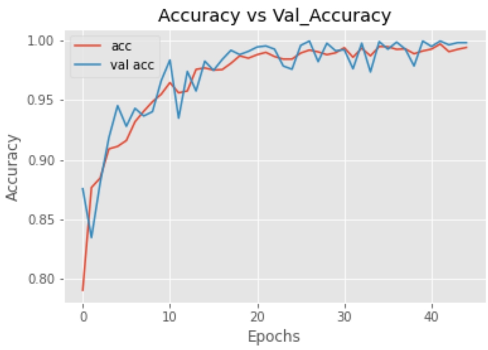
Explosion Detection (Supervised Approach)

In our experiment, over 7 thousands (7,000) explosion and non-explosion images were collected. After preprocessing data, several deep learning models such as Convolutional Neural Net- work, Recurrent Neural Network along with Hybrid model of CNN, LSTM (Long Short Term Memory), GRU (Gated Recurrent Unit) were used for training and testing.

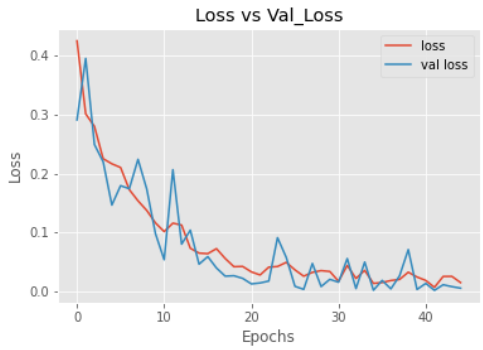
After training these supervised models, highest 99.82% test accuracy were obtained in CNN using the “adam” optimizer and 45 epochs. Among the CNN-LSTM, CNN-GRU hybrid mod-

Table 6.1: Train and Test Accuracy of Supervised Models. (Explosion Detection)

Model Name	Labeled Data		Test Acc.			Train Acc.		
	Explosion	Non-Explosion	Adam	RMS Prop	SGD	Adam	RMS Prop	SGD
CNN	3599	3715	**99.82**	99.13	99.27	99.24	95.65	97.38
VGG16	3599	3715	**99.91**	97.27	97.27	99.79	99.59	97.11
RNN(LSTM)	3599	3715	87.10	**91.20**	77.16	89.16	92.39	70.53
RNN(GRU)	3599	3715	93.35	**98.36**	76.95	98.99	99.11	73.44



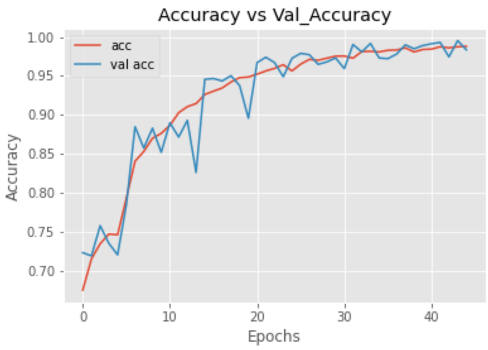
(a) Accuracy



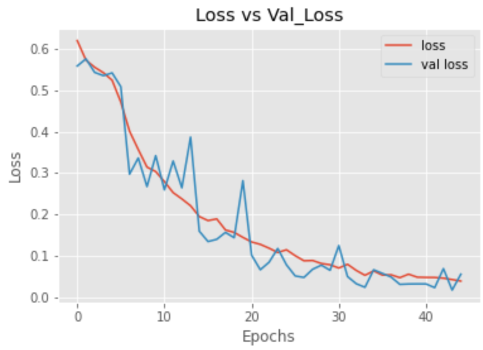
(b) Loss

Figure 6.1: Comparison between Accuracy (a) and Loss (b) of CNN Model

els, highest 98.36% test accuracy were obtained by CNN-GRU which used “RMS Prop” as optimizer and ran for 45 epochs. We also experimented with a popular pre-trained model known as VGG16. The model provided 99.91% test accuracy with “adam” optimizer.

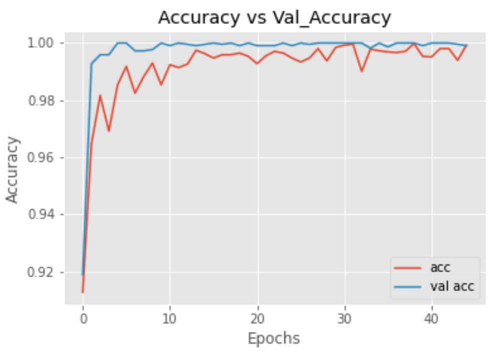


(a) Accuracy

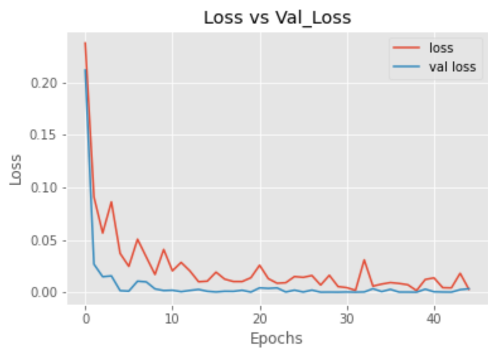


(b) Loss

Figure 6.2: Comparison between Accuracy (a) and Loss (b) of Hybrid CNN-GRU Model



(a) Accuracy



(b) Loss

Figure 6.3: Comparison between Accuracy (a) and Loss (b) of pre-trained VGG16 Model

Explosion Detection (Semi-supervised Approach)

In our Semi-supervised approach, we used Generative Adversarial Network (GAN) specifically designed for classification. The model was trained with only 500 label images (explosion and non-explosion) and 6814 unlabeled images and eventually we obtained 95.81% test accuracy using "adam" optimizer.

Table 6.2: Test Accuracy of Semi Supervised GAN (Explosion Detection)

Labeled Data		**Unlabeled Data**	**Optimizer**	**Test Acc.**
Explosion	Non-Explosion			
250	250	6814	Adam	**95.81**
250	250	6814	RMS Prop	51.04
250	250	6814	SGD	91.98

6.2 Explosion Classification — Supervised and Semi-Supervised

Explosion Classification (Supervised Approach)

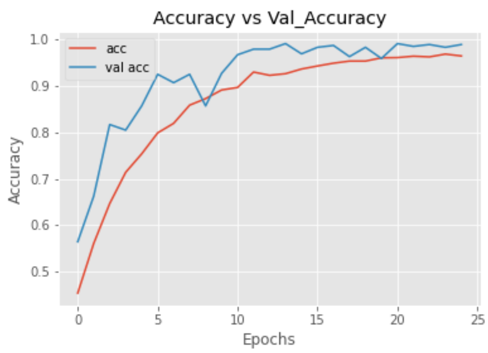
In classification of explosion, we experimented with over two thousands five hundred images. 80% of the total dataset was used for training and the rest of 20% was used for testing purposes.In two methods the classification experiment was performed, supervised approach and semi supervised approach.

In supervised approach, we used convolutional neural network, hybrid models like CNN-GRU (Gated Neural Network), CNN-LSTM (Long Short Term Memory)and pre-trained VGG16.

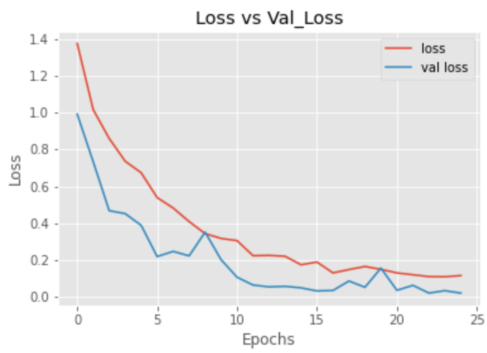
Table 6.3: Train and Test Accuracy of Supervised Models. (Explosion Classification)

Model Name	Test Acc.			Precision			Recall		
	Adam	RMS Prop	SGD	Adam	RMS Prop	SGD	Adam	RMS Prop	SGD
CNN	97.99	99.00	90.36	97.89	99.21	91.36	97.70	99.02	88.52
CNN-LSTM	56.22	75.10	54.02	86.72	81.16	89.53	27.19	63.87	26.63
CNN-GRU	85.54	77.31	53.01	87.84	82.60	90.03	78.02	70.44	26.39
VGG16	99.60	99.80	94.98	99.61	99.80	95.25	99.61	99.80	94.14

After training these supervised models for classifying explosion, highest 99.00% test accuracy were obtained in CNN using the “RMS Prop” optimizer. Among the CNN-LSTM, CNN-GRU hybrid models, highest 85.54% test accuracy were obtained by CNN-GRU which used “adam” as optimizer. We also experimented with a popular pre-trained model known as VGG16. The model provided 99.60% test accuracy with “adam” optimizer.

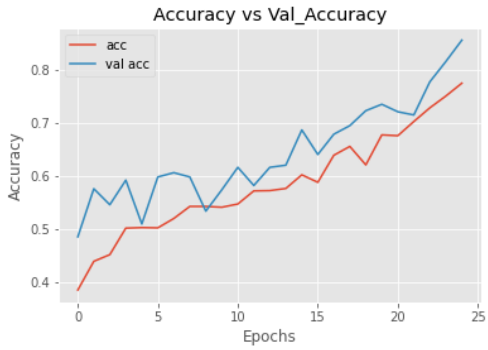


(a) Accuracy

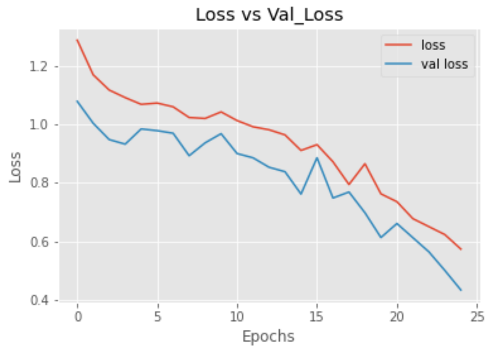


(b) Loss

Figure 6.4: Comparison between Accuracy (a) and Loss (b) of CNN Model (Optimizer - RMS Prop)

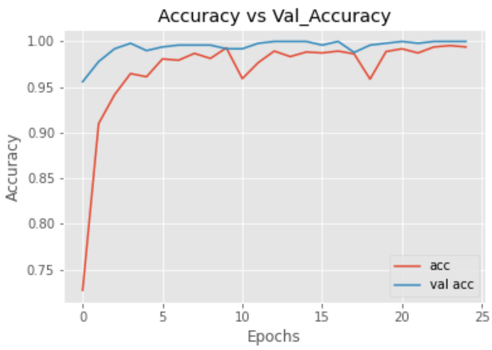


(a) Accuracy

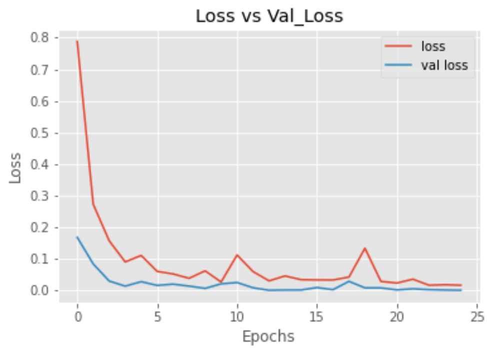


(b) Loss

Figure 6.5: Comparison between Accuracy (a) and Loss (b) of hybrid CNN-GRU Model (Optimizer - Adam)



(a) Accuracy



(b) Loss

Figure 6.6: Comparison between Accuracy (a) and Loss (b) of VGG-16 Model (Optimizer - Adam)

Explosion Classification (Semi-supervised Approach)

In our Semi-supervised approach, we used Generative Adversarial Network (GAN) specifically designed for classification. The model was trained with only 1,739 labeled images where the actual dataset contains over sever thousands of data. Eventually we obtained 92.89% test accuracy using "adam" optimizer.

Table 6.4: Test Accuracy of Semi Supervised GAN. (Explosion Classification)

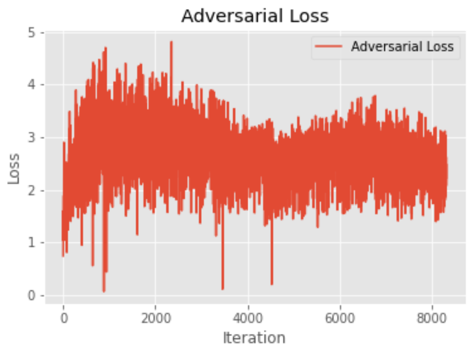
Labeled Data				**Unlabeled Data**	**Optimizer**	**Test Acc.**
Electrical	House	Vehicle	Non-Explosion			
125	125	125	125	1239	Adam	92.76
125	125	125	125	1239	RMS Prop	54.75
125	125	125	125	1239	SGD	90.48

6.3 River Path Segmentation from Satellite Image

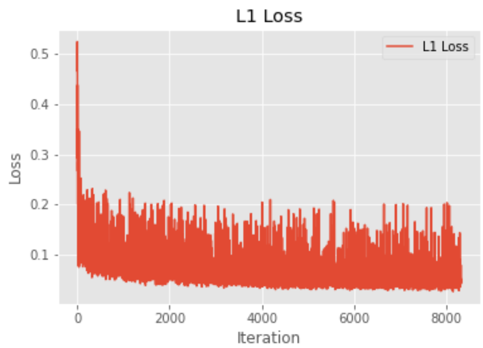
For our second case study (River path segmentation using GANs) we trained the GANs model that was introduced in CVPR 2017 with our dataset and was able to decrease the L1 and Adversarial loss.

Table 6.5: Adversarial and L1 loss of GANs for Satellite Image Segmentation

Iterations	Adversarial Loss	L1 Loss
1	0.740	0.467
2	0.930	0.471
3	1.000	0.525
..	..	..
..	..	..
3820	1.838	0.051
3821	2.382	0.138
..	..	..
..	..	..
8298	2.074	0.034
8299	3.024	0.047
8300	2.738	0.090



(a)



(b)

Figure 6.7: Adversarial Loss and L1 loss of GANs model during Satellite Image Segmentation

Sample output of satellite image segmentation for river path detection using GANs

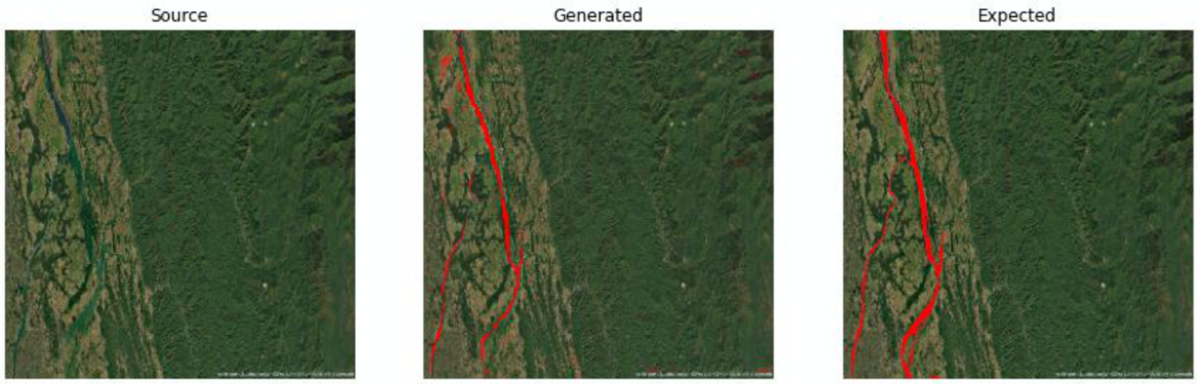


Figure 6.8: River Path Segmentation from Satellite Image (Sample-1)

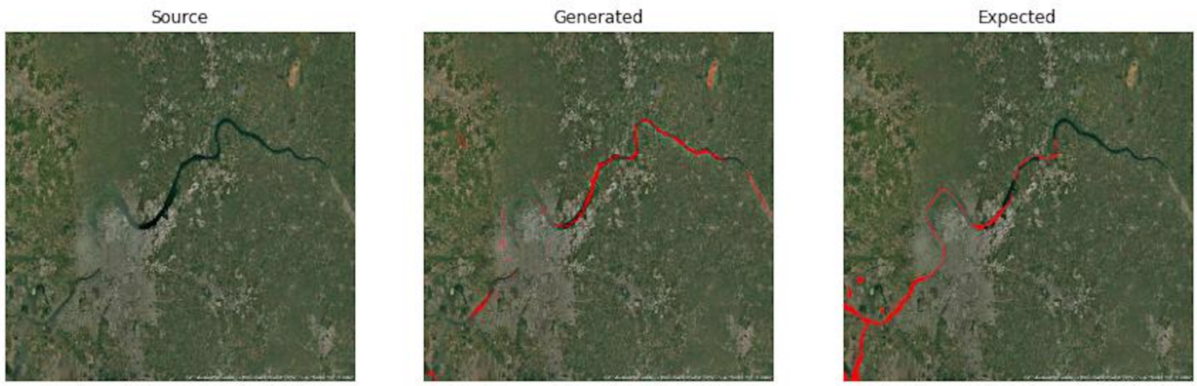


Figure 6.9: River Path Segmentation from Satellite Image (Sample-2)

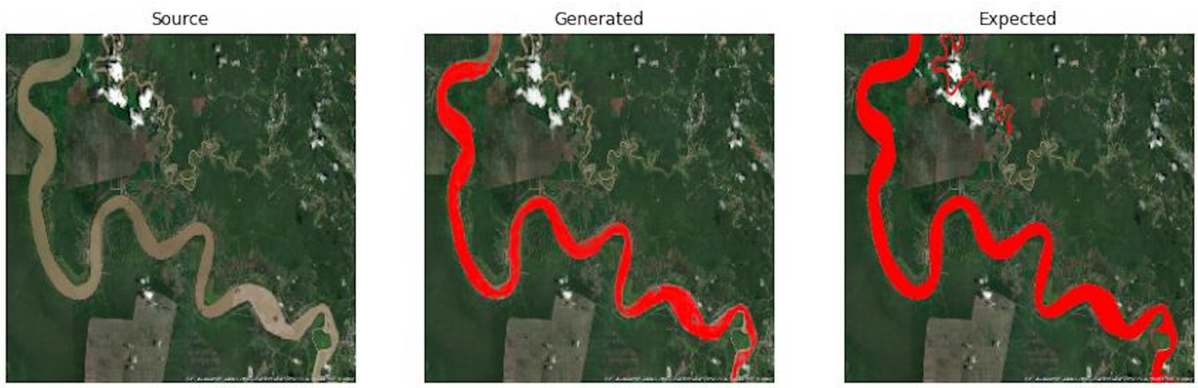


Figure 6.10: River Path Segmentation from Satellite Image (Sample-3)

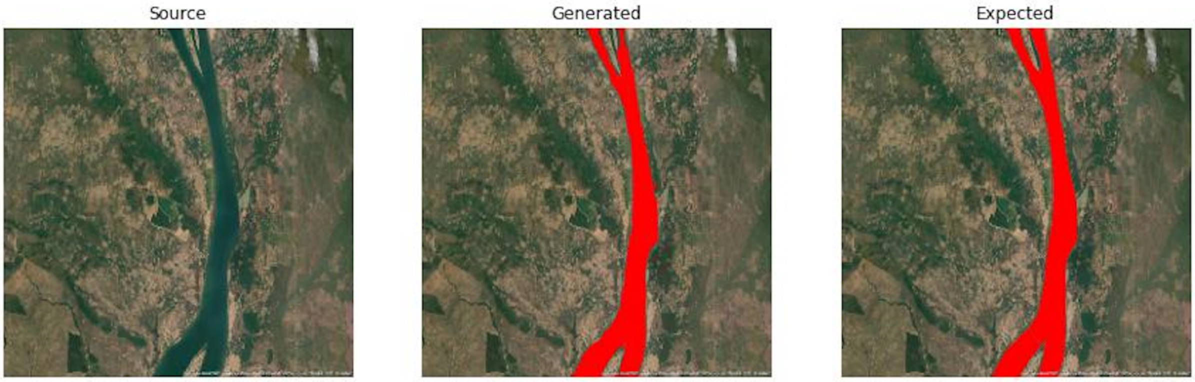


Figure 6.11: River Path Segmentation from Satellite Image (Sample-4)

Figure 6.12: River Path Segmentation from Satellite Image (Sample-5)

Chapter 7

Future Work and Conclusion

In this thesis, we have tried to find ways to solve problems related to explosion detection from CCTV footage and river detection from satellite images using deep learning techniques. Although we have given our best efforts, there are limitations to our approaches. Thus to improve our methods, we have listed several procedures that are stated below.

7.1 Future Work

(i). As we are detecting explosions and fire, a possible scenario would be the model might predict the sunlight as an explosion. To prevent such problems, such data will be added to enrich our data set.

(ii). Throughout our experiment we have seen some models performing better than others. In future, we can ensemble models together so that if a given image is passed to the model, the prediction will be done by several deep learning algorithms and if most of them predict one class over another that class will be the final outcome.

(iii). The detection of the position of the fire YOLOv5 provided 70% accuracy which can be increased by enriching the dataset and using segmentation of the image. In our future work, a more in depth and accurate object detection model (YOLO, YOLOv5) will be tested in order to get higher accuracy.

(iv). In our second case study we segmented river path from satellite image. In future, our challenge will also be segmenting forests, houses, etc. objects from the satellite image to get more in-depth information.

(v). We will also consider machine learning based approach for satellite image segmentation, along with deep learning model such as U-Net. This will help us to compare our approach with other models which will help us to experiment with new approaches and insights.

7.2 Conclusion

Digital image processing has played a vital role in solving most real-life computer vision problems and offered a whole new area of research for computer scientists. In our thesis, we have used deep learning models to perform image processing and analysing in real-time by exploring two important domains of computer vision — classification and image segmentation. We detected and classified explosion from CCTV footage that can help people by creating automated alarm systems and may reduce the danger during any explosion. We have also segmented river path from satellite images without any hassle of external software which can help to understand the behaviour of any river and can be helpful for flood prediction and many other related applications.

References

[1] A. Z. Khuzani, R. Agrawal, and N. Mashhadi, "Fire detection in video sequences using a machine learning system and a clustered quantitative image marker," in *2019 IEEE Global Humanitarian Technology Conference (GHTC)*, pp. 1–5, IEEE, 2019.

[2] "Student Notes: Convolutional Neural Networks (CNN) Introduction." https://indoml.com/2018/03/07/student-notes-convolutional-neural-networks-cnn-introduction/. Accessed: 25 Jun, 2021.

[3] "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way." https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. Accessed: 25 Jun, 2021.

[4] "Visualizations of RNN units." https://kvitajakub.github.io/2016/04/14/rnn-diagrams/. Accessed: 25 Jun, 2021.

[5] "Bidirectional Recurrent Neural Network (BiRNN)." https://www.programmersought.com/article/67195619712/. Accessed: 25 Jun, 2021.

[6] "From scratch—An LSTM model to predict commodity prices." https://medium.com/@vinayarun/from-scratch-an-lstm-model-to-predict-commodity-prices-179e12445c5a. Accessed: 25 Jun, 2021.

[7] "A deep dive into the world of gated Recurrent Neural Networks: LSTM and GRU." https://medium.com/analytics-vidhya/lstm-and-gru-a-step-further-into-the-world-of-gated-rnns-99d07dac6b91. Accessed: 25 Jun, 2021.

[8] "VGG16 – Convolutional Network for Classification and Detection." https://neurohive.io/en/popular-networks/vgg16/. Accessed: 25 Jun, 2021.

[9] "Deep Convolutional Generative Adversarial Networks(DCGANs)." https://medium.datadriveninvestor.com/deep-convolutional-generative-adversarial-networks-dcgans-3176238b5a3d. Accessed: 25 Jun, 2021.

[10] "Semi-supervised learning with Generative Adversarial Networks (GANs)." https://towardsdatascience.com/semi-supervised-learning-with-gans-9f3cb128c5e. Accessed: 25 Jun, 2021.

[11] A. Malhi and R. X. Gao, "Pca-based feature selection scheme for machine defect classification," *IEEE transactions on instrumentation and measurement*, vol. 53, no. 6, pp. 1517–1525, 2004.

[12] H. Yu and J. Yang, "A direct lda algorithm for high-dimensional data—with application to face recognition," *Pattern recognition*, vol. 34, no. 10, pp. 2067–2070, 2001.

[13] N. Kwak, C.-H. Choi, and J. Y. Choi, "Feature extraction using ica," in *International Conference on Artificial Neural Networks*, pp. 568–573, Springer, 2001.

[14] F. Hou, "This a.i. entrepreneur is working to bring machine learning to more industries," Nov 2021.

[15] C. Tao, J. Zhang, and P. Wang, "Smoke detection based on deep convolutional neural networks," in *2016 International conference on industrial informatics-computing technology, intelligent technology, industrial information integration (ICIICII)*, pp. 150–153, IEEE, 2016.

[16] J. B. Maheen and R. Aneesh, "Machine learning algorithm for fire detection using color correlogram," in *2019 2nd international conference on intelligent computing, instrumentation and control technologies (ICICICT)*, vol. 1, pp. 1411–1418, IEEE, 2019.

[17] G. Chandan, A. Jain, H. Jain, *et al.*, "Real time object detection and tracking using deep learning and opencv," in *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pp. 1305–1308, IEEE, 2018.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[20] C. Tang, Y. Ling, X. Yang, W. Jin, and C. Zheng, "Multi-view object detection based on deep learning," *Applied Sciences*, vol. 8, no. 9, p. 1423, 2018.

[21] M. J. Shafiee, B. Chywl, F. Li, and A. Wong, "Fast yolo: A fast you only look once system for real-time embedded object detection in video," *arXiv preprint arXiv:1709.05943*, 2017.

[22] A. Odena, "Semi-supervised learning with generative adversarial networks," *arXiv preprint arXiv:1606.01583*, 2016.

[23] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[24] T. Chavdarova and F. Fleuret, "Sgan: An alternative training of generative adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9407–9415, 2018.

[25] Y. Zhan, D. Hu, Y. Wang, and X. Yu, "Semisupervised hyperspectral image classification based on generative adversarial networks," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 2, pp. 212–216, 2017.

[26] Y. Li and G. Baciu, "Hsgan: Hierarchical graph learning for point cloud generation," *IEEE Transactions on Image Processing*, vol. 30, pp. 4540–4554, 2021.

[27] G. Rajesh, A. R. Benny, A. Harikrishnan, J. JacobAbraham, and N. P. John, "A deep learning based accident detection system," in *2020 International Conference on Communication and Signal Processing (ICCSP)*, pp. 1322–1325, IEEE, 2020.

[28] S. Ghosh, S. J. Sunny, and R. Roney, "Accident detection using convolutional neural networks," in *2019 International Conference on Data Science and Communication (IconDSC)*, pp. 1–6, IEEE, 2019.

[29] D. Tian, C. Zhang, X. Duan, and X. Wang, "An automatic car accident detection method based on cooperative vehicle infrastructure systems," *IEEE Access*, vol. 7, pp. 127453–127463, 2019.

[30] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[31] S. Pare, A. K. Bhandari, A. Kumar, G. K. Singh, and S. Khare, "Satellite image segmentation based on different objective functions using genetic algorithm: a comparative study," in *2015 IEEE international conference on digital signal processing (DSP)*, pp. 730–734, IEEE, 2015.

[32] J. N. Kapur and H. K. Kesavan, "Entropy optimization principles and their applications," in *Entropy and energy dissipation in water resources*, pp. 3–20, Springer, 1992.

[33] X. Xu, S. Xu, L. Jin, and E. Song, "Characteristic analysis of otsu threshold and its applications," *Pattern recognition letters*, vol. 32, no. 7, pp. 956–961, 2011.

[34] M. P. De Albuquerque, I. A. Esquef, and A. G. Mello, "Image thresholding using tsallis entropy," *Pattern Recognition Letters*, vol. 25, no. 9, pp. 1059–1065, 2004.

[35] V. Khryashchev, L. Ivanovsky, V. Pavlov, A. Ostrovskaya, and A. Rubtsov, "Comparison of different convolutional neural network architectures for satellite image segmentation," in *2018 23rd conference of open innovations association (FRUCT)*, pp. 172–179, IEEE, 2018.

[36] V. Khryashchev and R. Larionov, "Wildfire segmentation on satellite images using deep learning," in *2020 Moscow Workshop on Electronic and Networking Technologies (MWENT)*, pp. 1–5, IEEE, 2020.

[37] M. Wu, C. Zhang, J. Liu, L. Zhou, and X. Li, "Towards accurate high resolution satellite image semantic segmentation," *IEEE Access*, vol. 7, pp. 55609–55619, 2019.

[38] "Theft Detection." https://www.kaggle.com/thatbrownkid/thefttest%7D. Accessed: 27 Jun, 2021.

[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[40] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4471–4480, 2019.

[41] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.