A PROJECT REPORT

ON

Object Detection using Mask-R-CNN

For the partial fulfillment of the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

(ARTIFICIAL INTELLIGENCE)

Submitted By:-

Saloni (2001920130130)

Shlok Singh Tomar (2001921520054)

Under the Supervision of

Ms. Anju Chandna

G.L. BAJAJ INSTITUTE OF TECHNOLOGY &

MANAGEMENT, GREATER NOIDA

Affiliated to

DR. APJ ABDUL KALAM TECHNICAL UNIVERSITY, LUCKNOW

2021-22

Declaration

We hereby declare that the project work presented in this report entitled **"Object Detection using Mask-R-CNN",** in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science & Engineering, submitted to A.P.J. Abdul Kalam Technical University, Lucknow, is based on my own work carried out at Department of Computer Science & Engineering, G.L. Bajaj Institute of Technology & Management, Greater Noida. The work contained in the report is original and the project work reported in this report has not been submitted by me/us for an award of any other degree or diploma.

Signature:

Name: Shlok Singh Tomar

Roll.no: 2001921520054

Signature:

Name: Saloni

Roll.no: 2001920130130

Date:

Place: Greater Noida

Certificate

This is to certify that the Project report entitled "**Object Detection using Mask-R-CNN done by Shlok Singh Tomar (2001921520054) and Saloni (2001920130130)** is an original work carried out by them in the Department of Computer Science & Engineering, G.L Bajaj Institute of Technology & Management, Greater Noida under my guidance. The matter embodied in this project work has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

Date:

Ms. Anju Chandna
Signature of the Supervisor

Dr. Sansar Singh Chauhan
Head of the Department

Acknowledgement

The merciful guidance bestowed to us by the almighty made us stick out this project to a successful end. We humbly pray with a sincere heart for his guidance to continue forever.

We pay thanks to our project guide Ms**. Anju Chandna** who has given guidance during this project. Her versatile knowledge has caused us in critical times during the span of this project.

We pay special thanks to our Head of Department Dr. Sansar Singh Chauhan who has been always present as a support and help us in all possible ways during this project.

We also take this opportunity to express our gratitude to all those people who have been directly and indirectly with us during the completion of the project.

We want to thank our friends who have always encouraged us during this project.

Last but not least thanks to all the faculty of the CSE department who provided valuable suggestions during the period of the project.

Abstract

In recent years, image segmentation has captured the interest of those who analyze visual data. In the realm of computer vision, a reliable image segmentation technique is essential because of the wide range of class divergence in images. In the past, the majority of segmentation algorithms focused on manually extracting feature maps. However, that results in a performance and image complexity constraint. Convolutional Neural Networks have demonstrated a significant performance improvement recently to overcome the issues associated with the traditional segmentation method.

This report aims to perform image segmentation using the Mask R-CNN architecture which is an extension of the Faster R-CNN Model which is preferred for object detection tasks. The Mask R-CNN draws a polygon around the object in addition to providing a class label and detecting object, and is good at pixel-level segmentation.

List of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image segmentation is a branch of digital image processing that focuses on partitioning an image into different parts according to its features and properties. The primary goal of image segmentation is to simplify the image for easier analysis. In image segmentation, we divide an image into various parts that have similar attributes. The parts in which we divide the image are called Image Objects.

We are going to perform image segmentation using the Mask R-CNN architecture. It is an extension of the Faster R-CNN Model which is preferred for object detection tasks.

The Mask R-CNN returns the binary object mask in addition to the class label and object bounding box. Mask R-CNN is good at pixel-level segmentation.

1.1 Motivation

Segmentation is widely spread in today's world for example self-driving cars. Image segmentation can be used in self-driving cars for giving easy distinctions between various objects. Be it traffic signals, signboards,

humans, and cars. It can help the driving instruction algorithm to better assess the surroundings before generating the next instruction. We come across image segmentation in our everyday lives and it has many benefits, it is money, time, and also life-saving, for example, in medical imaging.

Image segmentation is used to extract clinically relevant information from medical reports. For example, image segmentation is used to segment tumours which resulted in saving many lives.
Another important implementation of Image segmentation is Circuit Board Defect Detection, in this, if there are some defects in a circuit in a company then the company has to bear the responsibility for the defective devices. If a camera backed with an Image Segmentation model keeps scanning for defects produced in the final product, a lot of money and time can be saved in fixing a defective device. Image segmentation is complicated and a bit confusing for some but this project gives an easy and clear understanding of its working using Mask R-CNN architecture.

1.2 Related Works

There has been a lot of development in the field of Computer Vision in recent years. Ever since Convolutional Neural Networks were introduced, the state of the art in domains such as classification, object detection, image segmentation, etc. has constantly been challenged.

These neural network models are being employed in real-time in emerging fields such as Autonomous Navigation.

Image segmentation is a sub-field of Computer Vision. Image segmentation is the process of classifying each pixel in the image as belonging to a specific category. Though there are several types of image segmentation methods, the two types of segmentation that are predominant when it comes to the domain of Deep Learning are:

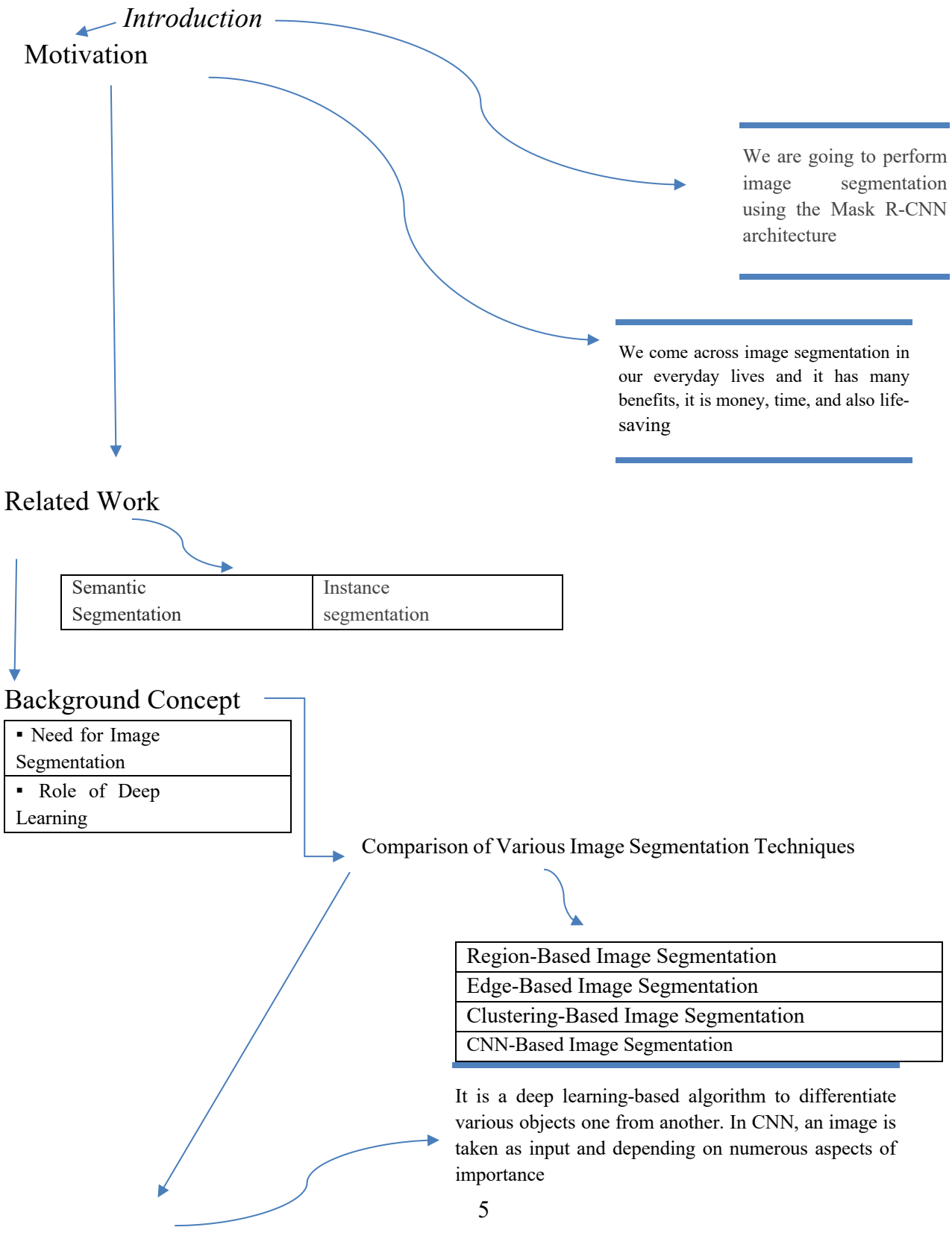
■ Semantic Segmentation:-

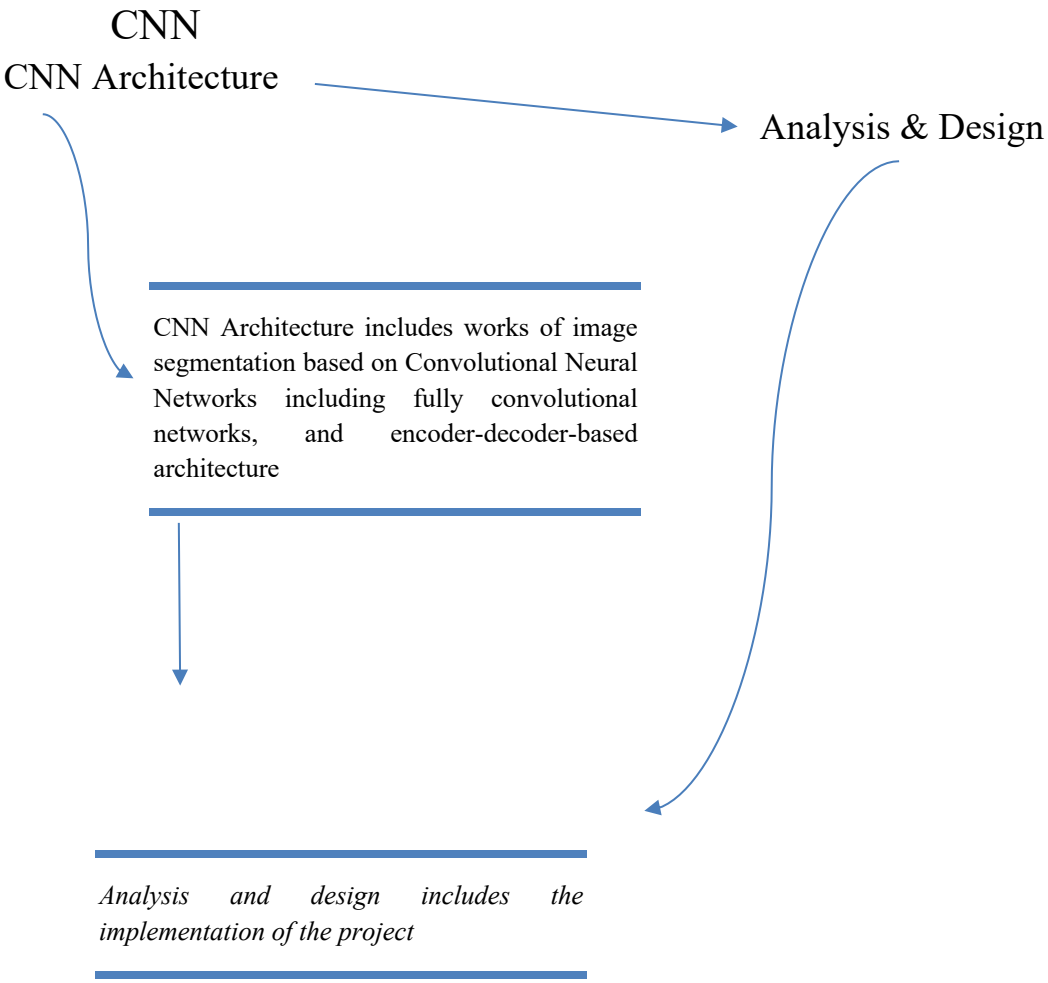
Semantic Segmentation is the process of segmenting the image pixels into their respective classes.

■ Instance segmentation:-

Instance Segmentation is more thorough and usually comes into the picture when dealing with multiple objects.

1.3 Report Structure




CNN
CNN Architecture
Analysis & Design
CNN Architecture includes works of image segmentation based on Convolutional Neural Networks including fully convolutional networks, and encoder-decoder-based architecture
Analysis and design includes the implementation of the project

Chapter 2

BACKGROUND CONCEPT

2.1 Introduction to Image Segmentation

Image processing and computer vision tasks have been revolutionized with the Convolutional Neural Network (CNN) advancement in the past few years. With this improvement, a state-of-the-art performance has been achieved in the field of image segmentation over the traditional approach. In general, image segmentation is the task of finding a group of pixels in an image, based on either similarity (bottom-up) or belonging to the same region or object (top-down). Hence, it means partitioning an image into multiple discrete regions or segments based on some criteria where the segments are meaningful and disjoint such as gray tone or texture. Due to the purely uniform characteristics and the small holes surrounded by the homogeneous regions of the image, segmentation becomes a challenging task. From the pixel classification point of view, image segmentation can be classified into a distinct category- semantic segmentation and instance segmentation. In semantic segmentation, all image pixels are categorized with a set of objects by performing pixel-level labeling (e.g., map pedestrian, tree, car, from an image). On the other hand, in instance segmentation, each region of interest in the image is detected and delineated. Hence it extends the scope of semantic segmentation.

Fig:- 2.1 (a) Semantic Segmentation

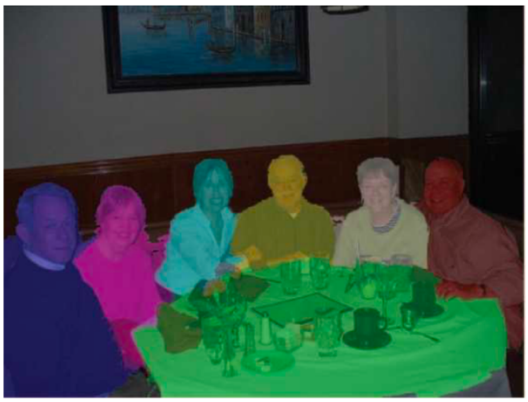


fig:- 2.2 (b) Instance Segmentation

Semantic segmentation provides more detailed localized information about the scene by answering the question of how many people are there or what are the people doing.

2.2 Need for Image Segmentation

Suppose, you are crossing a road and you see various objects around like vehicles, traffic lights, footpaths, zebra-crossing, and pedestrians. Now, while crossing the road your eyes instantly analyze each object and process their locations to take a decision of whether to cross the road at that moment or not. Can computers do this task? The answer to this question was ‗No' since a long time before the breakthrough inventions in computer vision and object identification. But now with the help of image segmentation and object identification techniques, it is possible for computers to see the real-world objects, and based on their positions they can now take necessary actions. Also, in the medical field, image segmentation plays a very important role. For example, we can consider the task of identifying cancer cells in the microscopic image of human blood. If you ask of identifying cancer cells, it's crucial to identify the shape of cells in the blood and any unusual growth in the shape of blood cells will be diagnosed as the presence of cancer cells. This makes the recognition of cancer in blood at an early stage so that it can be cured within time.

2.3 Role of Deep Learning in Image Segmentation

As all other machine learning techniques work in a way of taking the labeled training data and based on the training, they process new input and take the decisions. But as opposed to this deep learning works with neural networks and can make conclusions of its own without the need for labeled training data. This method is useful for a self-driving car so that it can differentiate between a signboard and a pedestrian. Neural networks use algorithms that are present in the network side by side and the output of one algorithm is subject to the outcome of another algorithm. This creates a system that can think as a human being for taking decisions. And this makes a model which is a perfect example of an artificial intelligence system.

2.4 Comparison of Various Image Segmentation Techniques:-

▪ 2.4.1 Region-Based Image Segmentation

It is the simplest way of segmenting an image based on its pixel values. This makes use of the fact that there will be a huge difference in pixel values at the edges of an object from the pixel values of its background pixels. So, in such cases, we can set a threshold value, and the difference in pixel values falling below the threshold can be segmented accordingly. In case there is one object in front of a single background only one threshold value will work. But in case there are multiple objects or overlapping among objects we might need multiple threshold values. This technique is also called threshold segmentation.

▪ 2.4.2 Edge-Based Image Segmentation

In an image, what divides an object from its background or other object is an edge. So, this technique makes use of the fact that whenever a new edge is encountered while scanning an image, we are detecting a new object. This detection of an edge is done with the help of filters and convolutions. Here, we have a filter matrix that we need to run over the image to detect the specified type of edge for which the filter was made.

▪ 2.4.3 Clustering-Based Image Segmentation

This technique work in the way that it divides the data points in an image into some clusters having similar pixel values. Based on the number of

objects that might be present in the image the value of a number of clusters is selected. Then during the clustering process, similar kinds of data points were classified into a single cluster. In the end, the image gets classified into several regions. Although this is a time taking process but it provides accurate results on small datasets.

▪ 2.4.4 CNN-Based Image Segmentation

This method is currently the state-of-the-art technology in the image segmentation field of research. It works on images that are of 3 dimensions i.e. height, width, and a number of channels. The first two dimensions tell us the image resolution and the third dimension represents the number of channels (RGB) or intensity values for red, green, and blue colors. Usually, images that are fed into the neural network are reduced in dimensions which reduces the processing time and avoids the problem of underfitting. Even though we take an image of size 224*224*3 which when converted into 1 dimension will make an input vector of 150528. So, this input vector is still too large to be fed as input to the neural network.

Comparison of Image Segmentation Techniques

Advantages

Region Based	Edge Based	Cluster Based	CNN Based
Simple calculations.	Good for images having better contrast between objects.	Works really well on small datasets and generates excellent clusters.	Provides most accurate results.
Fast operation speed.			

Disadvantages

Region Based	Edge Based	Cluster Based	CNN Based
If there is no significant grayscale difference or an overlap of the gray scale pixel values, it becomes difficult to get accurate segments.	Not suitable when there are too many edges in the image.	Computation time is too large and expensive.	Takes comparably longer time to train the model.

Table:2.4.4

Chapter 3

Convolutional Neural Network (CNN)

Convolutional Neural Network is a deep learning-based algorithm to differentiate various objects one from another. In CNN, an image is taken as input and depending on numerous aspects of importance i.e., weights and biases have been assigned.

The building blocks of CNN are similar to the architectural pattern of the human brain. It consists of a sequence of layers:-
▪ Convolutional Layer, ▪ Pooling Layer, and ▪ Fully Connected Layer.

Each layer transforms its knowledge into another layer through activation functions.

3.1 Convolutional Layer

Convolution operation means the dot product between a kernel and the image. It is done by applying a kernel on small patches on an image and sliding the kernel over the whole image.
In figure 3.1, there is an RGB image separated by its three channels red, green, and blue. This image can be any size. The goal is to transform the image into such a form without losing features which is easier for further processing. Convolutional layer achieve this by
using a filter d defined with very small sizes e.g., 3 × 3, 5 × 5, etc. After that, a feature map is created by convolving the filter over the input image or the previous layer of the input image.
The resulting d feature map is a new 3D structure of the input layer called a tensor.

For example, the tensor will be 32 × 32 × 32 × d for an input of size 32 × 32 × 32. This layer contains very few parameters- if the map contains 3 × 3 × 3 values, the total parameters will be 3 × 3 × 3 + 1(weight), i.e. 28 parameters.

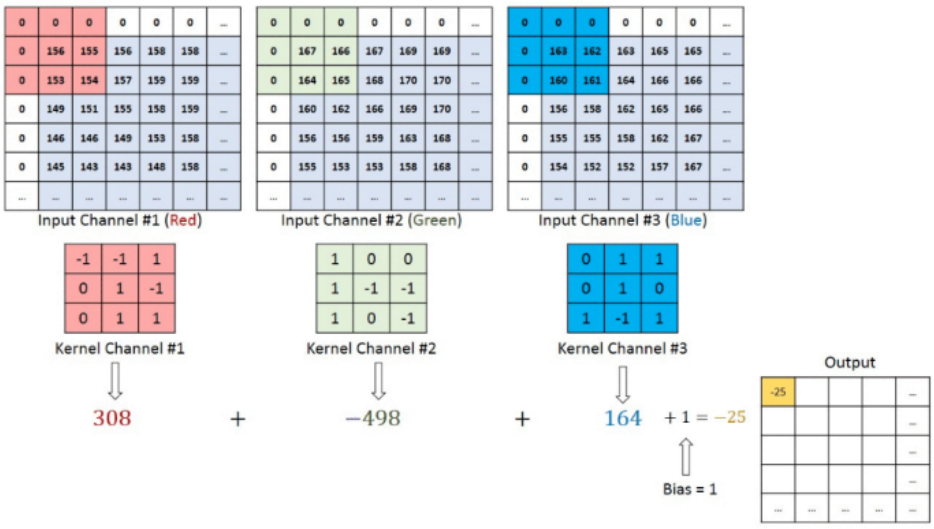


Fig 3.1: Convolutional Layer with a 3 × 3 × 3 Kernel

3.2 Pooling Layer

After reducing the size of the convolution layer, the elements are still too high. For further dimensionality reduction, a pooling layer needs to be inserted in-between successive convolution layers. The goal of this layer is to reduce the size of the spatial domain to reduce the computing parameters in the network for controlling overfitting. It is also useful to extract the dominant features. Two types of pooling can be performed in the pooling layer- Max Pooling and Mean Pooling. The maximum value of the image covered by the filter is returned by the max-pooling layer. Similarly, the average value of the image covered by the filter is returned by mean pooling.

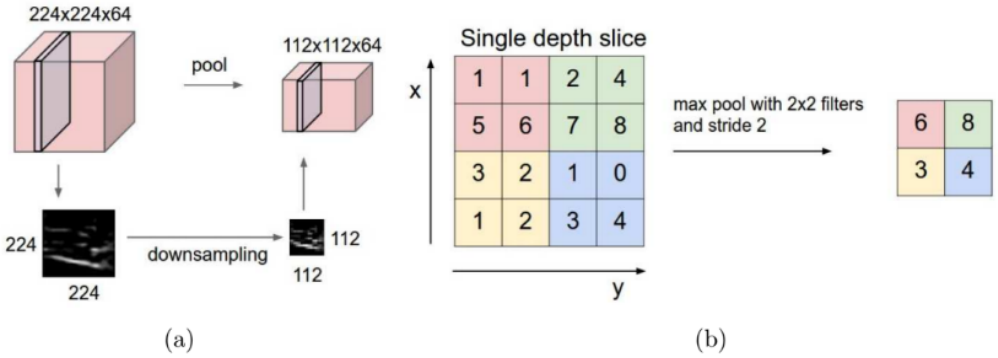


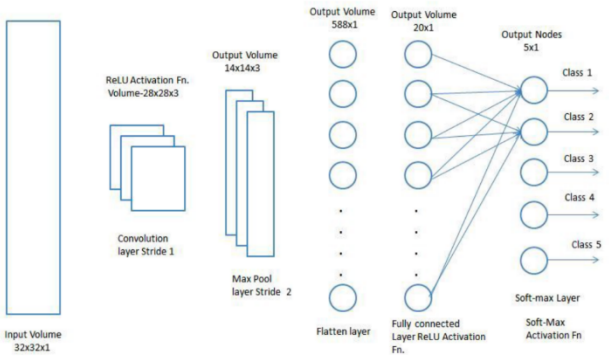
Fig3.2: Pooling Layer

This is pooled with a filter with size 2 and produces the output image with size 112 × 112 × 64. Here, the depth is preserved. In the figure, the max-pooling down-sampling operation with a stride of 2 has been shown. Here, each operation is performed over 4 numbers (2 × 2 square).

3.3 Fully Connected Layer

A fully connected layer is a combination of all activations. All the non-linear high-level features obtained from the convolutional layer are combined in this layer. Hence it can be seen as a Multi-level perception/regular Neural Network. After that, matrix multiplication followed by a bias offset has been performed for learning a non-linear function in that space.
The illustration of a fully connected layer has been shown in the figure.

Fig: 3.3: Fully Connected Layer



Chapter 4

CNN Architecture

This chapter includes works of image segmentation based on Convolutional Neural Networks including fully convolutional networks, and encoder-decoder-based architecture. Apart from the architecture, the similarity and challenges of different approaches are also explained.

4.1 Fully Convolution Networks

Fully Convolutional Networks (FCN) is the first deep learning-based image seg- mentation algorithm. An FCN learns to predict the output tensor based on only convolutional layers. The goal of the algorithm is to enable taking an image of arbitrary size as input and produce a segmentation of the same size. To manage the arbitrary-sized input-output, the fully connected layer is replaced with fully convolutional layers. Hence, instead of classification scores, a spatial segmentation map has been produced by this algorithm.

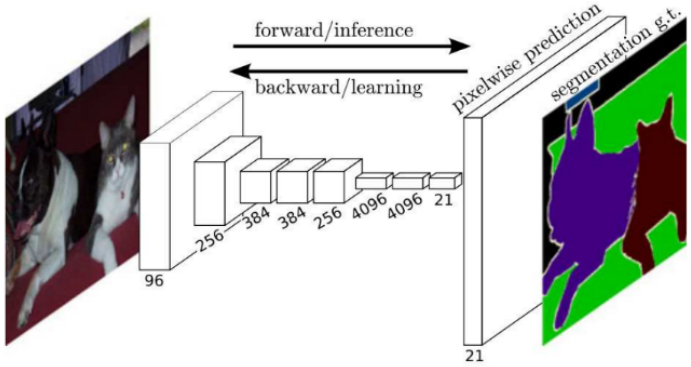


Figure 4.1: Fully Convolutional Networks (FCN)

4.2 SegNet

SegNet is an encoder-decoder algorithm used for pixel-wise semantic segmentation. It is composed of an encoder network with 13 convolutional layers which corresponds to the 13 convolutional layers in the VGG16 network and a decoder network followed by a pixel-wise classification layer. To handle the increasing number of features, this algorithm contains a max-pooling layer to decrease the spatial resolution. These lower-resolution input feature maps then compute the pooling indices to perform non-linear upsampling. Hence the need for learning to be upsampled is eliminated.

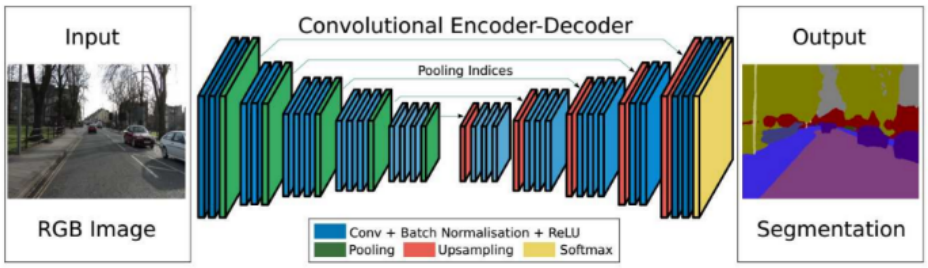


Figure 4.2: An illustration of the SegNet architecture

A high-level illustration of the SegNet architecture is presented in figure 4.2. In the figure, there is not a fully connected layer hence there only exists only convolutional
layers.

4.3 Mask R-CNN

Segmenting the background is useless for many applications. Mask R-CNN uses a faster regional convolutional network (R-CNN) approach to extract the bounding boxes around the interesting objects, followed by the prediction of a mask to segment the object.

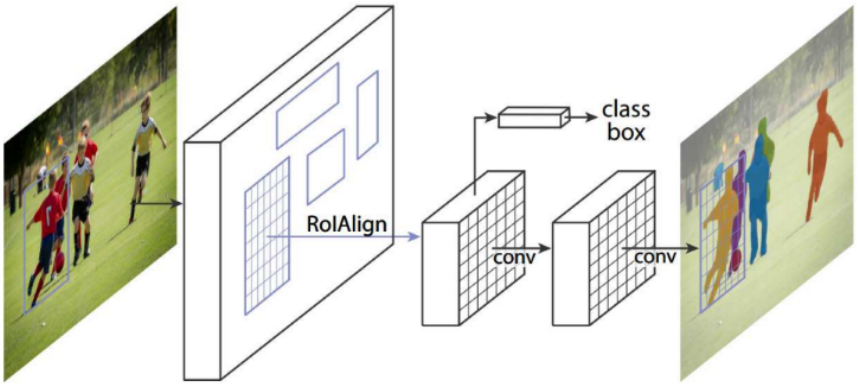


Figure 4.3: Semantic segmentation using Mask R-CNN

The general Faster R-CNN architecture is composed of a two-step process- a Region Proposal Network (RPN) which is used to propose the bounding box candidates and a RoIPool layer to extract the features for each bounding box candidate. These two steps process is adopted by the Mask R-CNN along with an additional step- a binary mask is also predicted for RoI. Therefore, compared to Faster R-CNN with two output branches, the output of the Mask R-CNN is composed wof3 branches (figure 4.3)- the bounding box candidates are computed in the first branch, and after that, the associated classes are computed in the second branch and finally the binary masks are computed in the third branch to segment the image.

The losses of the bounding box coordinates, the predicted class, and the segmentation masks are combined in the loss function of the Mask R-CNN and trained of them jointly. Both the classification and bounding box loss remain identical.
Only the segmentation masks are encoded into binary masks of resolution of m × m for each RoI. A per-pixel sigmoid function is applied to compute the average binary cross-entropy loss that encodes the binary masks.

During the RoIPooling, the region of feature maps can be slightly misaligned from the region of the original image, since this operation quantizes the floating- point number of RoI into a discrete granularity. Specificity at the pixel level of the image is required to segment the image accurately. Therefore, this misalignment can be guided to a wrong segmentation. To solve this alignment problem a RoIAlign is used. In figure 4.4, the feature maps and the RoI are represented by the dashed line and solid line respectively and the 4 dots in each bin represent the sampling points.
Bilinear interpolation is applied from nearby grid points on the feature map to compute each sampling point value. Therefore, no quantization is required.

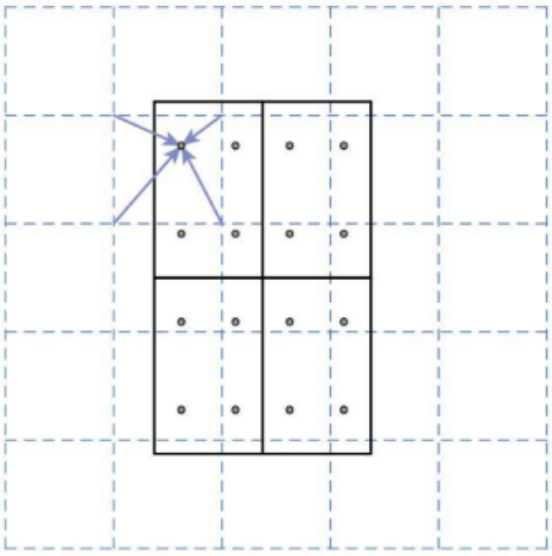


Figure 4.4: RoIAlign working procedure

Chapter 5

Analysis & Design

5.1 Hardware

Provisioned from Google Colaboratory

RAM: 12 GB
Disk: 78 GB
GPU: 15 GB

```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 460.32.03    Driver Version: 460.32.03    CUDA Version: 11.2     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   44C    P8     9W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
```

Fig: 5.1 Hardware

5.2 Sample Code

5.2.1 Dataset Annotation

Dataset(images for the class to be identified) is collected and the collected images are annotated manually for image segmentation on an open-source platform MakeSense.
The dataset is then exported as a COCO JSON file.

5.2.2 Installation

TensorFlow is installed in a Google Colab Notebook. Further, the Mask RCNN model is cloned from the GitHub directory and installed in the Google Colab directory.

```
!wget https://pysource.com/extra_files/maskrcnn_colab_demo_commit_17.zip
!unzip maskrcnn_colab_demo_commit_17.zip
import sys
sys.path.append("/content/maskrcnn_colab/mrcnn_demo")
from m_rcnn import *
%matplotlib inline
```

Fig: 5.2.1 Installation

5.2.3 Image Dataset

The annotated dataset is then extracted.

```
# Extract Images
images_path = "/content/dataset.zip"
annotations_path = "annotations.json"

extract_images(os.path.join("/content/",images_path), "/content/dataset")
```

Fig: 5.2.2 Image Dataset

```
Extracted: 43 images
```

Next, the class for which the model is going to be trained is specified.

```
dataset_train = load_image_dataset(os.path.join("/content/", annotations_path), "/content/dataset", "train")
dataset_val = load_image_dataset(os.path.join("/content/", annotations_path), "/content/dataset", "val")
class_number = dataset_train.count_classes()
print('Train: %d' % len(dataset_train.image_ids))
print('Validation: %d' % len(dataset_val.image_ids))
print("Classes: {}".format(class_number))
```

5.2.4 Training

The model is trained for the custom dataset.

Configuration loading:

Fig: 5.2.3 Training:-

```
# Load Configuration
config = CustomConfig(class_number)
# config.display()
model = load_training_model(config)
```

Training is started:

```
train_head(model, dataset_train, dataset_train, config)
```

The model is trained for 39m 47s up to EPOCH 2/5

```
Starting at epoch 0. LR=0.001

Checkpoint Path: /content/maskrcnn_colab/logs/object20221124T1639/mask_rcnn_object_{epoch:04d}.h5
Selecting layers to train
fpn_c5p5               (Conv2D)
fpn_c4p4               (Conv2D)
fpn_c3p3               (Conv2D)
fpn_c2p2               (Conv2D)
fpn_p5                 (Conv2D)
fpn_p2                 (Conv2D)
fpn_p3                 (Conv2D)
fpn_p4                 (Conv2D)
rpn_model              (Functional)
mrcnn_mask_conv1       (TimeDistributed)
mrcnn_mask_bn1         (TimeDistributed)
mrcnn_class_conv1      (TimeDistributed)
mrcnn_class_bn1        (TimeDistributed)
mrcnn_mask_conv2       (TimeDistributed)
mrcnn_mask_bn2         (TimeDistributed)
mrcnn_class_conv2      (TimeDistributed)
mrcnn_class_bn2        (TimeDistributed)
mrcnn_mask_conv3       (TimeDistributed)
mrcnn_mask_bn3         (TimeDistributed)
mrcnn_bbox_fc          (TimeDistributed)
mrcnn_mask_conv4       (TimeDistributed)
mrcnn_mask_bn4         (TimeDistributed)
mrcnn_mask_deconv      (TimeDistributed)
mrcnn_class_logits     (TimeDistributed)
mrcnn_mask             (TimeDistributed)
Epoch 1/5
500/500 [==============================] - 1300s 3s/step - batch: 249.5000 - size: 4.0000 - loss: 0.4297 - val_loss: 0.2529
Epoch 2/5
500/500 [==============================] - 1197s 2s/step - batch: 249.5000 - size: 4.0000 - loss: 0.1685 - val_loss: 0.0918
```

5.2.5 Detection/Testing

The model is loaded:

```
test_model, inference_config = load_test_model(class_number)
```

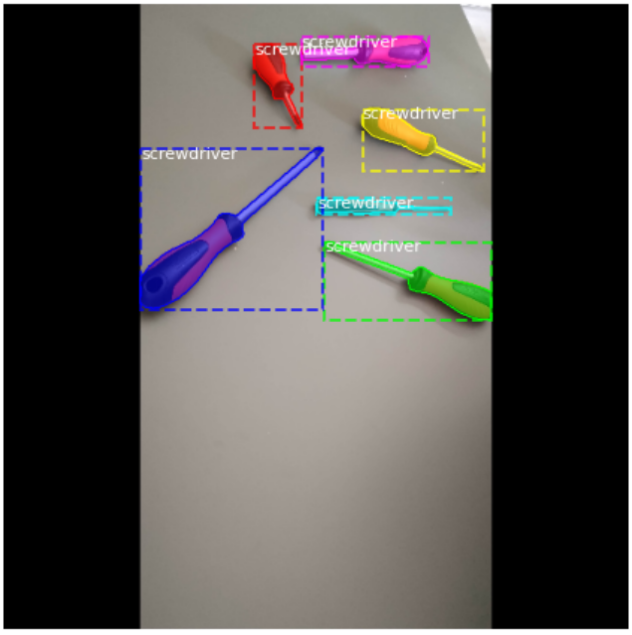
A test is run on a random image:

```
test_random_image(test_model, dataset_val, inference_config)
```

Table: 5.2.4 Detection/Testing

```
original_image           shape: (512, 512, 3)         min:    0.00000  max:  255.00000  uint8
Trained model result
Processing 1 images
image                    shape: (512, 512, 3)         min:    0.00000  max:  255.00000  uint8
molded_images            shape: (1, 512, 512, 3)      min: -123.70000  max:  151.10000  float64
image_metas              shape: (1, 14)               min:    0.00000  max:  512.00000  int64
anchors                  shape: (1, 65472, 4)         min:   -0.70849  max:    1.58325  float32
```

Fig: 5.2.5 (a) Detection



Fig; 5.2.5 (b) Annotations

Chapter 6

Discussion

In this report, numerous image segmentation methods have been discussed by highlighting their key contributions, advantage, and disadvantage. It is really hard to select the right method for a problem among so many different options. This hard task can be done in an optimal way by analyzing the parameters that affect the choice.

The availability of datasets and their annotation is considered one of the most important aspects that affect the performance of convolutional neural network-based image segmentation. When it is needed to work with a small-scale dataset, it is recommended to pre-train the network with a widely used dataset of a similar domain. However, pixel-level segmentation labels may not be available all the time. In those cases, pre-trained parts of the networks such as classification, localization,n, etc. can enhance the learning rate. Selection of the appropriate methods for constructing the segmentation model is the next important aspect. For numerous fully connected methods, the pre-trained classifier can be used. By combining information from various depths of the segmentation network, some multi-scale feature fusion can be carried out. For encoder-decoder-based architecture, the encoder part can be constructed by using a pre-trained classifier like VGGNet, ResNet or DenseNet. Moreover, the downsampling and upsampling operation need to be designed carefully in the encoder-decoder-based networks so that the output can be generated of the same size as the input.

Chapter 7

Conclusion

Convolutional Neural Networks have seen a new wave in the field of image segmentation. Starting with the traditional image segmentation approach, this report thoroughly explained the introduction of convolutional neural networks for segmenting the image. The pros and cons of some of the prominent state-of-the-art algorithms of image segmentation are also discussed. These explanations aim to allow the reader to grasp the basic concepts of CNN-based image segmentation algorithms.

Furthermore, the quantitative performance analysis of these algorithms on some of the most popular benchmarks along with the accuracy metrics has also been explained. Finally, different approaches that can be taken to enhance the performance and accuracy of CNN-based image segmentation have been discussed.

Chapter 8

References

1. CS231n Convolutional Neural Networks for Visual Recognition, https:// cs231n.github.io/convolutional-networks/#pool
2. Arnab, A., Zheng, S., Jayasumana, S., Romera-Paredes, B., Larsson, M., Kir- illov, A., Savchynskyy, B., Rother, C., Kahl, F., Torr, P.H.S.: Conditional Ran- dom Fields Meet Deep Neural Networks for Semantic Segmentation: Combining Probabilistic Graphical Models with Deep Learning for Structured Prediction. IEEE Signal Processing Magazine 35(1) (Jan 2018), conference Name: IEEE Signal Processing Magazine
3. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. arXiv:1511.00561 [cs] (Oct 2016), http://arxiv.org/abs/1511.00561, arXiv: 1511.00561
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected crfs. arXiv preprint arXiv:1412.7062 (2014)
5. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence 40(4), 834–848 (2017)