International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 06 Issue: 12 | December - 2022 Impact Factor: 7.185 ISSN: 2582-3930

EVOLUTION OF IMAGE SEGMENTATION METHODS

A. NAGA VAMSHI REDDY, K. SRIDHAR REDDY, Y. SOUREESH VARMA
Dr. H. BALAJI (Guide)

Sreenidhi Institute of Science and Technology
Hyderabad

ABSTRACT:

It's always fascinating to know how machines does the processing that humans do all the time using senses with high speed and accuracy. Image segmentation is a crucial task in computer vision and image processing, with numerous segmentation algorithms being found in the literature. It has important applications in scene understanding, medical image analysis, robotic perception, video surveillance, augmented reality, and
image compression, among others. This paper discusses the evolution of methods used by machines for image segmentation. This research compares various methods for image segmentation and finds out the better one. This paper mainly concentrates on CNN based image segmentation method. Implementation and results of Faster R-CNN algorithm are provided.

INTRODUCTION

Since the beginning of the field, image segmentation has been a basic issue in computer vision. It involves dividing up images (or video frames) into various segments and objects and is a crucial part of many visual understanding systems. It is used in a wide variety of applications, including medical image analysis (such as tumour boundary extraction and measurement of tissue volumes), autonomous vehicles (such as navigable surface and pedestrian detection), video surveillance, and augmented reality, to name a few. Some of the early stage methods included no neural networks or deep learning and are mostly based on comparing pixels among each other. These methods cannot be customized for a specific purpose. In light of this, the widespread popularity of deep learning (DL) has inspired the creation of fresh

methods for segmenting images using DL models.

Only two early stage methods are discussed here to give an overview of how those are used for image segmentation. However, it is enough for you to understand the advantages and limitations of the same which sets the stage for preferring deep learning based image segmentation methods.

WHAT IS IMAGE SEGMENTATION?

How do autonomous cars detect the objects from the images of its camera? An image taken by the car during the motion usually contains multiple objects. Image is seen as a matrix of pixels and in this case each pixel either belongs to the background or to the active object. So, the determination of each pixel in the image so it either belongs to a background class or specific object class is called image segmentation.

Image segmentation is a technique that divides a digital image into various subgroups known as Image segments, which serves to simplify future processing or analysis of the image by decreasing the complexity of the original image. In simple words, segmentation is the process of giving pixels labels.

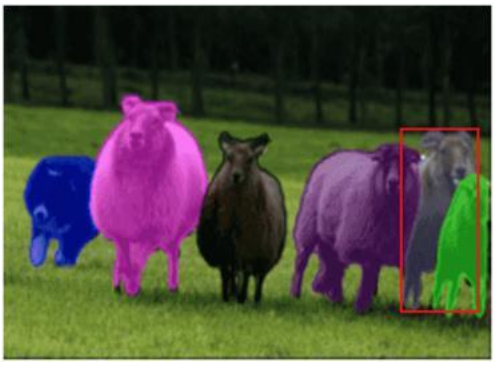


Fig1: Image is segmented for each instance of the animal

THRESHOLD BASED SEGMENTATION METHOD:

A straightforward type of picture segmentation is image thresholding segmentation. It is a technique for converting an original image into a binary or multi-colour image by applying a threshold value to the pixel intensity.

We shall take into account the intensity histogram of each pixel in the image during thresholding. After that, we will choose a threshold to segment the image. For instance, we used a threshold of 60 when taking into account image pixels with a range of 0 to 255. Therefore, all pixels with values of 0 (black) or greater will be delivered, and all pixels with values of larger than 60 will be provided with a value of 255. (white).

We can segment an image into sections based on the brightness of the object and background in an image containing an object and backdrop. But to separate an image into an object and a backdrop, this threshold must be precisely calibrated. **Global thresholding:** We utilise a bimodal picture in this technique. A picture that has two intensity peaks in the intensity distribution plot is said to be bimodal. One for the background and one for the object. The global threshold is then calculated for the entire image and applied to the entire image. The fact that this threshold performs so poorly when the image is lit poorly is a drawback.

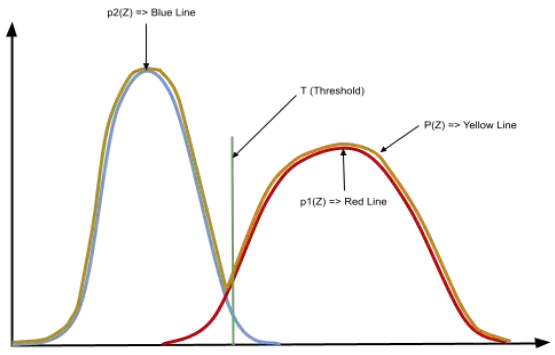


Fig2: Threshold is found plotting pixel intensities

Input:

Fig3: Input to the threshold based model

Output:

Fig4: Output from the model

EDGE BASED SEGMENTATION:

Edge-based segmentation uses a variety of edge detection operators to locate edges in a picture. These borders identify areas in an image where the grey levels, colour, texture, etc. diverge. The degree of grey may vary as we walk from one area to another. Therefore, we can locate the edge provided we can identify that discontinuity. There are several edge detection operators available, however the generated image is an intermediate segmentation result and shouldn't be mistaken for the segmented image as a whole. In order to segment the image, we must execute additional processing on it. In order to limit the number of segments rather than chunks of small borders that can impede the process of region filling, additional procedures include integrating the edges segments that were obtained into one segment. To give the object's boundary a seamless appearance, this is done. In order to obtain the final segmented image, region-based or any other type of segmentation can be applied on the intermediate segmentation result obtained through edge segmentation.

Output:

Fig5: output from the model

DEEP LEARNING BASED IMAGE SEGMENTATION:

Convolutional Neural Networks are used for Image segmentation as deep learning networks.

WHAT IS CNN?

Convolutional Neural Networks, often known as CNNs, is a type of neural network designed for deep learning algorithms that are particularly useful for tasks involving pixel data and image recognition.

CNN is the most effective neural network for detecting and recognizing the

item from the input. We employed it in our project due to its correctness.

Inside Of CNN:

The input must pass through each of CNN's three layers in order to produce its output. The CNN's three tiers are

- Convolutional layer
- Pooling Layer
- Fully Connected Layer

Convolutional Layer: In this layer, most computing takes place. A kernel or filter inside this layer moves over the image's receptive fields during the convolution process to determine if a feature is present.

The kernel traverses the entire picture over several numbers of iterations. A dot product between the input pixels and the filter is calculated at the end of each cycle. A feature map or convolved feature is the result of the dots being connected in a certain pattern. In this layer, the picture is ultimately transformed into numerical values that the CNN can understand and extract pertinent patterns from. In this layer, the picture is ultimately transformed into numerical values that the CNN can understand and extract pertinent patterns from

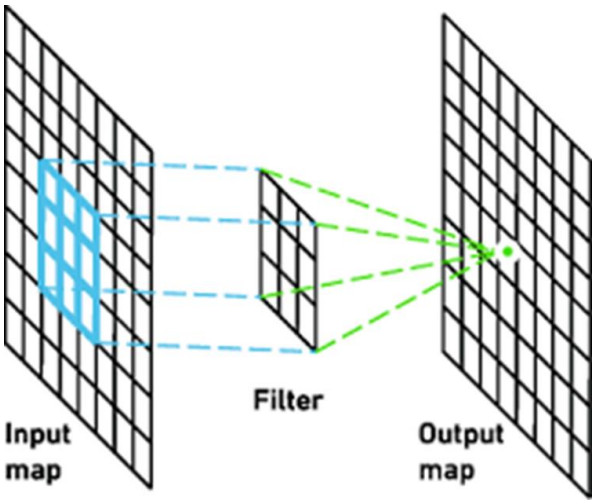


Fig6: Convolutional Layer

Pooling Layer: The pooling layer similar to the convolutional layer sweeps a kernel or filter over the input image. Contrary to the convolutional layer, the pooling layer has fewer input parameters but also causes some information to be lost. Positively, this layer simplifies the CNN and increases its effectiveness.

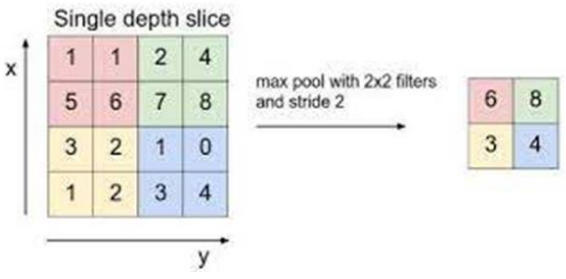


Fig 7: Pooling Layer

Fully Connected Layer: Based on the characteristics gathered in the preceding layers, picture categorization in the CNN takes place in the FC layer. Fully lined in

this context indicates that every activation unit or node of the subsequent layer is connected to every input or node from the preceding layer.

WORKING OF CNN

Multiple layers of a CNN are possible, and each layer trains the CNN to recognize the many aspects of an input picture. Each picture is given a filter or kernel to create an output that grows better and more detailed with each layer. The filters may begin as basic characteristics in the bottom levels.

To examine and identify characteristics that specifically reflect the input item, the complexity of the filters increases with each additional layer. As a result, after each layer, the output of each convolved picture becomes the input for the following layer. The CNN recognizes the picture or objects it represents in the final layer, which is an FC layer.

The input image is processed via a number of different filters during convolution. Each filter performs its function by turning on specific aspects of the image, after which it sends its output to the filter in the subsequent layer. The procedures are repeated for dozens, hundreds, or even thousands of layers as each layer learns to recognize various characteristics. Finally, the CNN is able to recognize the full object thanks to the picture data flowing via its numerous layers.

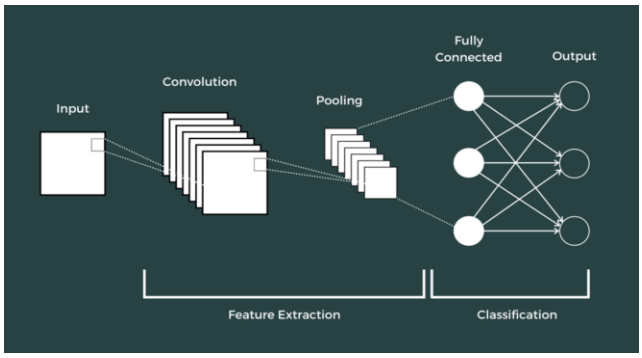


Fig8: CNN layers

ADVANTAGE OF CNN

CNN is particularly renowned for its results' great accuracy despite the volume of data. As CNN advances, it removes all the extraneous data from the input provided in order to improve the accuracy of the outputs. This is because CNN has from its previous phase. When compared to single-layered neural networks, many layers significantly boost the output accuracy. Any device can easily implement CNN construct models; mobile phones can even run them. CNN is employed in many different applications, including automotive and face recognition.

R-CNN ALGORITHMS

Models for object detection using regions with CNNs are based on the following three processes:

Find regions in the image that might contain an object. These regions are called *region proposals*. Extract CNN features from the region proposals. Classify the objects using the extracted features.

There are three variants of an R-CNN. Each variant attempts to optimize, speed up, or enhance the results of one or more of these processes.

R-CNN

The R-CNN detector first generates region proposals using an algorithm such as Edge Boxes. The proposal regions are cropped out of the image and resized. Then, the CNN classifies the cropped and resized regions. Finally, the region proposal bounding boxes are refined by a support vector machine (SVM) that is trained using CNN features.

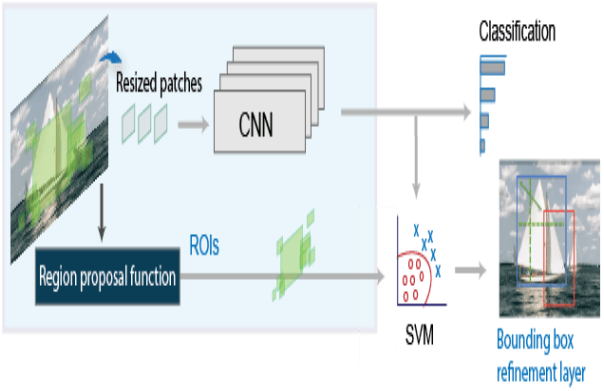


Fig9: RCNN Pipeline

Fast R-CNN

The Fast R-CNN detector, like the R-CNN detector, generates region recommendations using a method akin to Edge Boxes. The Fast R-CNN detector processes the complete image as opposed to the R-CNN detector, which shrinks and resizes region proposals. Fast R-CNN pools CNN features corresponding to each area proposal, whereas an R-CNN detector must categorise each region. Because computations for overlapping regions are shared in the Fast R-CNN detector, it is more effective than R-CNN.

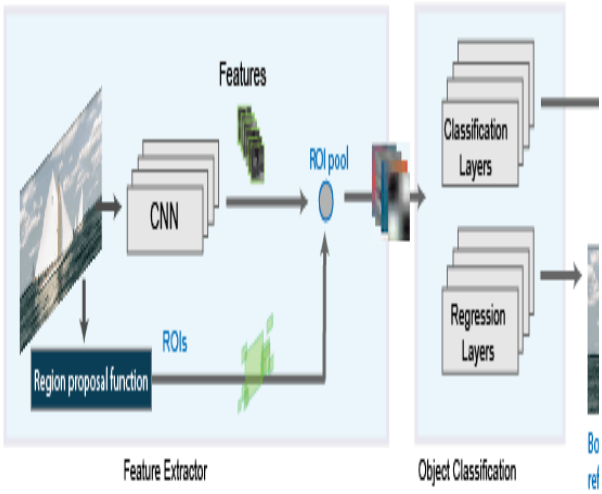


Fig10: Fast R-CNN pipeline

Faster R-CNN

Instead of employing an external method like Edge Boxes, the Faster R-CNN detector incorporates a region proposal network (RPN) to create region suggestions directly in the network. Anchor Boxes are used by the RPN to detect objects. The network generates region ideas more quickly and accurately for your data.

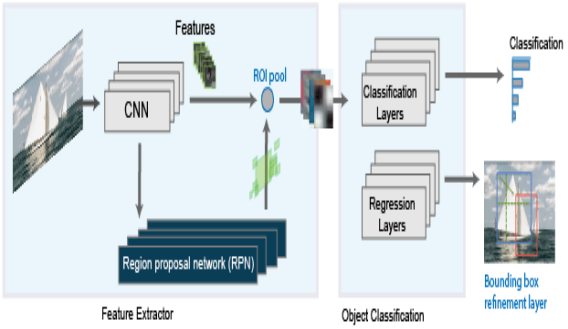


Fig11: Faster R-CNN pipeline

STEPS FOR IMPLEMENTATION

- Preparing Data Set
- Importing Libraries
- Data Visualization
- Training the model
- Testing the model

PREPARING DATASET

Our goal will be to tackle a Blood Cell Detection problem while working on a healthcare-related dataset. Our job is to use microscopic image readings to identify every Red Blood Cell (RBC), White Blood Cell (WBC), and Platelet in each image that is obtained.

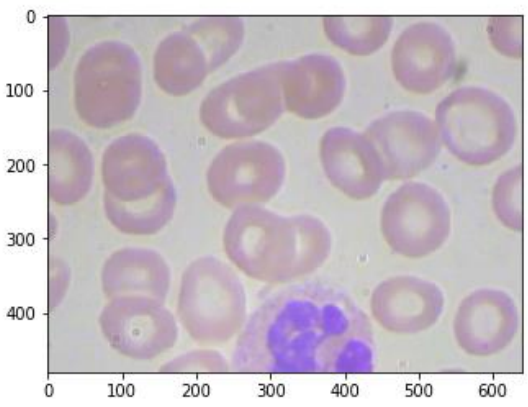


Fig12: Dataset of this Project

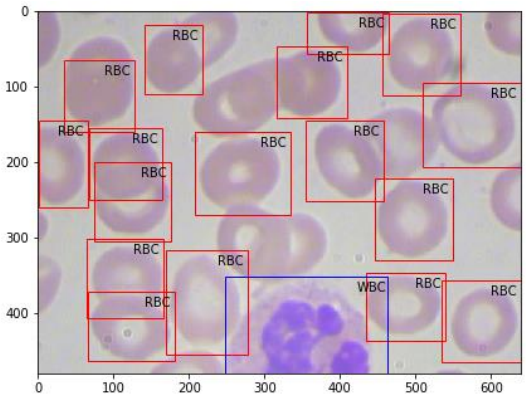
IMPORT LIBRARIES

Before we actually get into the model building phase, we need to ensure that the right libraries and frameworks have been installed. The below libraries are required to run this project:

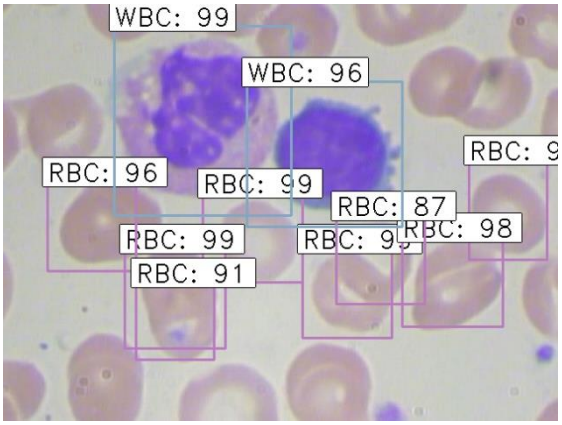
- pandas
- matplotlib
- tensorflow
- keras – 2.0.3
- numpy
- opencv-python
- sklearn
- h5py

DATA VISUALIZATION

```
fig = plt.figure()

#add axes to the image
ax = fig.add_axes([0,0,1,1])

# read and plot the image
image = plt.imread('images/1.jpg')
plt.imshow(image)

# iterating over the image for different objects
for _,row in train[train.image_names == "1.jpg"].iterrows():
    xmin = row.xmin
    xmax = row.xmax
    ymin = row.ymin
    ymax = row.ymax

    width = xmax - xmin
    height = ymax - ymin

    # assign different color to different classes of objects
    if row.cell_type == 'RBC':
        edgecolor = 'r'
        ax.annotate('RBC', xy=(xmax-40,ymin+20))
    elif row.cell_type == 'WBC':
        edgecolor = 'b'
        ax.annotate('WBC', xy=(xmax-40,ymin+20))
    elif row.cell_type == 'Platelets':
        edgecolor = 'g'
        ax.annotate('Platelets', xy=(xmax-40,ymin+20))

    # add bounding boxes to the image
    rect = patches.Rectangle((xmin,ymin), width, height, edgecolor = edgecolor, facecolor =

    ax.add_patch(rect)
```



RESULTS



SOFTWARE AND HARDWARE REQUIREMENTS

In this project we have used Windows 10, Intel Core i7-9700 CPU of 3.00 GHz, 32.0 GB RAM, and NVIDIA GeForce RTX 2080 graphics card.

CONCLUSION

R-CNN algorithms have significantly improved the performance of object detection tasks. The number of computer vision applications being developed has unexpectedly increased in recent years, and R-CNN is at the core of most of them.

ACKNOWLEDGMENTS

We would like to express our special thanks to our mentor Dr. H. Balaji who gave us a golden opportunity to do this wonderful project on this topic which also helped us in doing a lot of research and we came to know about so many new things. We are really thankful to them.

Secondly, we would also like to thank my friends who helped us a lot in finalizing this project within the limited time frame.