Chapter 2
Evaluating Differences in Small Object Localization Using Semantic Segmentation and Single Shot Detector (SSD) Bounding Box Algorithm

Rushikesh Chopade, Aditya Stanam, and Shrikant Pawar

Abstract In the computer vision task of semantic segmentation, each pixel in an image is classified into a particular class or category. Unlike object detection, which detects objects and provides bounding boxes, semantic segmentation assigns a class label to each individual pixel. This results in a pixel-wise classification map that identifies different objects or regions within an image. In the present study, we have tried to compare two very popular techniques used for detecting the object in an image using computer vision. Semantic segmentation and bounding box algorithms have been compared in a controlled environment, keeping the dataset, train-test split, batch size, training epochs, and other factors constant. We have compared these 2 techniques for loss minimization and area of overlap functions. A gradual decrease in the training loss has been observed in semantic segmentation technique, while the bounding box algorithms generate a steep decrease specifically in the later epochs. Choosing appropriate object detection techniques should address the problem of small regions of interest (ROI) compared to the total image area of class imbalance problem in semantic segmentation.

R. Chopade
Department of Geology and Geophysics, Indian Institute of Technology, Kharagpur, West Bengal 721302, India

A. Stanam
The University of Iowa, Iowa City, IA 52242-5000, USA
e-mail: aditya-stanam@uiowa.edu

S. Pawar (✉)
Department of Computer Science & Biology, Claflin University, Orangeburg, SC 29115, USA
e-mail: spawar@claflin.edu

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2025

J. C. Bansal et al. (eds.), *Congress on Smart Computing Technologies*, Smart Innovation, Systems and Technologies 396, https://doi.org/10.1007/978-981-97-8096-9_2

Copyrighted material

2.1 Introduction

In the computer vision task of semantic segmentation, each pixel in an image is classified into one of several predefined classes or categories. It is a fundamental problem in image understanding and has various applications, including autonomous driving, medical image analysis, and object recognition [1]. In semantic segmentation, the goal is to partition an image into regions and assign a class label to each pixel within those regions [2]. Unlike object detection, which localizes objects with bounding boxes, semantic segmentation provides a pixel-wise classification of objects and their boundaries [2]. This fine-grained level of understanding is essential for various computer vision tasks. Semantic segmentation can be performed using

for various computer vision tasks. Semantic segmentation can be performed using various techniques like Convolutional Neural Networks (CNNs), Fully Convolutional Networks (FCNs), CRFs (Conditional Random Fields), and transfer learning [3, 4]. The problem of small regions of interest (ROI) compared to the total image area is commonly known as a class imbalance problem in semantic segmentation. In this study, we have attempted to compare which technique (semantic segmentation and bounding box algorithms) gives better results in terms of loss minimization and a greater area of overlap when compared to a true ROI.

2.2 Materials and Method

Datasets

Digital Imaging and Communications in Medicine (DICOM) format chest radiograph images from the CANDID-PTX dataset are analyzed in this study [5]. From a total of 19,237 images, 335 acute rib fractures images are extracted for this study. Any rib that showed cortical damage on a chest radiograph without callus formation, an indication of healing, was classified as an acute rib fracture. There are a total of 973 different annotations provided by different radiologists for 335 images. The same image having different annotations by different radiologists are treated as different images for dataset enhancement. All images are 3D images with RGB channels.

The acute rib fracture data is supplemented with a semantic segmentation labeling file containing Run-Length Encoding (RLE) of the masks. Run length encoding is a simple morse-like representation of a 2D image [6]. The image is 3D while the RLE notations provided are based on 2D representation of the images. So selecting any one channel and projecting the mask obtained from the RLE needs to be done for getting the ROI. The 1024*The 1024 image is represented as a one-dimensional array with rows appended one after the other. The run length encoding provided for the images with the dataset contains a string of comma-separated numeric values. When the mask begins, the first number in the string is the pixel number followed by the lengths of the mask and background pixels alternatively. For each image, a unique mask has been derived from the RLE string. The row number and column

number of each pixel has been found by taking the integer part and the residual part when the string entries are divided by the pixel dimension (replication code of this data generation has been provided in the supplementary section). The conversion of RLE to masks is required for training the semantic segmentation algorithm. To make the notations compatible for the bounding box algorithm, the same RLE has been transformed to bounding box coordinates consisting of the minimum and maximum value for both coordinates of the bounding boxes.

The maximum and minimum pixel intensity value in an image defines the absolute white and absolute dark area in that image. In case of multiple images, the maximum and minimum value of the pixel in the image should be the same in order to help the algorithm understand the absolute white and absolute dark regions. While making the mask prediction the background is absolute 0 and the ROI is absolute 1. Thus, normalizing the pixel intensity would enhance the learning while training a semantic segmentation algorithm. In the CANDID-PTX dataset, the maximum pixel intensity in each image is different. So the pixel intensity of every image has been normalized in a range of 0 to 1 only for training the semantic segmentation algorithm. While training the two algorithms, we have kept all the training conditions similar. The batch size, image resolution, number of epochs, optimizer and its learning rate are kept the same in both the training processes. In addition, the train-test split has also been kept the same with the same random state to produce the same classification of images in the training set and the validation set.

Model architecture:

To compare both the techniques, following different model architectures and loss functions have been implemented:

U-Net Model Architecture for Semantic Segmentation Object Localization:

A standard transformer based U-Net architecture has been used with the binary focal loss for semantic segmentation algorithm. The primary reason for choosing U-Net architecture is the requirement of mask creation. Creating a mask is a generative AI problem and hence is addressed using a transformer based model. In the U-Net archi-

tecture is attached in the GItHub repository. The rib fracture dataset is split into 70–30% train-test split for the model training and validation. The image distribution in the training and test set has been kept constant using random state values equal to 0. The model has been trained using a mini-batch gradient optimization algorithm in batches of 4 using a custom image data generator. The algorithm is trained for 10 epochs using adam optimizer. The learning rate used in the optimizer is 0.001. The image is downsampled to 512*512 from 1024*1024 to reduce computational cost. Binary focal loss has been used to train and validate the semantic segmentation algorithm. The primary reason behind selecting a binary focal loss over other loss functions (Jaccard loss, Dice loss, etc.) is the small size of ROI (0.08–0.5% of the

Copyrighted material

total image area) in the acute rib fracture dataset. This is a class imbalance problem where the positive class/ROI covers a very tiny area as compared to the negative class/background area. In another study (data not shared), we have found that tuning the hyperparameters alpha (α) (0.01) and gamma (γ) (0.1) in the binary focal loss can efficiently handle this class imbalance problem. The mathematical representation of the implemented binary focal loss is shown below:

$$L(y, \hat{p}) = -\alpha y(1-\hat{p})\gamma \log(\hat{p}) - (1-y)\hat{p}\gamma \log(1-\hat{p})$$

where, y is ground truth pixel value; p^ is predicted pixel value; alpha (α) is the weighting factor which governs the tradeoff between the precision and recall by weighting errors for positive class, the modulating factor's focusing parameter, gamma (γ), indicates the extent to which forecasts with higher confidence levels

A VGG-16 feature extractor with pretrained ImageNet weights and an input size of 512*512*3 resolution has been used for the SSD model. In order to assist the algorithm learn more quickly, a feature extractor was used to gather object features in a particular order. To build the backbone of the algorithm, the rectified linear activation layer (ReLU) comes after the VGG-16 feature extractor and is followed by a dropout layer. To tackle the issue of large variation, a dropout layer with 25% dropout nodes has been employed. Following the application of the dropout layer, the resultant image has 16*16*512 resolution in its dimensions. A 2D convolutional layer, a ReLU activation layer, and a batch normalization layer are the compression layers that are added to the model architecture after the dropout layer. The model divides into two branches—a classification branch and a bounding box regression branch—after three compression layers. Using the provided labels, the classification branch divides the image into the appropriate classes. An activation layer using the sigmoid activation function and a flattening layer come after the 2D convolutional layer. After flattening, the classification breach's final output form equals 16. After flattening and using a similar strategy to the bounding box regression branch, the classification breach's final output shape equals 64. To obtain the final output shape with 16 bounding box predictions, the flattened layers of the classification and bounding box regression branches are combined. To produce a single confidence value, the non-max suppression technique is used. The graphical representation of the SSD architecture is provided in the supplementary file section in GitHub repository. To compute loss for the SSD algorithm, a custom cost function has been proposed. The cost function is divided into two sections: the first section deals with bounding box loss and the second part with classification loss. The intersection over union (IoU) policy serves as the foundation for the bounding box loss portion of the custom cost function. Similar to the semantic segmentation algorithm, mini-batch gradient descent technique has been used to optimize the algorithm. The training and test dataset have been fed to the algorithm in batches of 4. The images have been downscaled to 512*512, and the data is split into 70–30% train-test split with random state

7. Chopade, R., Stanam, A., Pawar, S.: Single shot detector application for image disease localization. Biorxiv (2021). https://doi.org/10.1101/2021.09.21.461307
8. Huynh, T., Nibali, A., He, Z.: Semi-supervised learning for medical image classification using imbalanced training data. Comput. Methods Programs Biomed. **216**, 106628 (2022). https://doi.org/10.1016/j.cmpb.2022.106628
9. Google Developers.: Descending into ML: Training and Loss (2023). Accessed from: https://developers.google.com/machine-learning/crash-course/descending-into-ml/training-and-loss