Detection Of Median Nerve

Mohammed Furqaan Khan

Abstract— The internship included understanding of deep learning and implementation of computer vision techniques. It included tasks in which I was assigned to add noise to the images of median nerve for variability of data. I labeled the images using the LabelMe [1] annotation tool. We used SSD-MobilNet [2] to train the model and get the desired results.

Index Terms— computer, detection, labelimg, median, nerve, object, ssd

1 INTRODUCTION

The information and work done was mostly experimental. I did not have access to the data set and had to work solely on one computer. I was first assigned to add noise to the images for more variability and robustness to the model. I then worked together in configuring anaconda, matplotlib, opencv etc.. a multitude of modules had to be downloaded and configured to begin the training process. I also learned about the working of the SSD-MobilNet, the dimensions of vectors, IOU and also learnt about convolutional neural networks for visual learning. The month of may included training the model. The training was successfully done and I was able to successfully import the checkpoints.. The final task of the two part project was to label the images using LabelMe which is used for image segmentation purposes. Segmentation is the process of dividing an image into different regions based on the characteristics of pixels to identify objects or boundaries to simplify an image and more efficiently analyze it. We use deep learning to work with our model. The design used for the object detection purpose is Single Shot Detector (SSD) owing to its sensible performance accuracy and high Speed images. The SSD-Mobilenet model provides the simplest accuracy and speed compensation which are applicable to all the object detection models.

2. AUGMENTATION

The images were augmented on a publicly available dataset and the codes were used for our dataset. The images were augmented to create variability to the dataset and robustness to the model. We added noise [3] to the images such as white noise, gaussian noise, changed the contrast in the images, randomly resized and cropped images(to get the zooming effect). Image data augmentation is used to expand the training data in order to improve the performance and ability of the model to generalize.

2.1 Adding Noise to images

The images we had did not capture the variability of median nerves in general, which is why we decided to use these augmentation techniques [4] to increase the robustness of our algorithm; it was necessary to add static or adaptive blocks of filter. For the purposes of image processing, we wanted to add random noise, such as white noise, Gaussian noise, that randomly changes some pixels to completely white or completely black. In some cases, we may speckle noise (uniformly add), only add white pixels (salt), or only add black pixels (pepper). White noise is a consistent level of noise added uniformly.

2.2 Brightness Augmentation

The brightness of the image can be augmented by either randomly darkening images, brightening images, or both. The intent is to allow a model to generalize across images trained on different lighting levels. The cause is to permit a model to generalize throughout photos educated on extraordinary lighting fixtures ranges. Values much less than 1.0 darken the picture, e.g. [0.5, 1.0], while values larger than 1 brighten the photograph, e.g. [1.0, 1.5], where 1.0 has no effect on brightness

2.3 Zoom Augmentation

For zooming effect we first randomly cropped one image's length, breadth and height, and then we resized it to its original size. A zoom augmentation randomly zooms the photo in and either provides new pixel values surrounding the image or interpolates pixel values respectively. A zoom augmentation randomly zooms the image in and both adds new pixel values around the image or interpolates pixel values respectively. e.g. [0.5,0.5] makes the object inside the photo 50% larger or closer, and values larger than 1 will zoom the photo out by 50%, e.g. [1.5, 1.5] makes the object within the image smaller or similarly away. A zoom of [1.0,1.0] has no effect.

3 LABELLING USING LABELME

The images used were a dataset in excess, it was a private dataset. LabelMe is a fairly simple tool which was used. I drew dotted bounding areas over the individual images of the median nerve. LabelMe is an annotation tool. We connect dots at the boundaries at the objects that we want to label for segmentation and it keeps the labeling information in a JSON file. It is used for marking various details. This task was completed by drawing precise contours around the median nerve for accurate detailing as it is tailored mainly for segmentation purposes. .

ISSN 2229-551

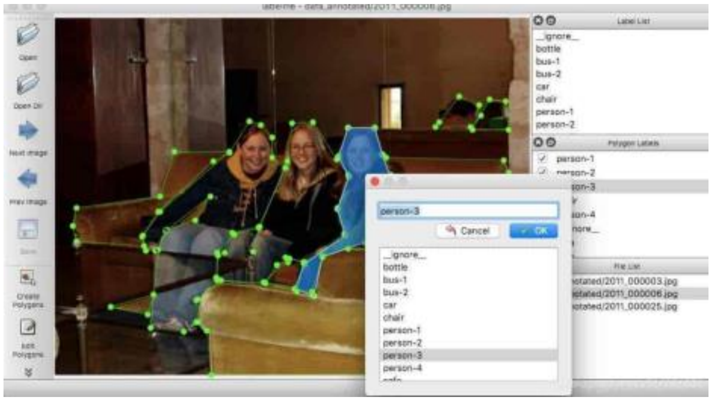

Figure 1 - : Example of the LabelMe interface

4 Training The Dataset Using SSD Mobilnet

The next task that was assigned to me was to train the dataset on the 400 original images(without pre-processing) on SSD-Mobilnet because it gives the best accuracy and speed compensation among the fastest detectors, SSD with Mobilenet was utilized for training. (SSD) is an acronym for Single Shot Multiple Box Detector. During training, the SSD simply requires one input image and ground truth frames for each item. At each location there exist multiple feature maps with varying scales, a small number of standard boxes with different aspect ratios. We anticipate form offsets and confidences for all object categories for each standard frame. We begin by matching these standard squares to the truth squares on the ground during training. The SSD-MobilNet model was implemented and trained on google collab. The goal of the model is to decide which of the default containers to use for a given photograph and then are expecting offsets from the chosen to default bins to achieve the final prediction. The structure of SSD consists of 3 fundamental additives: I) Base network ii) Greater characteristic layers iii) Prediction layers

4.1 Architecture

Per feature map location, SSD [5] discretizes the output space of bounding boxes into a set of default boxes at various aspect ratios and scales. The SSD method uses a feedforward convolutional network to generate a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to get the final detections. The early network layers are based on the base network, which is a common architecture for high-quality image categorization (truncated before any classification layers). Any basic image classification network starts with the base network, which is effectively the first layer. The MobilNet is the base network that we are using. At the end of the truncated base network, we add convolutional feature layers. These layers shrink in size over time, allowing detections to be predicted at several scales. Using a collection of convolutional filters, each new feature layer (or alternatively an existing feature layer from the base network) can yield a fixed set of detection predictions. Several feature layers are added to the end of a base network in our SSD model, which predict the offsets to default boxes of various scales and aspect ratios, as well as their related confidences.

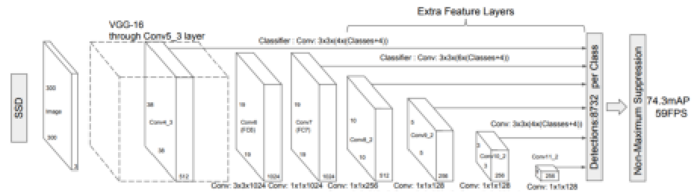

Figure 2 - Architecture of the SSD MobilNet

5 Conclusion

Using deep learning techniques and intensive training of the model, we were able to accurately detect median nerves and provide help to CTS patients and would be something that is useful and a benefactor when it comes to diagnostics. We also added gaussian noise to the images, in turn making sure that the model consists of variability, clarity and robustness. The processing formula proposed in our study can be integrated into the commercialized machine software, and introducing the newly acquired model knowledge in the daily clinical routine can be considered. We used the SSD-MobilNet which offered a good accuracy and speed trade off.

6 References

[1] T. Labelme
[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, SSD: Single Shot MultiBox Detector, Lecture Notes in Computer Science 1 (2016) 21–37. doi:10.1007/978-3-319-46448-0_2.
[3] A. A.-S. Selami, A. A. & Fadhil, A Study of the Effects of Gaussian Noise on Image Features, Kirkuk University Journal / Scientific Studies (2016).
[4] Stojnev, A. Ilić (2020)
[5] H.-T. & Choi, Lee, & Ho-Jun, Kang, & Hoon, Yu, & Sungwook, Park, Ho-Hyun (2021).