Application of Semantic Segmentation using OpenCV and Deep learning

Jagruth S[1], Dr. Prakash Biswagar[2], Shreya P Pejathaya[3], Dr. B Muthukumaraswamy[4], Shyamala A A[5], Mrs. Neethu S[6]

[1, 2] Electronics and Communications, R V College of Engineering, Bengaluru, India

[3,5,6] Telecommunication, R V College of Engineering Bengaluru, India

[4]Open Tech. Enquero Global, Bengaluru, India

Abstract- **One area that has attained a great progress in computer vison is Object detection using deep learning. This article proposes an article location framework that depends on a multi-locale completely convolutional neural system (FCN) that likewise encodes semantic division mindful highlights and dependent on U-net model.. Semantic segmentation or image segmentation is the unique way of clustering parts of an image together which belong to the same object class. One of the best examples of semantic segmentation is Self-Driving cars. Self-driving cars are also known as autonomous cars and they combine sensors and software to control and navigate the vehicle. The system is developed on python to detect vehicles. The system is trained and tested against images and videos. The system averaged at 20ms to process the image.**

Index terms- **Deep learning, fully convolutional neural network (FCN), Python, OpenCV, U-Net**

I.INTRODUCTION

In computer vision, picture division is the path toward isolating a propelled picture into different segments (sets of pixels, also implied as picture objects). The goal of division is to smooth out and also change the depiction of a picture into something that is continuously critical and more straightforward to ask about .[1] Picture division is typically need to discover things and cutoff focuses (lines, twists, etc.) in pictures. Even more unquestionably, picture division is the path toward giving out an imprint to each pixel in an image such pixels with a proportionate name share certain properties.

The eventual outcomes of picture division may have a ton of segments that everything considered spread the whole picture, or a social affair of shapes removed from the image. All of the pixels during a region are near in regards to some trademark or enrolled property, like concealing, force, or surface. Adjoining zones are in a general sense uncommon in regards to a proportionate characteristic(s).[1] When applied to a stack of pictures, normal in clinical imaging, the ensuing shapes after picture division are much of the time wont to make 3D amusements with the assistance of contribution figurings like strolling strong shapes.

II. LITERATURE SURVEY

Deep learning has increased an unbelievable impact on how the planet is adjusting to AI since recent years. Some of the supported item location calculations are Region based Convolutional Neural Networks (R-CNN), Faster-R-CNN, Single Shot Detector (SSD) and You Only Look Once (YOLO). Among these, Faster-R-CNN and SSD have better precision, while YOLO has better performs when speed is given inclination over exactness. Profound learning consolidates SSD and Mobile Nets to perform proficient usage of location and following. This calculation performs proficient item discovery while not settling on the performance.[2]

The crude AI calculations that are available separate every issue into little modules and fathom them exclusively. These days prerequisite of detection calculation is to figure end to complete and set aside less effort to process. Continuous identification and arrangement of items from video records give the motivation to creating numerous sorts of diagnostic viewpoints like the amount of traffic during a specific zone throughout the years or the whole populace in an area. By and by, the assignment generally experiences moderate preparing of arrangement and discovery or the event of wrong location on account of the consolidation of little and light-weight datasets. To beat these issues, YOLO based recognition and grouping approach (YOLOv2) for improving the calculation and preparing speed and at an equal time proficiently distinguish the items inside the video records. The characterization calculation makes a jumping box for each class of articles that it is prepared, and creates an explanation portraying the genuine class of item. The YOLO based recognition and characterization (YOLOv2) utilization of Graphics Processing Unit (GPU) to expand the calculation speed and procedures at 40 casings for each second.[3]

In [4] present YOLO, a replacement approach to manage object location. Earlier work on object affirmation repurposes classifiers to perform distinguishing proof. Or on the other hand perhaps, it marks object affirmation as a lose the faith issue to spatially withdrew skipping boxes and related class probabilities. One neural framework predicts bouncing boxes and refinement probabilities really from full pictures in a solitary appraisal. Since the whole territory pipeline could even be one system, it is routinely top tier as far as possible truly on ID execution. Our bound together plan is extremely snappy. Our base YOLO model techniques pictures perpetually at 45 lodgings for each second. A humbler change of the system, Fast YOLO, structures a confounding 155 lodgings for each second while so far accomplishing twofold the mAP of other advancing locators. Stood apart from top level zone frameworks, YOLO submits greater control mistakes yet could in addition be an increasingly little aggregate in danger to foresee bogus positives on foundation. At long last, YOLO learns general delineations of things.

Top tier object revelation frameworks depend on district recommendation estimations to hypothesize object territories. Advances like SPPnet and Fast R-CNN have decreased the period of time of those distinguishing proof frameworks, revealing area suggestion computation as a bottleneck. During this work, it presents a Region Proposal Network (RPN) that offers full-picture convolutional features with the acknowledgment sort out, as such engaging nearly sans cost district suggestion. A RPN may be a totally convolutional orchestrate that at the same time predicts object limits and degradation scores at each position. The RPN is readied all the way to get first rate region proposals, which are used by Fast R-CNN for acknowledgment. It further union RPN and Fast R-CNN into one framework by sharing their convolutional features using the starting late notable expressing of neural frameworks with 'thought' instruments, the RPN part tells For the significant VGG-16 model, our disclosure structure incorporates an edge pace-of 5 fps (checking all methods) on a GPU, while achieving top tier object area precision on PASCAL VOC 2007, 2012, and MS COCO datasets with only 300 suggestions for each picture.[5]

Conventional article locators initially produce recommendations. at that point the highlights are removed. At that point a classifier on these proposition is executed. Be that as it may, the speed is moderate and in this way the exactness isn't fulfilling. YOLO a wonderful article discovery approach bolstered profound learning presents one Convolutional Neural Network (CNN) for area and grouping. All the completely associated layers of YOLO's system are supplanted with a mean pool layer for the point of replicating a substitution organize. The misfortune work is upgraded after the extent of bouncing directions mistake is expanded. A substitution object recognition strategy, Optimized You Only Look Once (OYOLO), is delivered, which is 1.18 occasions quicker than YOLO, while beating other area based methodologies like R-CNN in precision. To upgrade precision further, it includes the blend of OYOLO and R-FCN to our framework. For testing pictures in evenings, pre-preparing is introduced utilizing the histogram adjustment approach.[6]

In a cross breed camera framework joining an omnidirectional and a Pan-Tilt-Zoom (PTZ) camera, the omnidirectional camera gives 360-degree flat field-of-see, while the PTZ camera gives high quality picture at a specific heading. This prompts a decent field-of-view and high quality camera framework. It exploits this half and half framework for constant item grouping and following for traffic scenes. The omnidirectional camera identifies the moving items and plays out an underlying order utilizing shape-based highlights. Simultaneously, the PTZ camera orders the items utilizing high-goals outlines and Histogram of Oriented Gradients (HOG). PTZ camera additionally performs high-goals following for the articles grouped in light of the fact that the objective class by the omnidirectional camera. The thing types it took a shot at are passerby, bike, vehicle and van. Broad investigations were directed to coordinate the order precision of the mixture framework with single camera alternatives.[7]

III. METHODOLOGY

The semantic segmentation system is developed using Deep Learning and OpenCV. The algorithm used for semantic segmentation is a convolutional neural network, this was selected as it the best neural network for image processing. The setting up of the neural network and functioning of OpenCV is explained below.

A. Convolution Neural Network

One of the category of DNN that is CNN is applied to analyze visual images. They are also mentioned as shift invariant or space invariant artificial neural networks, supported their shared-weights architecture and translation invariance characteristics. They main applications are in image and video recognition, recommender systems, image classification, medical image analysis, tongue processing, and financial statistic.

CNN are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all or any neurons within the next layer. The “fully-connectedness” of those networks makes them susceptible to overfitting data. Typical ways of regularization include adding some sort of magnitude measurement of weights to the loss function. CNN adopt an exceptional strategy towards regularization: they money in of the various leveled design in information and amass increasingly complex examples utilizing littler and more straightforward examples. Therefore, on the size of connectedness and complexity, CNN are on the lower extreme.

CNN suggests that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design may be a major advantage.

A CNN consists of an input and an output layer, also has multiple hidden layers. The hidden layers of a CNN typically contain a series of convolutional layers that convolve with a multiplication or other scalar product. The enactment work is generally a Rectified liner Unit layer and is in this way followed by extra convolutions like pooling layers, completely associated layers and standardization layers, referenced as concealed layers on the grounds that their sources of info and yields are veiled by the actuation capacity and last convolution.

Though the layers are colloquially mentioned as convolutions, this is often only by convention. Mathematically, it is technically a sliding scalar product or cross-correlation. This has significance for the indices within the matrix, therein it affects how weight is decided at a selected index point.

B. OpenCV

OpenCV is a library of programming functions mainly focused towards computer vision. Originally developed by Intel, it had been later supported by Willow Garage then Itseez (which was later acquired by Intel). The library is cross-platform and free to be used under the open-source BSD license.

OpenCV supports some models from deep learning frameworks like TensorFlow, Torch, PyTorch (after converting to an ONNX model) and Caffe consistent with an outlined list of supported layers. OpenCV promotes OpenVisionCapsules, which is a portable format and also compatible with all other formats.

Some of the applications of OpenCV are:

- Motion detection
- Segmentation and recognition
- Motion tracking and so on

OpenCV is written in C++ and its essential interface is in C++, however it despite everything holds a less far reaching however broad more seasoned C interface. There are bindings in Python, Java and MATLAB/OCTAVE. The Application Program Interface (API) for these interfaces are often found within the online documentation. Wrappers in other languages like C#, Perl, Ch, Haskell, and Ruby are developed to encourage adoption by a wider audience.

Since version 3.4, OpenCV.js is a JavaScript binding for selected subset of OpenCV functions for the online platform.

All the new developments and algorithms in OpenCV are now developed within the C++ interface.

C. U-net

U-Net is a CNN that was developed for biomedical image segmentation at the Computer Science Department of the University of Freiburg, Germany. The network is predicated on the FCN and its architecture was modified and extended to be used with fewer training images and to yield more precise segmentations. Segmentation of a 512x 512 image takes but a second on a contemporary GPU.

The U-Net architecture stems from the so-called FCN first proposed by Long and Shelhamer.[8]

The primary thought is to enhance a typical contracting system by progressive layers, where pooling activities are supplanted by upsampling administrators. Thus these layers increment the goals of the yield. What is more, a successive convolutional layer can then learn to assemble a particular output supported this information.

One significant alteration in U-Net is that there are an outsized number of highlight channels inside the upsampling part, which license the system to spread setting data to higher goals layers. As a result, the far reaching way is pretty much symmetric to the contracting part and yields a u-formed design. The network only uses the valid part of each convolution without fully connected layers.[8] To predict the pixels within the border region of the image, the missing context is extrapolated by mirroring the input image. This tiling strategy is vital to use the network to large images, since otherwise the resolution would be limited by the GPU memory.

The network consists of a contracting path and an expansive path, which provides it the u-shaped architecture. The contracting way could likewise be a run of the mill convolutional organize that comprises of rehashed utilization of convolutions, each followed by an amended long measure (ReLU) and a maximum pooling activity. During the withdrawal, the spatial data is diminished while highlight data is expanded. The extensive pathway joins the component and spatial data through an arrangement of up-convolutions and links with high-goals highlights from the contracting way.

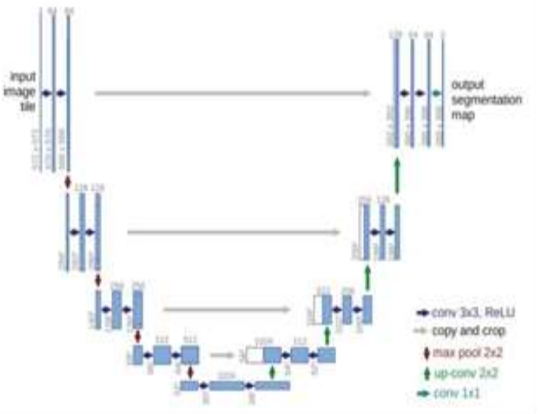

Fig. 1. U-net architecture [9]

The Fig. 1 show the architecture that was used as an inspiration for the project. The u-net is convolutional specification for fast and precise segmentation of images. Up to now it is beated the earlier best strategy (a sliding-window convolutional arrange) on the ISBI challenge for division of neuronal structures in microscopy stacks. It has won the Grand Challenge for Computer-Automated Detection of Caries in Bitewing Radiography at ISBI 2015, and it has won the Cell Tracking Challenge at ISBI 2015 on the two most testing transmitted light microscopy classes (Phase differentiation and DIC microscopy) by an outsized edge. [4]

D. Dataset Preparation and Augmentation

Cityspace dataset is used, it is 4.5 GB in size and is made from casings gathered from two of recordings while driving the vehicle around Mountain View zone in overwhelming rush hour gridlock. The informational collection contained a name document with bounding boxes stamping different vehicles, trucks and people on foot. The whole dataset contained around 22000 pictures. The vehicles and trucks were joined into one class vehicle and dropped all the bounding boxes for people on foot. For the most part in light of the fact that the quantity of vehicles far surpassed the quantity of trucks and people on foot in the informational collection.

The dataset is first divided into two i.e. training dataset and testing dataset. Training dataset consists of 20000 pictures and testing dataset consists of 2000 pictures. The training dataset is augmented before using it for training the network. Augmentation is done to increase the dataset size to improve the network. Three types of augmentation are used which are:

- Stretching: This type of augmentation is done by stretching a part of the picture to the size of the original picture size. This type of augmentation is done so the network can detect different size cars.
- Translation: This type of augmentation is done by moving the original picture in X or/and Y axis. This type of augmentation is done so that the network can detect cars in almost any part of the picture.
- Brightness: This type of augmentation is done by changing the brightness of the picture. This done so that the network can detect cars in different lighting conditions.

After the training dataset is augmented the next step involves target set preparation. Here a mask is used to detect the cars in the pictures. These masks are used by the network while training to decide whether network has detected the cars or not. This mainly done by bounding box method.

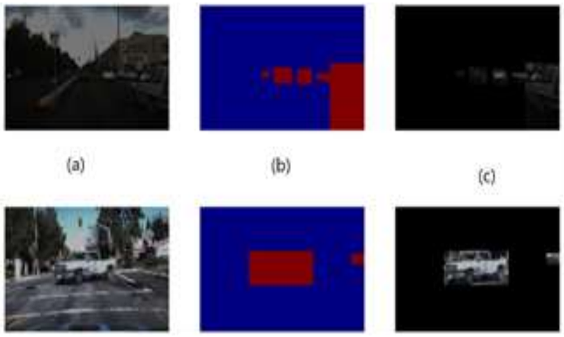

Fig. 2. Target set preparation (a) actual image from dataset (b) image of mask used (c) image to check if mask is valid or not

Fig. 2 implies few examples of the mask used and checking the validity of the mask.

E. Training the neural network

Training the network requires two main parameters to be set those are learning rate and number of iterations. While setting the number of iterations one should consider the time required to train the network. With increase in number of iterations the efficiency of the network and time required to train it increases. The number of iterations was set to 10000 for optimal efficiency and training time.

The parameter is to set learning rate, here one should use Stochastic Gradient Descent (SGD). SGD is a technique to enhance the effectiveness of the system by varying the learning rate. The method employed is Adam which takes advantage of two SDG techniques namely AdaGrad and RMSProp. In this technique running averages of both the gradients and the second moments of the gradients are used.

The standard parameter values recommended for Keras while using Adam are learning rate at 0.001, beta 1 at 0.9, beta 2 at 0.999, epsilon at 1e-08.[20] All the standard values were kept same expect for learning rate which was lowered to 0.0001 to make the neural network more efficient.

IV. RESULTS AND DISCUSSION

The results of the implemented work are discussed in this section. Packages such as Numpy, Keras and Matplotlib are used to implement the network and plot masks to detect cars in the pictures.

The system was trained for two hours using the training dataset of the Cityspace dataset. The predicted mask is compared to the ground truth mask for checking the correctness of the system.

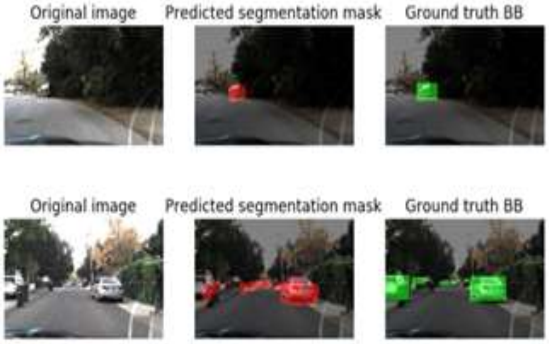

Fig. 3. Compression of the predicted mask and ground truth mask of training dataset

Fig. 3 gives comparison between the predicted mask given by the system to the ground truth mask while training the system. From the compression the predicted mask resembles the ground truth mask.

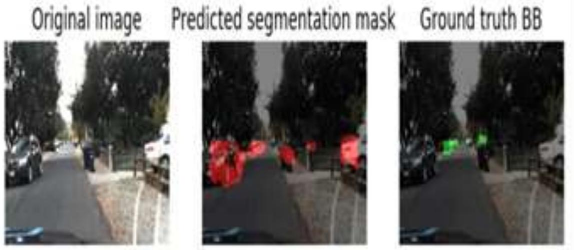

Fig. 4. Compression of the predicted mask and ground truth mask of testing dataset

Fig. 4 shows an example of testing dataset picture with predicted mask and ground truth mask. Here the predicated mask and ground truth mask do not resemble. The ground truth mask ignores the car on left lane and the car that is parked, in real case scenario the system should have the ability to detect all cars which is shown by the trained network.

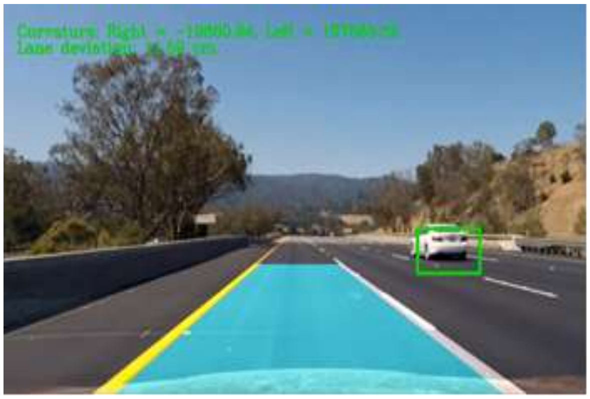
Curvature: Right = [illegible], Left = [illegible]
Lane deviation: [illegible] cm

Fig. 5. Lane deviation, curvature and car tracking in video

Fig. 5 is a screenshot of a video output given by the modified program. Here the program keeps track of cars, lane curvature and lane deviation. These tracking can be used to guide the car for self-driving cars.

V. CONCLUSION

Semantic image segmentation is a key application in image processing and computer vision domain. Besides briefly reviewing on traditional semantic image segmentation, the project also talks about u net model for object detection. FCN presents a simple and scalable object detection algorithm Object detection was performed with the help of City Space Dataset. Total images in the dataset was 22000 which included 20000 training images and 2000 testing images. Hardware requirement for the project is: Inteli7, 8th Gen quad core at 1.8 GHz, NVIDIA GPU components Distribution: Anaconda Navigator. The software requirements for the project are Jupyter Notebook. The frameworks include Tensor Flow and Pytorch. The objectives of the project are as follows to understand the concept of semantic segmentation, understand and review concepts of object detection algorithms, mathematical models and selecting the most suitable model and last objective is to analyze the data obtained with respect to IoU.

Through this project we understood the concepts of semantic segmentation and deep learning. U-Net model was understood and was applied for object detection. Semantic Segmentation for object detection was implemented and results were analyzed with respect to IoU. First the dataset preparation was done. Further the dataset was segregated for training and testing. Three types of augmentation such as Stretching, Translation and Brightness was performed.

Finally, object detection was performed on the given dataset. The object detection was performed using semantic segmentation and the results were marked. Parameters such as curvature right/left, Lane deviation was measured. Accuracy of 0.87 was obtained for object detection with respect to IoU.

REFERENCES

[1] G. S. Linda G. Shapiro, Computer Vision. Pearson Education (US), Jan. 23, 2001, 608 pp., isbn: 0130307963. [Online]. Available: https://www.ebook.de/de/product/3246508/linda_g_shapiro_george_stockman_computer_vision.html.

[2] G. Chandan, A. Jain, H. Jain, and Mohana, “Real time object detection and tracking using deep learning and OpenCV,” in 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE, Jul. 2018. doi: 10.1109/icirca.2018.8597266.

[3] A. P. Jana, A. Biswas, and Mohana, “YOLO based detection and classification of objects in video records,” in 2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE, May 2018. doi: 10.1109/rteict42901.2018.9012375.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Jun. 2016. doi: 10.1109/cvpr.2016.91.

[5] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-CNN: Towards real-time object detection with region proposal networks,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, Jun. 2017. doi: 10.1109/ tpami.2016.2577031.

[6] J. Tao, H. Wang, X. Zhang, X. Li, and H. Yang, “An object detection system based on YOLO in traffic scene,” in 2017 6th International Conference on Computer Science and Network Technology (ICCSNT), IEEE, Oct. 2017. doi: 10.1109/ iccsnt.2017.8343709.

[7] I. Baris and Y. Bastanlar, “Classification and tracking of traffic scene objects with hybrid camera systems,” in 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), IEEE, Oct. 2017. doi: 10.1109/itsc.2017. 8317588.

[8] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 4, pp. 640–651, Apr. 2017. doi: 10.1109/tpami.2016.2572683.

[9] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in Lecture Notes in Computer Science, Springer International Publishing, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[10] Z. Xiaoqing, “Efficient and balanced virtualized resource allocation based on genetic algorithm in cloud,” in 2017 10th International Symposium on Computational Intelligence and Design (ISCID), IEEE, Dec. 2017. doi: 10.1109/iscid.2017.187.

[11] Weikun Wang and Giuliano Casale, “Evaluating Weighted Round Robin Load Balancing for Cloud Web Services,” 16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, 2014.