***Abstract* - Semantic segmentation, the process of assigning semantic labels (car, background, grass, etc) to each point within an image, is a crucial part of robots ability to sense what is around them. In this work, we attempt to incorporate our understanding of Hidden Markov Models to better sense the semantics of a frame within a video. Traditionally, semantic segmentation is done in an instantaneous way, adding the dimension of time aims to improve on the traditional versions of semantic segmentation.**

Introduction

As robots compete to execute more complex tasks, their surroundings become increasingly complex. A robot's ability to perform these complex tasks is entirely dependent on their ability to understand the world around them. This dilemma of “how to sense?” is something we talked about deeply within the first half of class and will be the umbrella that encapsulates this paper. Under this umbrella we explore the integration of Hidden Markov Models (HMMs) with semantic segmentation algorithms, aiming to enhance performance, particularly in autonomous vehicle navigation scenarios.

Semantic segmentation is commonly used in fields such as autonomous vehicles, medicine or agriculture, allowing cars, doctors and farmers to identify items easily [5]. Traditional semantic segmentation was trained strictly through convolutional neural networks (CNN). The layers of the convolutional neural network allow for the model to capture high-dimensional features of the image as well as non-linear relationships between features [see methodology]. This approach heavily relies on the quality of the training model - a constraint of many models.

Strictly relying on a model's training quality can result in incorrect classifications of class instances or certain semantic labels. If not addressed, these mishaps due to training bias or limited training sets can become more detrimental. Rather than relying on larger datasets and further constraining these models, new and non-linear ways to improve these models should be implemented. We propose, especially for models on the edge (i.e. engaging directly with humans) that have comparatively lower computational power, to add the dimension of time and utilize HMMs to improve semantic segmentation.

Humans currently use dynamics in order to process daily scenarios such as crossing a street. In order to cross we would have to process the speed other people walk at, and the speed of the cars at the intersections, leaving little static objects. As robots integrate more and more into the human world and their environments become more dynamic, the models that they use will need to be more adaptable and dynamic. Hence, it is critical that these models can effectively understand the dynamics around them to engage in these environments more effectively.

Contributions

In our work, we utilize the common ResNet network and aim to enhance its performance in dynamic environments through the addition of a HMM. Our model's segmentation of the image will be dependent on previous observations - just like humans. Rather than treating each frame as a standalone, the model will incorporate previous observations to determine the

likelihood of current observations. We propose to integrate the HMMs ability to understand systems dynamics as the network classifies pixels within an image. Our findings aim to contribute to advancing the understanding and practical implementation of semantic segmentation methods, informed by a synthesis of existing research and novel insights gained through our experimentation.

Related Work

Semantic Segmentation: Current segmentation techniques involve running a video through a model such as UNet or Resnet, having it classify the pictures based on colors and then outputting a mask for the desired images we would like to separate. A larger scale version would be to use a dataset that consists of test, validation and train data as well as annotations of the labels. The accuracy is then tested to see if it correctly applies the labels based on what was provided from the test folder.

HMM: Hidden Markov models is a model that utilizes states and observations. There is a transition matrix that helps show the movement from one state to another as well as an emission matrix which shows the likelihood of each state occurring. In regards to semantic segmentation, sometimes 2D-HMM is used to help segment images of noisy classes. [2] In addition, it can also help find the relation between two different states within an image through diagonal transformations. This can help analyze images with more complex features as it captures the spatial dependencies between the images.

Related Papers:
Semantic SLAM for Dynamic Scenes: This paper improves semantic segmentation within SLAM by detecting and filtering out moving objects within its environment.

A General Two-Dimensional Hidden Markov Model and its Application in Image Classification: This paper extends the traditional 2D-HMM by incorporating diagonal state dependencies while ensuring causality, and proposes corresponding algorithms for training (EM and GFB) and classification (2D Viterbi). The application to aerial image segmentation demonstrates the superiority of the proposed model compared to existing 2D-HMM approaches.

Methodology

Foremost, it is important to note that we pursued semantic segmentation (classification of objects by class) as opposed to instance segmentation (differentiate each object within each class) or panoptic segmentation (a synergy of semantic and instance segmentation).

In seeking how best to implement our proposal, we delved into understanding semantic segmentation and CNNs. To preface, we concluded that CNNs were an area that would not be able to innovate as it is a highly complex and rapidly evolving field of research. Nonetheless, we believe it is important to understand the underlying CNN within our overall model.

A:1. Introduction to Convolutional Neural Networks

Introduced in 2012, CNNs have proven to be the optimal way to process grid-like data, such as images, and have revolutionized the field of computer vision, enabling breakthroughs in tasks like image classification, object detection, and semantic segmentation. Unlike regular neural networks, convolutional networks use specialized layers that apply small sets of weights, also known as filters or kernels, to different parts of the input image. This allows the network to learn and recognize local patterns, such as edges or textures, regardless of their position in the image. Moreover, CNNs reduce the number of parameters by sharing these weights across the entire image, making the network more efficient and easier to train.

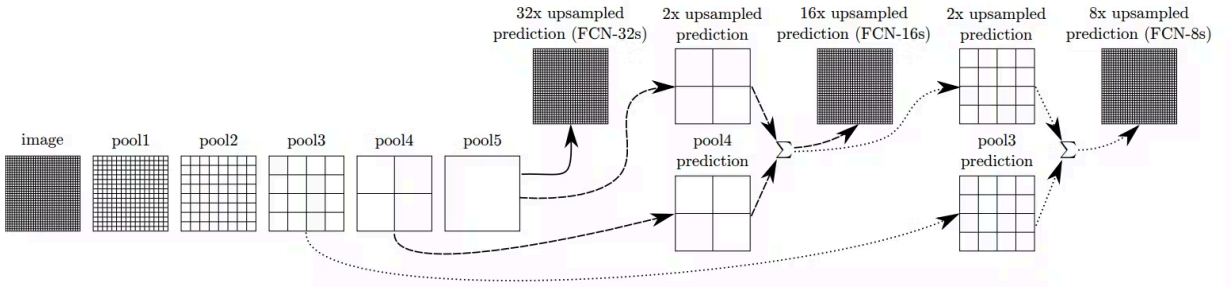


Fig 1. Fully Convolutional Network for Semantic Segmentation

These architectural differences allow CNNs to efficiently learn hierarchical features and scale well to high-dimensional data, making them the preferred choice for a wide range of computer vision tasks.

A:2. Choosing Convolutional Neural Network

The step was to choose architecture. There are many new models including the earlier mentioned SAM, we decided to use the “classic” ResNet architecture that consists of convolution layers along with batch normalization and ReLU activation functions. It uses skip connections in order to optimize the amount of layers the image goes through. In addition, it utilizes a bottleneck architecture that helps reduce the features and complexity, making it ideal for segmentation.

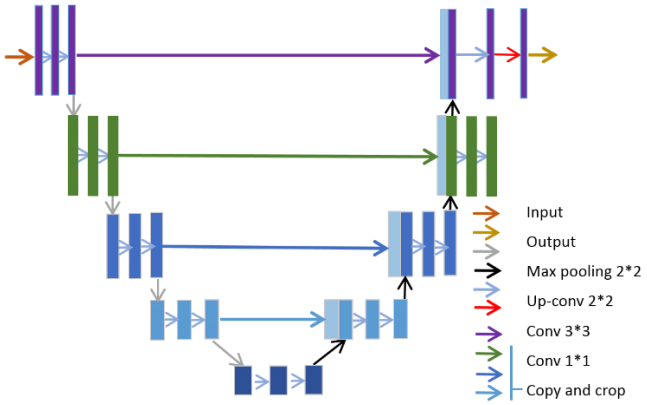


Fig 2. UNet Architecture

B. Semantic Segmentation

In this work, we used DeepLabs Deep Convolutional Neural Network (DCNN) ResNet-101 model to initialize our semantic segmentation.

1) *Feature Extraction*

The image is first passed through the REsNet-101 backbone network, consisting of multiple residual blocks. The output of the the residual blocks can be understood as

$$X_k = F(x_{k-1}) + x_{k-1},$$

where F(x_(k-1)) is equal to the residual function composed of convolution layers, batch normalization, and activation functions.

2) *Atrous Spatial Pyramid Pooling (ASPP)*

Receiving a feature map output from the ResNet-101 backbone, the information goes through the atrous spatial pyramid pooling (ASPP) module. The purpose of the ASPP module is to capture multi-scale contextual information by applying atrous convolutions with different dilation rates.

$$y[i, j] = \sum_{m,n} x[i + r \cdot m, j + r \cdot n] \cdot w[m, n]$$

in the equation w is the kernel weights and r is the dilation rate.

3) *Bilinear Upsampling*

A common problem within CNNs is the downsampling aspect. In order to combat this, our chosen model performs bilinear interpolation-based upsampling to make sure the output's dimensions match the original input. Bilinear interpolation achieves this upsampling by interpolating pixel values based on the surrounding pixels.

4) *Final Convolution Layer and Softmax Activation*

The final step is passing the upsampled feature map through a softmax activation function to obtain class probabilities. This is the classification step.

The result of these steps is state-of-the-art performance in semantic segmentation.

C. Hidden Markov Model

Once applying a traditional semantic segmentation model, our innovation occurs. Understanding that if in one frame there is a cluster of pixels labeled as a car, in the next frame the nearby pixels have a high likelihood of being a car as well. With this in mind, we implement a hidden markov model with the state being the location of objects in pixel space and

observations being the segmentation returned from the CNN. We have a high transition probability for the car class and a low transition probability for other classes.

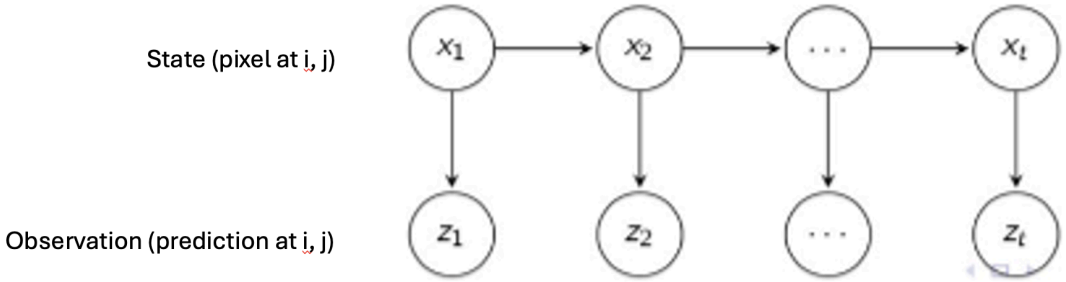


Figure 3. Hidden Markov Model for our Problem (Lecture Notes)

By incorporating HMMs into the segmentation process, we can predict smoother and more coherent outputs, thus improving the overall performance and reliability of semantic segmentation techniques.

D. Additional Highlights

1) We utilized batch processing in order to improve the efficiency of training.

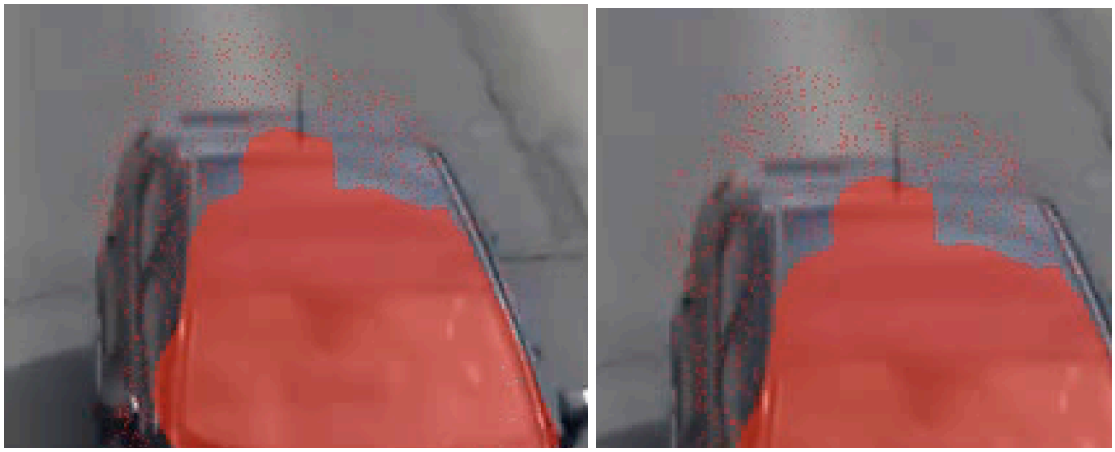
Experiments & Results

Through a series of experiments and analyses, we assess the impact of this integration on segmentation accuracy and its potential benefits for real-world applications. In our analysis - since we performed this on sampled videos, was by eye. Nonetheless, we are able to see the difference that the markov model makes in segmentation. We have included full videos of these experiments in the appendix.

Within the methodology it should be clear to see that the transition probability determines the likelihood of a pixel's label changing. A high transition probability represents the HMM will heavily favor previous labels, and the inverse is true for low transition probabilities.

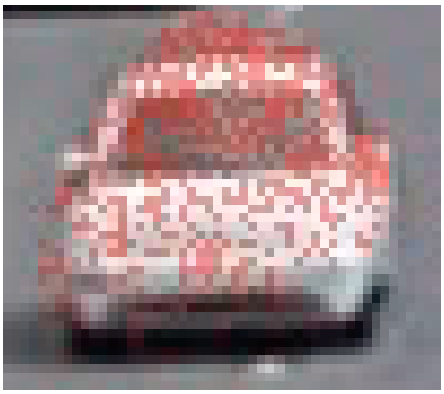
Experiment Details:
The parameter of interest was the transition probabilities. The following shows the progression of the models we tested. We have attached screenshots and respective videos.

Stage 0) No HMM
Stage 1) Transition Probability = 0.1 | Transition Probability = 0.01 | Transition Probability = 0.06



Slight difference in the number of persistent pixels (right side has less than the left due to a higher transition probability.

Stage 2) Transition Probability Car = 0.9 & Transition Probability Other = 0.1



Stage 3) Transition Probability Car = 0.9 & Transition Probability Other = 0.1 & Nearest Neighbor = 20 Pixels

Conclusion/Future Work

Through this exploration, we sought not only to expand our understanding of segmentation methodologies but also to assess the feasibility and effectiveness of integrating HMMs. By evaluating the impact of this integration, we aimed to contribute valuable insights to the field of computer vision, potentially paving the way for advancements in semantic segmentation techniques.

Subsequently, we explore the integration of hidden Markov models (HMMs) with semantic segmentation algorithms, aiming to enhance performance, particularly in autonomous vehicle navigation scenarios. We were able to find a method that helped in smoothing the process for segmentation in order to get slightly better results for the masking of the results.

In the future we may want to apply this technique to a larger dataset that consists of train, validation and test data. Currently we just ran on one mp4 video so we know it works for this video. However, we later noticed that it may not be the case for other videos with smaller objects such as a tennis match. Having a larger dataset can help confirm our model works in many different fields. With these datasets, there is also a possibility of having accuracy scores attached so that we are not just using visuals to test if there is a change. In addition, it may be good to test different neural network architectures to see how HMM improves segmentation for those options.

References

[1] Feiya Li, Chunyun Fu, Dongye Sun, Jian Li, and Jianwen Wang. "SD-SLAM: A Semantic SLAM Approach for Dynamic Scenes Based on LiDAR Point Clouds." arXiv preprint arXiv:2402.18318 (2024).

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. "Deep Residual Learning for Image Recognition." arXiv preprint arXiv:1512.03385 (2015)

[3] Baumgartner, Josef & Flesia, Ana & Gimenez, Javier & Pucheta, Julian. (2013). A New Approach to Image Segmentation with Two-Dimensional Hidden Markov Models. Proceedings - 1st BRICS Countries Congress on Computational Intelligence, BRICS-CCI 2013. 10.1109/BRICS-CCI-CBIC.2013.43.

[4] X. Ma, D. Schonfeld and A. Khokhar, "A General Two-Dimensional Hidden Markov Model and its Application in Image Classification," 2007 IEEE International Conference on Image Processing, San Antonio, TX, USA, 2007, pp. VI - 41-VI - 44, doi: 10.1109/ICIP.2007.4379516.

[5] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. "Segment Anything." arXiv preprint arXiv:2304.02643 (2023).

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. "Rethinking Atrous Convolution for Semantic Image Segmentation." arXiv preprint arXiv:1706.05587 (2017).