**A REPORT**

**ON**

**‘Unsupervised Learning of Image Segmentation Based on Differentiable Feature Clustering’**

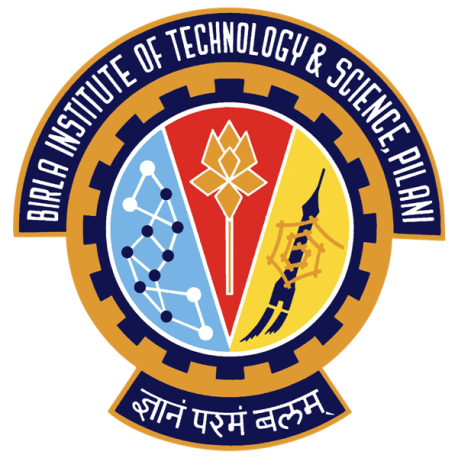
**BY**

| Name of the Student | ID Number |
| --- | --- |
| ***GOLI NAGA SANDESH*** | ***2019AAPS0346H*** |
| ***SRI MIHIR DEVAPI UNGARALA*** | ***2019AAPS0306H*** |
| ***GORIPARTI SAIVINAY*** | ***2019AAPS0255H*** |
| ***CHALLA SAI RESHWANTH*** | ***2019A3PS0433H*** |
| ***A RAHUL*** | ***2019A7PS1312H*** |
| ***SHAIK MOHAMMED SHAQUIB*** | ***2019A7PS1325H*** |

Prepared on completion of the

Machine Learning Course No. BITS F464

**AT**



**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

**BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE**

**PILANI (RAJASTHAN)**

**Title of the Project**:

“Unsupervised Learning of Image Segmentation Based on Differentiable Feature Clustering”

**Key Words**: Checkout, Automated Test Equipment, MIL-STD-1553, RS-232, DIO, Interface Control Document

**Project Areas**: Image Segmentation

**INTRODUCTION**

For decades, image segmentation has been a focus of computer vision research. Object detection, texture recognition, and image compression are all image segmentation applications. A set of pairings of photos with pixel-level semantic labels, such as "sky" or "bicycle," is used for training in supervised image segmentation.

The objective is to train a system that classifies the labels of the image pixels into recognised categories. This study looked into using convolutional neural networks (CNNs) for unsupervised picture segmentation. The proposed method is similar to supervised picture segmentation.CNN assigns labels to pixels that indicate which cluster they belong to. However, there are no training images or ground truth labels for pixels in unsupervised image segmentation. As a result, once a target image is supplied, the pixel labels and feature representations are optimised together. The gradient descent algorithm updates their parameters. A technique in which many unique cluster labels are needed is also proposed to help in cluster separation. Finally, the following three criteria for cluster label prediction are introduced:

(a) Pixels with similar properties should be labelled together.

(b) The same name should be applied to spatially continuous pixels.

(c) There should be a high number of distinct cluster labels.

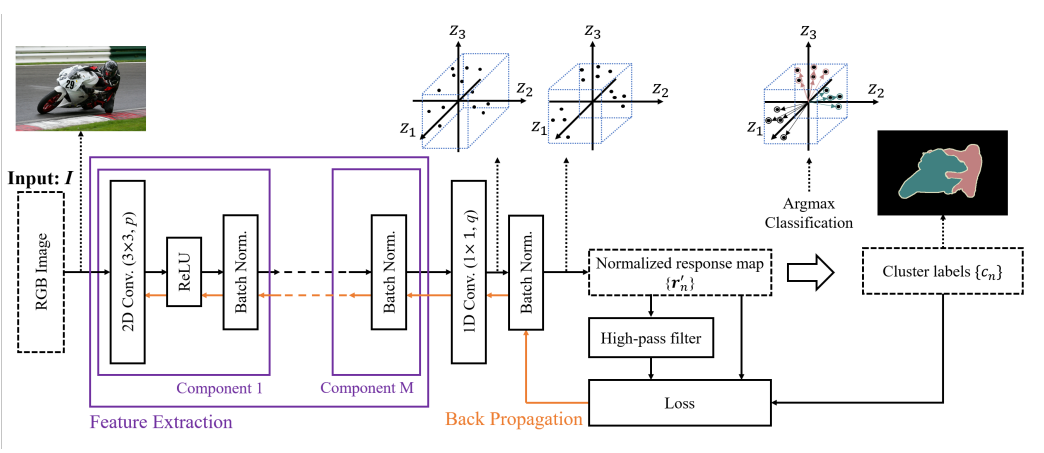
Even though these criteria are conflicting, the proposed approach minimises the combination of similarity loss and spatial continuity loss to discover a reasonable label assignment solution that meets all of the criteria above. This research makes four significant contributions:

1. We propose a novel unsupervised image segmentation end-to-end network that includes normalisation and an argmax function for differentiable clustering.
2. We present a spatial continuity loss function that overcomes the restrictions of earlier work's fixed segment bounds.
3. We offer an adaptation of the suggested method for segmentation using scribbles as user input, which demonstrated higher accuracy and efficiency than previous methods.

Finally, we present an extension of the suggested method: unseen image segmentation utilising networks that have been pre-trained with a few reference images and do not require retraining. On multiple picture segmentation benchmark datasets, the effectiveness of the suggested method was tested.

**METHOD**

**Network architecture**:

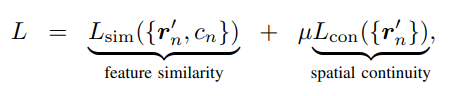


On the whole, the architecture can be shown in the above figure where, the input image I is supplied into the CNN, which uses a feature extraction module to extract deep features {xn}. The response vectors {rn} of the features in q-dimensional cluster space are then calculated by a one-dimensional (1D) convolutional layer, where q = 3 in this example. The three axes of the cluster space are represented by z1, z2, and z3. The response vectors are then standardized across the cluster space's axes using a batch normalization method. Furthermore, cluster labels {cn} are determined by utilizing an argmax function to assign cluster IDs to response vectors. The feature similarity loss is then computed using the cluster labels as pseudo targets. Finally, the feature similarity loss and the spatial continuity loss are computed and backpropagated.

This architecture solves two problems which are the Constraint of feature similarity and the Constraint of the number of unique cluster labels. The first constraint is solved by normalization of the obtained results into q distinct labels or clusters using the argmax classification which is equivalent to assigning each pixel to the closest point among the q representative points, which are placed at an infinite distance on the respective axis in the q-dimensional space. The Second constraint is solved by using the strategy of classifying the pixels into an arbitrary number q` (1 ≤ q` ≤ q) of clusters, where q is the maximum possible value of q`. A large q` indicates over-segmentation, whereas a small q` indicates under-segmentation. To train a neural network, we set a large number to the initial (maximum) number of cluster labels q. Then, in the iterative update process, similar or spatially close pixels are integrated by considering feature similarity and spatial continuity constraints. This phenomenon leads to reducing the number of unique cluster labels q`, even though there is no explicit constraint on q.

**Loss function:**

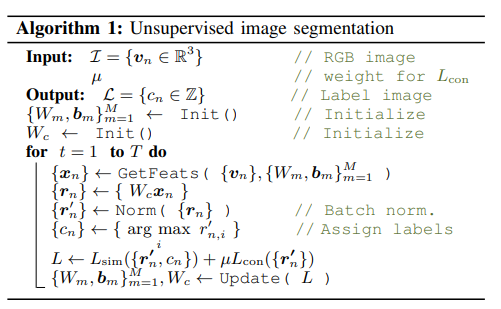
The loss function L used consists of a constraint on feature similarity and a constraint on spatial continuity, denoted as follows:



where µ represents the weight for balancing the two constraints.

The loss function helps us to solve two constraints which are the constraint on feature similarity and the constraint on spatial continuity. The first constraint is solved through the minimization of this loss function by network weight updation which facilitates the extraction of more efficient features for clustering. The second constraint is solved through the spatial continuity loss function which helps in suppressing the complicated patterns or textures due to an excessive number of labels.

**Final Algorithm Using Backpropagation:**

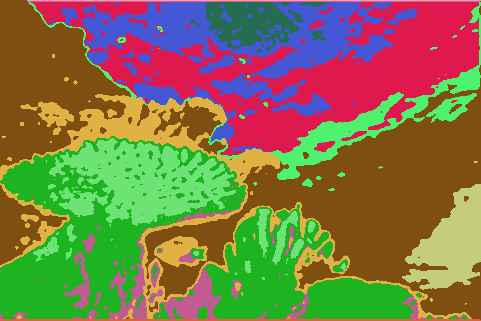
****

On the whole, the algorithm based on the architecture can be shown in the above figure where the following two sub-problems are solved: the prediction of cluster labels with fixed network parameters and the training of network parameters with the (fixed) predicted cluster labels. The former corresponds to the forward process of a network followed by the proposed architecture described in Architecture. The latter corresponds to the backward process of a network based on gradient descent. Subsequently, we calculate and backpropagate the loss L described in the Loss function to update the parameters of the convolutional filters {Wm} m=1 to M as well as the parameters of the classifier Wc. This forward-backward process is iterated T times to obtain the final prediction of the cluster labels {cn}.

**RESULTS**

**Implementation of the paper:**

The following input image is passed into the CNN described above, which outputs the below image. CNN started with 1000 clusters and ended up with 11 clusters with a loss of 0.039 for 1000 iterations over the network.

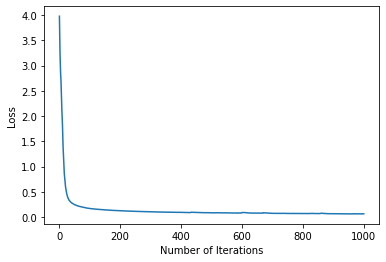
 

loss: 0.03915729373693466

number of labels: 11

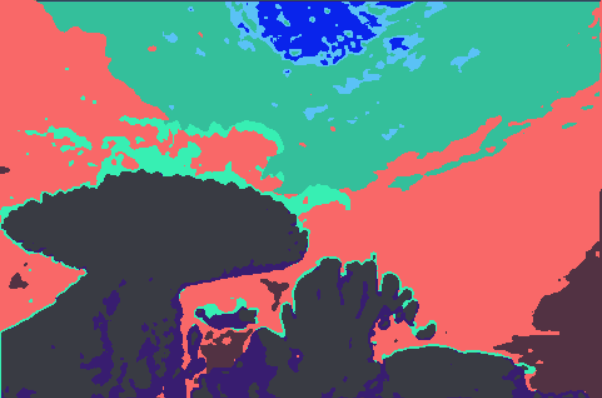
iterations: 100

**Number of iterations vs Loss**

****

From the above graph of loss vs the number of iterations, we can see that there is a drastic decrease below hundred iterations and from 100 to 400 the decrease in loss is very less. And from 400, it is negligible. So we chose 400 iterations for further simulations.

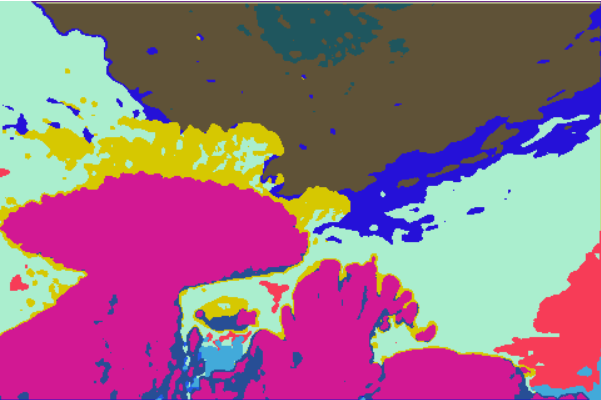
**Effect of the parameter minLabels:**



loss: 0.07659568637609482

number of labels: 11

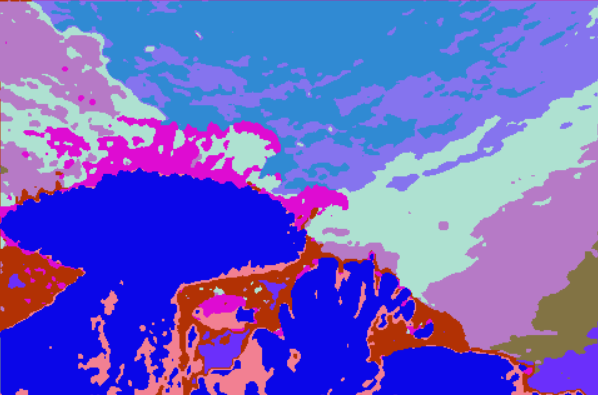
minLabels=3



number of labels: 13

loss: 0.07456070184707642

minLabels=4



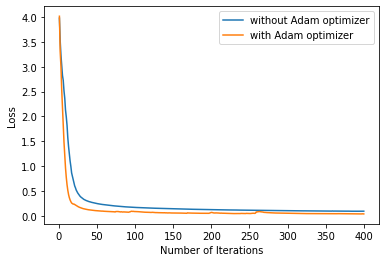
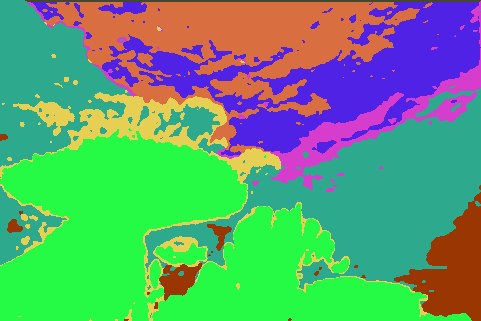
number of labels: 13

loss: 0.0838036984205246

minLabels=5

Here we took 400 iterations. By changing the value of minimum labels, we can see that the loss value remains almost the same. However, the number of labels depends on the minimum labels. Though using a minimum of 4 clusters may look better compared to 3 from above visuals, and the difference in loss is less and the time taken to train is lot more. So we decided to go with a minimum of 3 clusters.

**Using Adam Optimizer**



loss for 400 iterations: 0.03915729373693466

We used adam as an optimizer with a learning rate of 0.1 and beta values 0.9 and 0.99. For 400 iterations, the above plot shows the loss with architecture in paper and loss with the Adam optimizer.

The loss using 1000 iterations by implementing the paper is almost near to the loss we get when the Adam optimizer is used for 400 iterations.

We can also see that the boundaries between the clusters are much more well-defined compared to before.

**CONCLUSION**

The CNN architecture described above is able to extract features from a given input image using convolution filters and cluster those features with the help of differentiable processes like an end-to-end network training model. Here everything is learned by the CNN (as one big task), and there is no decapsulated extra-step like Feature-extraction.

The above CNN assigned image pixels a cluster label and simultaneously updated the convolution filter for better separation of clusters using the backpropagation of the loss to normalized responses of convolution layers.

**CONTRIBUTIONS**

| **Name** | **Contributed work** |
| --- | --- |
| ***GOLI NAGA SANDESH*** | Implemeneted paper |
| ***SRI MIHIR DEVAPI UNGARALA*** | Further improvements |
| ***GORIPARTI SAIVINAY*** | Implemented paper |
| ***CHALLA SAI RESHWANTH*** | Further improvements |
| ***A RAHUL*** | Further improvements |
| ***SHAIK MOHAMMED SHAQUIB*** | Implemented paper |
|  |  |