COMP 508 FINAL PROJECT REPORT

CLUSTERING APPEARANCE AND SHAPE BY LEARNING JIGSAWS

1. Introduction

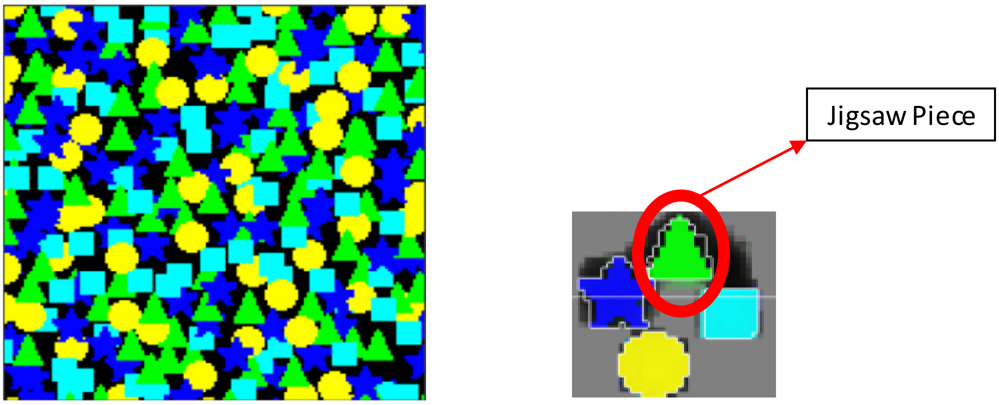
The aim of this project is to implement the Jigsaw Model appeared in the paper, “Clustering Appearance and Shape by Learning Jigsaws”, by A. Kannan, J. Winn and C. Rother. The paper was published at “Advances in Neural Information Processing Systems (NIPS)” in 2006.

Jigsaw model is a patch based appearance model which is widely used in computer vision applications. Appearance models range from histogram based models that disregard spatial information, to template-based models that try to capture the entire spatial layout but very sensitive to variation [1]. However, jigsaw model lies in the middle of these two extremes and balance their benefits.

The main problem in patch based models is that these models require predefined set of fixed patch sizes and shapes (often rectangles and circles) by hand. However, in the jigsaw model the shape, size and appearance of patches are learned automatically from the repeated structures in an image without supervision.

The Jigsaw Model of Kannan et al [1] proposes that there is no need to predefine patch shapes and sizes since natural images provide enough information to discover the shape and size of objects. Moreover, they show that the automatically discovered patches in the jigsaw model are strongly related to semantic object parts in the image. For instance, when we apply the jigsaw model to face images, the automatically learned jigsaw pieces are strongly related to face parts such as eyes, noses, eyebrows and cheeks.

The jigsaw model is a patch-based probabilistic generative model that learns appropriate sized and shapes of the patches from a latent image called as jigsaw. In [1] they call this latent image as jigsaw because it contains all the necessary jigsaw pieces that can be used to generate the source images. An example of training image, jigsaw and Jigsaw pieces shown in Figure 1.



(a) Trainin Image

(b) Jigsaw

Figure 1. a. Trainin Image, b. Jigsaw Image and Jigsaw Piece

Jigsaw model can be used as a tool in many computer vision applications for both image understanding and image synthesis such as object recognition, detection, image segmentation and image classification, object synthesis, image de-noising, super resolution, texture transfer between images and image in-painting [1].

While implementing the project, I applied the computer vision concepts covered during semester suchas color systems, object recognition, pattern recognition, connected component labelling, Bayesian decision making and probabilistic modelling, image segmentation and graph partitioning.

2. Related Work

Jigsaw Model is an extended version of Epitome Model of Jojic et al.[2]. Epitome model is a probabilistic patch based generative model as well. The main difference between these two models is how to choose patch size. In epitome model user should set a fix size and shape of patch in order to learn epitome which is compacter version of the training image containing all the necessary information to reconstruct the source image. In contrast, in the jigsaw model the probabilistic inference process chooses the proper shaped and sized pieces from the training image while learning the jigsaw image.

Another work related with jigsaw model is Freeman et al. [3]. They propose a machine learning model in order to infer underlying scenes from images and define a Markov Random Field over image patches as jigsaw model defines. However this model require fixed size of image patches.

The texture synthesis algorithm of Efros et al. [4] construct a new image such as jigsaw and epitome models by selecting different patch shapes. However, this work does not learn patch appearance.

The disadvantage of working with fixed size and shape of patches appears when the object part is of different size and shape than the fixed patch. This negative impact increase when the object part is on the edges. Additionally, fixed patch size models disregard the shape of the object part which is much discriminative feature than the appearance alone. [1]

3. Jigsaw Model

Jigsaw model aim to learn a condensed image, jigsaw. In the paper [1], they define jigsaw pieces (patches) as a set of spatially grouped pixels in jigsaw image. The jigsaw model tries to find a jigsaw such that pieces of the jigsaw satisfy the following criteria [1]:

i. **Shared**: Each piece is similar in shape and appearance to many regions of the training images.
ii. **Discriminative**: Each piece is as large as possible;
iii. **Exhaustive**: All parts of the training images can be reconstructed from the set of jigsaw pieces.

In contrary to other patch based models, jigsaw model allows the jigsaw pieces to have arbitrary shapes in order to satisfy the requirements explained above. The consequence of meeting jigsaw requirements, the model able to capture both the appearance and the shape of repeated image structures, for example, eyes, noses and mouths in a set of face images.

a. What is Jigsaw?

In [1] they define a jigsaw J to be an image such that each pixel z in J has an intensity value $\mu(z)$ and an associated variance $\lambda^{-1}(z)$ (λ is called as precision). Jigsaw pieces which are grouped together in the jigsaw image are used to generate (reconstruct) images.

b. Probabilistic Model

The probabilistic model that jigsaw based on is a generative image model which generates an image by joining together pieces of the jigsaw and then adding Gaussian noise of variance given by the jigsaw. The variables used in the jigsaw model is explained below:

$I = Trainin\,Image$

$L = Offset\ Map$

$J = Jigsaw\ Image$

$i = image\,pixels$

$z = Each\,pixel\ in\ jigsaw\ image\ (J)$

$l = Offset\ value\ in\ the\ offset\ map$

$|J| = (width, weight)\ Dimension\ of\ the\ jigsaw\ (J) image$

The jigsaw model defines an offset map (L) associated with each training image (I) which determines the jigsaw pieces used to reconstruct the image. Size of the offset map is same as the image size and the offset map keep a position for each pixel in the training image (I). Additionally, more than one image (I) pixel can map to the same jigsaw pixel. Offset map defines a position in the jigsaw image by a two dimensional offset $l_i = (l_x, l_y)$, which maps a 2D point i in the image to a 2D point z in the jigsaw using the formula:

$$z = (i - \ l_i\ ) mod\ |J|$$

The subtle and tricky property of this formula is that if two adjacent pixels in the image have the same offset label, then they map to adjacent pixels in the jigsaw. This crucial property keeps the smoothness of the jigsaw image which enforce the model catch repeated structures in the jigsaw image.

Given the mapping and the jigsaw, the probability distribution of an image:

$$P(\,I\,|\,J, L\,) = \prod_i N\,(I\ (i);\ \mu\,(i - l_i\,), \lambda\,(i - \ l_i)^{-1}$$

In order to reconstruct images consisting of coherent pieces of the jigsaw, in [1] they define Markov Random Field (MRF) on the offset map to encourage neighboring pixels to have same the same offsets (smoothness).

$$P\,(\,L\,) = \ \frac{1}{Z}\exp[\,-\,\textstyle\sum_{(i,j)} \psi\,(\,l_i, l_j\,)\,]$$

$E\ = The\ set\ of\ \ edges\ in\ 4\ connected\ grid$

$\psi = Interaction\,potential\ defines\ a\ Pott's\,model\,on\ the\ offsets$

The Pott's model defined on the MRF:

$$\psi\,(l_i, l_j\,) = \ \gamma\,\delta\,(\,l_i\ \neq\ l_j\,)$$

$\gamma = The\ parameter\ which\,influences\ the\ typical\ size\ of\ \ the\ learning\ pieces$

c. Markov Random Field and Pott's Model

Markov Random Field (MRF) is a graphical model of a joint probability distribution. It consists of an undirected graph in which the nodes represent random variables (pixels) as shown in Figure 2. Given disjoint subsets of nodes A, B, and C, A is conditionally independent of B given C if there is no path from any node in A to any node in B that doesn't pass through a node of C as shown in Figure 2.

Markov property can be defined as: given its neighbor set, a node n is independent of all other nodes in the graph. Markov property can be formulated [5]:

$$P(X_n|X_N - X_n) = P(X_n \mid X_{Nn})$$

$X_n = Subset\ of\ node\ n$

$X_N = All\ nodes$

$X_{Nn} = Neighbor\ Nodes\ of\ Subset\ X_n$

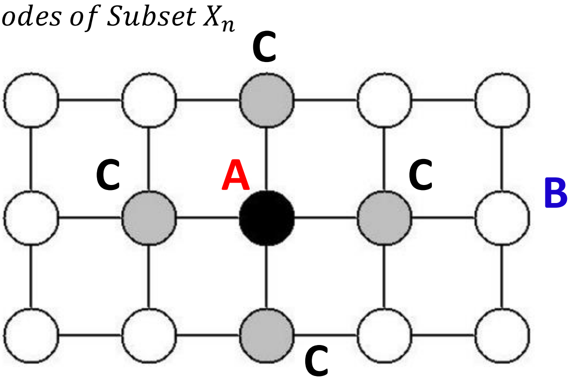


Figure 2. Markov Random Field

Figure 2. Markov Random Field: Given the grey nodes, the black node is conditionally independent of all other nodes.

In jigsaw model MRF defined on the offset map in order to encourage neighboring pixels to have the same offset. So as to meet this requirements Pott's model defined as interaction potential on MRF. Pott's model decrease optimality in proportion to local interaction strength γ when the neighboring pixels assigned to different offset labels.

$$\psi\,(l_i,l_j) = \gamma\,\delta\,(\,l_i \neq l_j\,)$$

$l_i \neq l_j \Rightarrow 1\ if$ neighboring pixels assigned to different offset labels.

$l_i = l_j \Rightarrow 0\ if$ neighboring pixels assigned to same offset labels.

Local interaction strength is the most important parameter in the model because it affects the size of jigsaw pieces in other words the smoothness of the reconstructed image.

While learning the jigsaw, some regions in the jigsaw image may be unused. In order to satisfy this case, in [1] they define a Normal-Gamma prior on μ and λ for each jigsaw pixel z in jigsaw as shown below.

$$P\,(J\,) = \prod_z N\,(\,\mu\,(z);\ \mu_0, (\beta\,\lambda\,(z\,)^{-1})\ Gamma\,(\,\lambda\,(z\,); a, b)$$

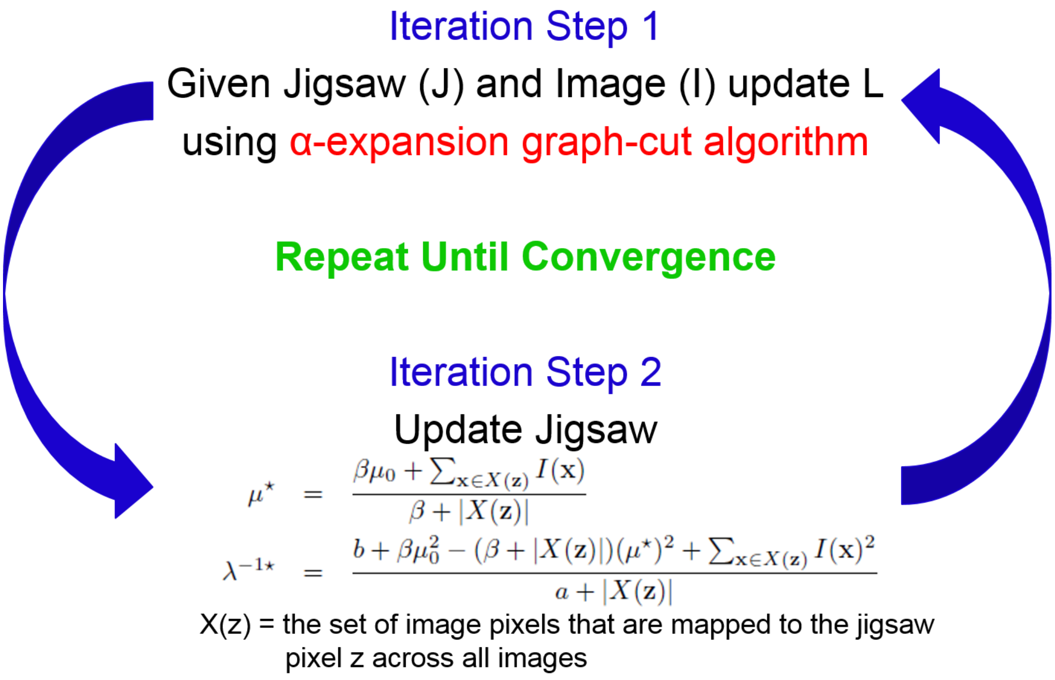
In [1] they fix the hyperparameters $\mu = 0.5, \beta\ = 1$ and b to three times the inverse data variance and a to the square of b. The local interaction strength is se to 5 per RGB channel.

d. Inference and Learning (Expectation and Maximization (EM) Algorithm)

The model defines joint probability distribution on the training image (I), jigsaw (J) and offset map L and try to maximize probability of:

$$P\ (J, \{I, L\}_1^N\ ) = \prod_{n=1}^{N} P(\ I_n \,|\, J, L_n\ ) P(L)$$

We know the training image (I) and the aim of model is to find jigsaw (J) and offset map (L) so as to maximize the joint probability. The model maximize the probability by applying Expectation Maximization algorithm. At the beginning of EM algorithm, the jigsaw image is set to the Gaussian noise with same mean and variance as the training image. Then find the offset map by applying alpha-expansion graph cut algorithm [6]. Upon finding the offset label (L) the model update jigsaw mean and variance. These iteration steps continue until alpha-expansion graph cut algorithm converges. The summary of EM algorithm is shown in Figure 3:



Iteration Step 1

Given Jigsaw (J) and Image (I) update L using α-expansion graph-cut algorithm

Repeat Until Convergence

Iteration Step 2

Update Jigsaw

$$\mu^{\star} = \frac{\beta\mu_0 + \sum_{\mathrm{x} \in X(\mathbf{z})} I(\mathrm{x})}{\beta + |X(\mathbf{z})|}$$

$$\lambda^{-1\star} = \frac{b + \beta\mu_0^2 - (\beta + |X(\mathbf{z})|)(\mu^{\star})^2 + \sum_{\mathrm{x} \in X(\mathbf{z})} I(\mathrm{x})^2}{a + |X(\mathbf{z})|}$$

X(z) = the set of image pixels that are mapped to the jigsaw pixel z across all images

Figure 3. EM Algorithm

Alpha expansion graph cut algorithm is a type of minimum cut and maximum flow algorithm. The graph cut algorithm assigns every pixel a label on the offset map (L) and find a labelling set which is smooth and consistent with the observed data. Moreover, the graph cut algorithm convert min cut max flow problem into Energy minimization problem which objects to minimize energy. Energy of labelling can be defined as:

$$E(f) = E\_smooth\ (f) + E\_data\ (f)$$

E_data (Data Cost) = (I – Assigned Jigsaw Pixel) ^ 2

E_smooth (Smooth Cost) = MRF & Potts Model

When the EM algorithm converged, we get a mapping of each pixel in training image to a jigsaw pixel as shown in Figure 4:

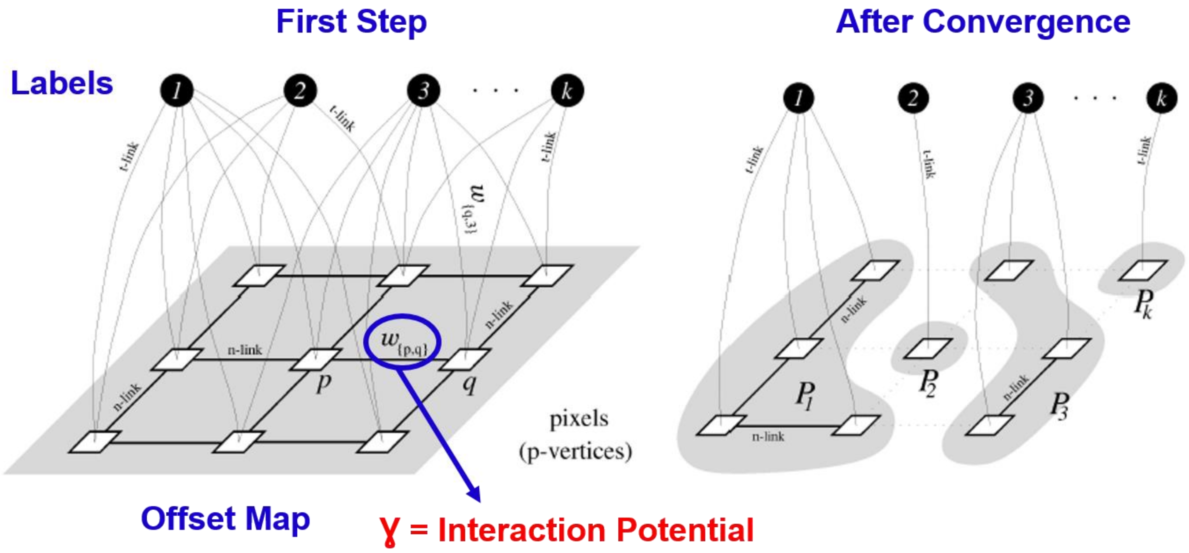
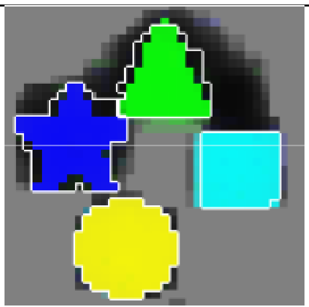
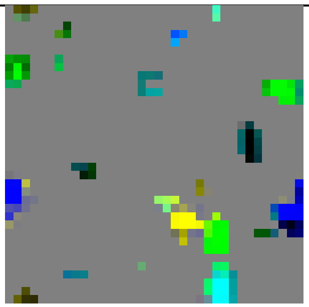
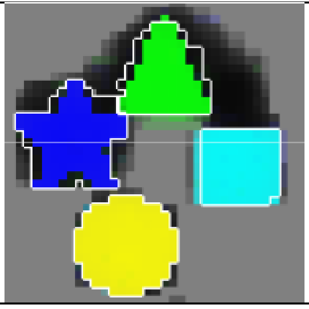
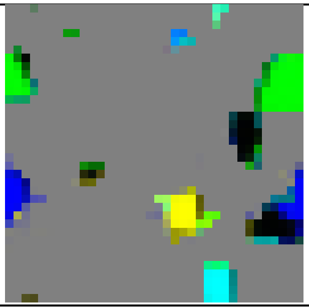


Figure 4. Convergence of Graph Cut Algorithm

4. Implementation, Results and Evaluation

I implemented the jigsaw model in Matlab. I used open source alpha-expansion graph cut algorithm Matlab wrapper [7] for my implementation. The graph cut matlab wrapper can be found in [7] which is supplied by authors of [6]. I applied the rest of the model myself in Matlab.

During the implementation the most difficult part is to figure out how the parameters affect the smoothness of jigsaw pieces. The crucial parameter which influence the sizes of jigsaw pieces is the interaction potential defined in Pott's model. In paper, they state that they fixed the interaction potential to 5. To figure out how this parameter affects the size of jigsaw pieces I made many tests and my results are shown in Figure 5:

Interaction Potential	Converged Iteration Nu	Original Jigsaw (36X36)	My Jigsaw (36X36)
2000	26		
3825	25		

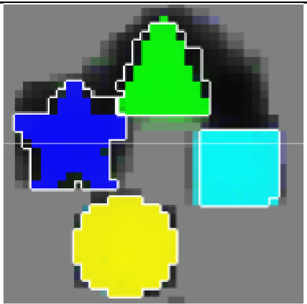
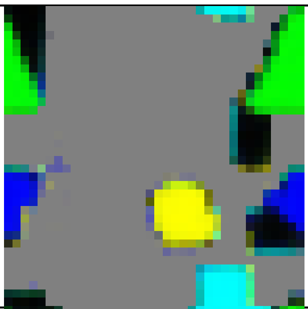
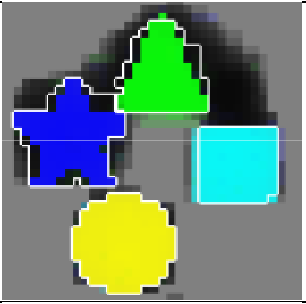
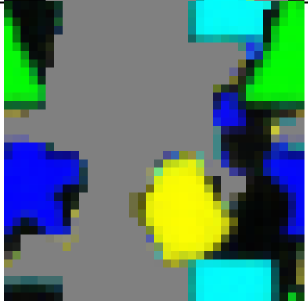
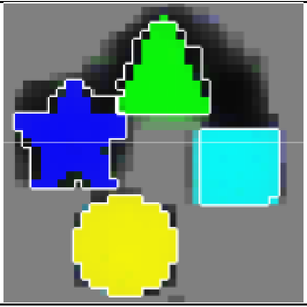
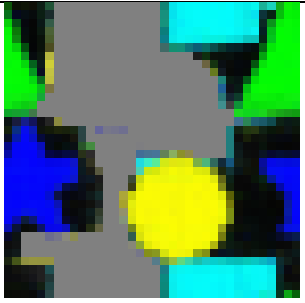
Interaction Potential	Converged Iteration Nu	Original Jigsaw (36X36)	My Jigsaw (36X36)
10000	16		
20000	20		
30000	15		

Figure 5. Importance of Interaction Potential (Smooth Cost Weight)

In the paper they tested the jigsaw model on 3 different test data. The first test data is so called toy example which is a hand-crafted 150x150 RGB image. I compared my results with the results in the paper regarding to toy example. The comparison is shown in Figure 6:

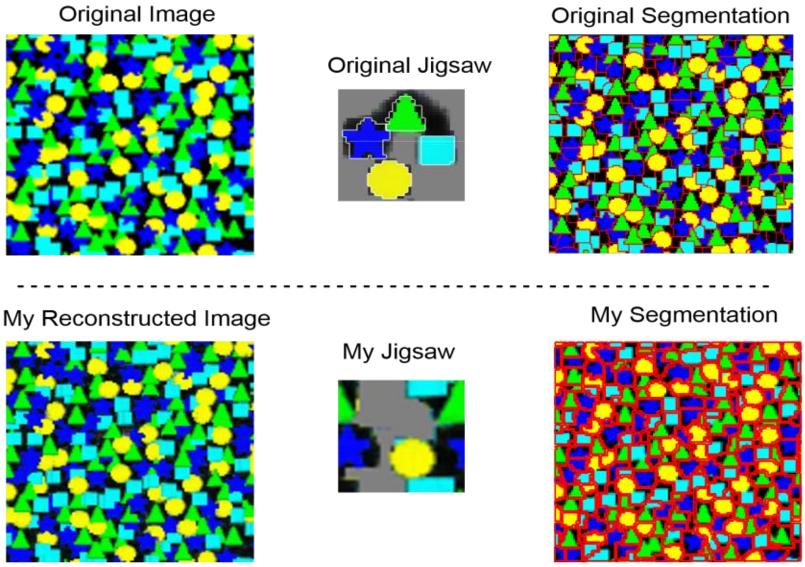
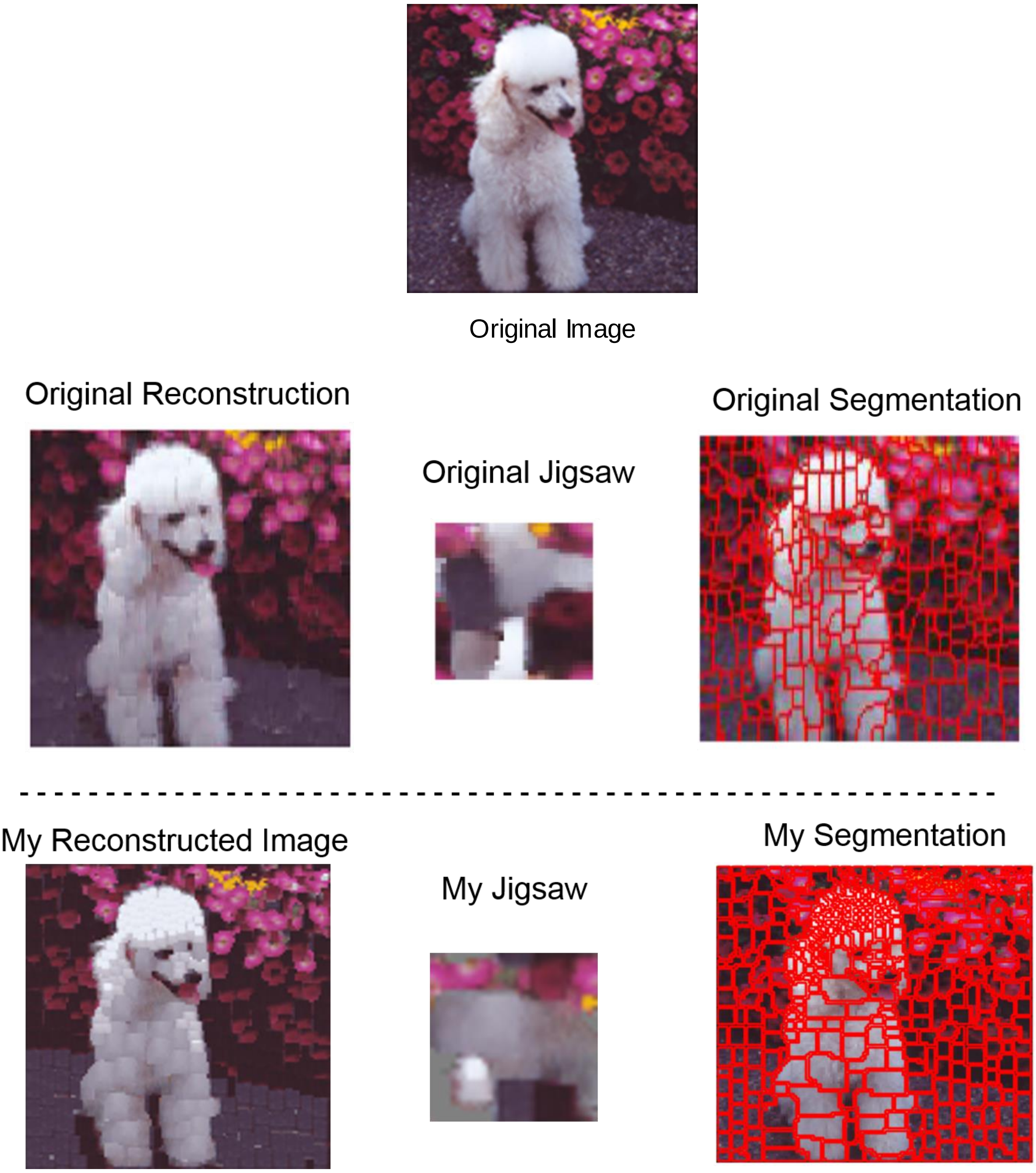


Figure 6. Comparison of Toy Example

Jigsaw pieces extended over borders in my jigsaw however it is normal. Since we use mod operator for offset labels, they are complementary and keep the integrity. Segmentation is finding the connected components in offset map. The objects in the red area exhibits the group of pixels which have the same offset label. As a result my model gives satisfactory result with toy example.

Another test data is a dog image which is used to compare Epitome model [2] with Jigsaw Model [1]. Actually the jigsaw model is an extension of Epitome model [1] and the comparison of these two models given in the paper. My test result with dog image and its comparison with the results in the paper shown in Figure 6:



Original Image

Original Reconstruction

Original Jigsaw

Original Segmentation

My Reconstructed Image

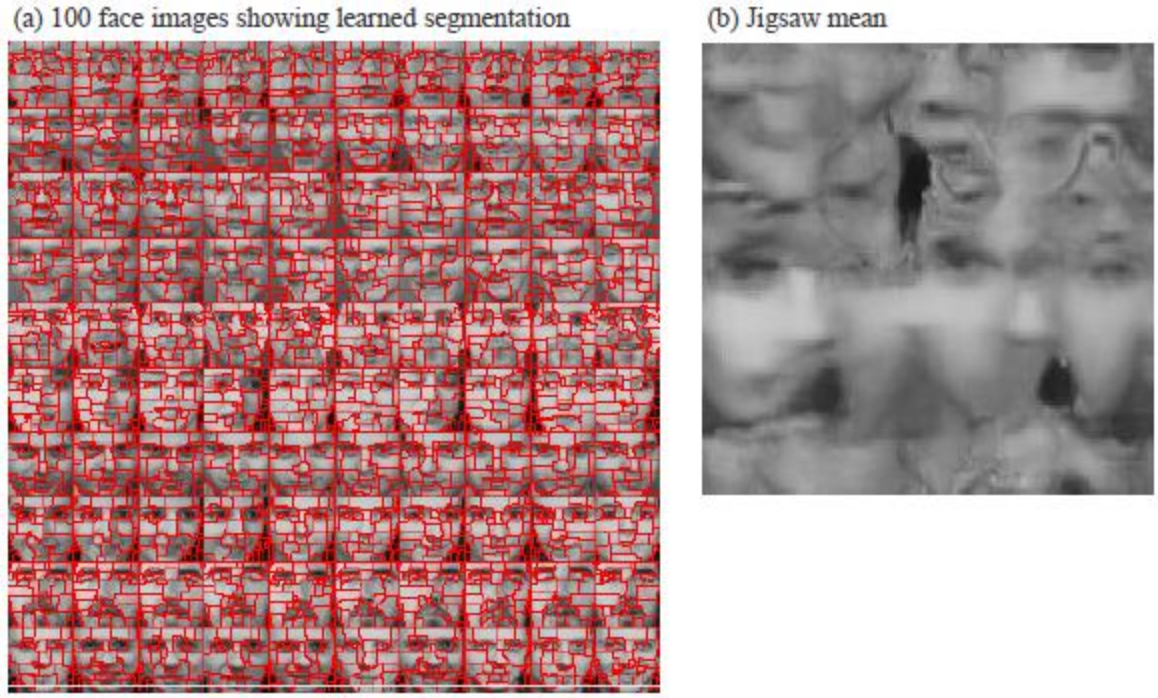
My Jigsaw

My Segmentation

Figure 7. Comparison of Dog Example

Since there is no information about the interaction potential value that they used to create jigsaw for dog image in the paper, the differences between my reconstructed image, jigsaw and segmentation is acceptable.

Finally the jigsaw model is tested with face images from Olivetti Database [8]. For test purposes they choose 10 different face images for each 10 person. Thus they tested the model with 100 face images. Size of each face is 64x64 and total size of test image is 640x640. The test image including segmentation and related jigsaw is shown in Figure 8:


(a) 100 face images showing learned segmentation

(b) Jigsaw mean

Figure 8. Test of Face Images and Jigsaw

The learned jigsaw contains a number of face elements such as eyes, noses etc in Figure 8. In the segmentation image, the jigsaw pieces strongly associated with a particular face part.

To test my jigsaw model I could not use the test image that is used in the paper because testing this image consume up all the memory on my desktop. I tried to test the image on Koc University Lufer Cluster however it consumed up all the memory on the cluster as well. Thus I simplified the problem and choose only 4 different images belongs to 4 different people and tested my model. My test image and results are shown in Figure 9:

a. Test Image

b. Jigsaw

c. Segmentation

Figure 9. Test Results of Face Images

Since I used only 4 image there is not enough repeated structure to learn semantic face parts in my jigsaw such as noise, mouth etc. However, in my segmented image the jigsaw pieces are strongly associated with some particular face parts such as eyes, cheek, mouth, noise etc. As a consequence, even though I used limited test data, my model gives satisfactory results.

In the paper they did not include the clustering step into the model during learning process and they extended this job as a future work [1]. Thus I did not implemented clustering step. However the idea behind the clustering step is very easy. If segmented part regions overlap in the jigsaw and the degree of overlap is above a threshold we can cluster these segmented parts into the same cluster.

5. Conclusion

As a conclusion, my jigsaw model implementation gives satisfactory results compared to the results shown in the paper [1]. Even though how the model should be implemented is not clear in the paper and they do not supply sufficient information about the roles of parameters, I applied the model successfully and get satisfactory results.

In this project I implemented the jigsaw model which is capable of learning the shape, size and appearance of repeated regions in a set of images. Additionally, I showed that for a set of images, the learned jigsaw pieces are strongly associated with particular face parts.

A practical issue for with learning jigsaws is the computational requirement. Every iteration of learning involves solving as many binary graph cuts as there are pixels in the jigsaw. For instance, for the toy example, it took about 30 minutes to learn a 36 x 36 jigsaw from a 150 x 150 image.

Additionally, transformations such as rotation, scalings and flip can be incorporated in the model with cost increasing linearly with the number of transformations [1].

References

[1] A. Kannan, J. Winn and C. Rother. Clustering Appearance and Shape by Learning Jigsaws. NIPS 2006.

[2] N. Jojic, B. Frey, and A. Kannan. Epitomic analysis of appearance and shape. In ICCV, 2003.

[3] W. Freeman, E. Pasztor, and O. Carmichael. Learning low-level vision. IJCV, 40(1), 2000.

[4] A. Efros and W. Freeman. Image quilting for texture synthesis and transfer. In ACM Transactions on Graphics (Siggraph), 2001.

[5] http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV0809/ORCHARD/

[6] Y Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. PAMI, 23(11).

[7] http://vision.csd.uwo.ca/code/

[8] http://cs.nyu.edu/~roweis/data.html