Segmenting Images with a Deep Auto-encoder and K-Means Clustering

Adia Meyers
Clayton State University
Email: ameyers3@student.clayton.edu

***Abstract*—The purpose of this research is to improve some of the current methods of image segmentation, thereby allowing for more accurate results when categorizing images. By specifically venturing into image segmentation and neural networks, we hope to find a collaborative and beneficial correlation between convolutional neural networks and categorizing images. This research combines various segmentation methods and evaluation methods in hopes of creating a robust algorithm. By performing a particular technique of image segmentation, the desired product is to be capable of classifying local, global, and multi class images. In turn, this will constitute for an improved and more accurate way of segmenting images.**

Keywords*—*Image segmentation, convolutional neural network

I. Introduction

Image segmentation is the process of isolating an image into a number of sections which are known as segments [5,6,7,8] so that the image can be classified. Image segmentation plays a large role in: classifying terrains in satellite images, medical image analysis, character recognition, and more. There are a variety of image segmentation techniques, such as thresholding, clustering, edge detection, implementing a watershed transform, using artificial intelligence for segmentation, and many more methods. Thresholding segments images by looking at the intensity values of pixels [8]. Clustering groups images with similar characteristics [4,14], and edge detection detects discontinues between objects in an image [12], therefore finding the boundaries of each object in an image. Watershed algorithms transform a grayscale image by acting as a topographic map, with the brightness of each point representing its height [2]. Artificial intelligence based classification has recently begun to play a larger role in image segmentation in the form of neural networks, in particular convolutional neural networks [5,6,10].

A neural network is an artificial intelligence system modeled after our very own human brains and is made of layers of nodes. Every nodes incoming connection has a weight associated with it. This weight is multiplied by the input. Neural networks consist of at least three layers. A input layer, hidden layer and an output layer. Neural networks learn by first evaluating training images. Then test images, usually of the same object type as the training images are evaluated. Convolutional neural networks are very similar to ordinary neural networks, meaning that they are made up of neurons that have learn-able weights and biases [1,10,16]. The main differences between the two being that convolutional neural networks do not make use of every feature in an input image and they are capable of conductiong dimensionality reduction. Convolutional neural network architectures make the explicit assumption that the inputs are images, which allows us to encode certain properties into the architecture [12]. The Convolutional neural network processes the input and output data. Any data that is given after the dataset is classified appropriately. What makes neural networks so unique, is that they do not require pre-made categories in order to classify images.

II. Related Work

Segmenting a multi-class image as well as a large scale image can be some of the most difficult aspects of image classification. In 1987, auto-encoders were first proposed as a means of aiding in image segmentation [1]. An auto-encoder is a unsupervised convolutional neural network that applies back-propagation, meaning that the output is the same as the input [10]. An auto-encoder is made of two parts, an encoder and a decoder. The encoder portion breaks down the input into a vector, and the decoder builds the vector back up into the original input image. In a paper entitled Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections, Mao et la. use a deep auto-encoder for image restoration. Auto-encoders pose as useful attributes for image segmentation due to their ability to perform dimensionality reduction and extract important features of an image [3,9,15].

Although auto-encoders aide in image segmentation, they do not actually classify an image. For this research project, a clustering algorithm will be implemented in order to segment the images. K-means clustering is an unsupervised clustering algorithm. The k means algorithm takes in the input as well as a required parameter (k) which will determine the number of clusters. The desired result is for the points in a similar cluster to have a minimized distance and for the distance between clusters to be maximized. Duan et al. implement a simple k means algorithm for the purpose of classifying fish images. It divides data into a predetermined classes on the basis of minimizing the error function.

III. Problem Definition

When implementing certain image segmentation methods such as using a watershed transform or thresholding algorithm, there always the constant needed requirement of pre-made categories. Using a convolutional neural network eliminates this prerequisite. Despite the fact that there are implementations of convolutional neural networks and image segmentation, there is still much room for improvement. For example, Meyer et la. establish a convolutional radial basis function solver paired with k means clustering to perform image segmentation [11]. Their experiment yielded accurate results, however, they still had an accuracy loss of about twenty percent on average. By replacing the radial basis function solver with a different convolutional neural network, we may be able to generate even more accurate results. This will also grant us the opportunity to create another method of image segmentation that could potentially outperform other segmentation methods as well.

IV. PROPOSED RESOLUTION

Our proposed resolution is to implement a model with a high accuracy for image segmentation. We will combine the methods of a conolutional neural network and a k-means clustering algorithm. The type of convolutional neural network to be implemented will be an auto-encoder. Refer to figure 1 for a representation of our network.

A. The Auto-encoder: The auto-encoder we developed mainly served as a means of extracting important features and reducing dimensions of images. Our auto-encoder is made of convolutions and deconvolutions. The encoder portion of the auto-encoder takes in a patch from the image as its input and break the input down into a vector known as the embedding of the auto-encoder. Then the rELU activation function is applied to the embedding and the decoder builds the vector back up to the original image. Lastly the L2 loss function will be calculated for the sole purpose of determining how well the output image matches the input image. This will measure the robustness of the auto-encoder.

B. T-SNE The vector from the embedding of the auto-encoder is extracted and the t-Distributed Stochastic Neighbor Embedding is applied to that vector in order to transform the vector points into two dimensional points. Ergo making it easier to plot the points for our k means resulting cluster graph.

C. The K-means clustering algorithm: In the second process, the two-dimensional vector points from the T-SNE replace the color values in the cluster vector. This will allow us to use distance instead of a color based vector as our cluster input vector. The K means algorithm has one parameter, which is the desired number of clusters (k).

D. Lastly, we plan to use the recall @ K method to determine the accuracy of our network. Recall @ K relies on the number of tests made [11]. R represents the total tests done and N represents the amount of test which were correct. N is divided by R and that determines the accuracy. The larger N is, constitutes for a better accuracy representation.

V. EXPERIMENT RESULTS

The first step in our experiment was to train the auto-encoder. There were 2 data sets which were run on the auto-encoder, mnist and cifar. There are 60,000 images in each of these data sets. 50,000 were used for training and 10,000 were used for testing. The cifar data set consists of random colored objects such as animals and cars. The mnist data set consisted of black and white images of handwritten numbers. The cifar images were 32x32 pixels and the mnist data set were images of 28x28 pixels. Each data set was trained for 6,000 epochs and a batch size of 1 due to their small sizes. The output results for the mnist data set is shown in figure 2 and the results for the cifar data set is shown in figure 3.

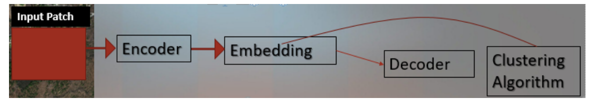

Fig. 1. A representation of our future image segmentation model.

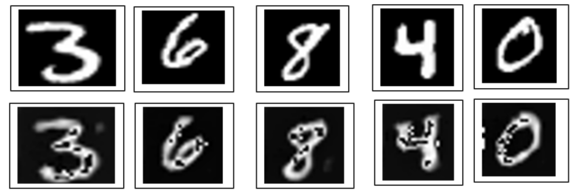

Fig. 2. Images from mnist datanset after being processed through our auto-encoder.

During the development of the k means algorithm, all of the figures 4 - 9 were developed from the mnist data set points. Figure 4 shows our graph results after applying the t-SNE to our data points without performing any k means on our data. As of right now, we are attempting to fix our t-SNE code, due to the fact that there should be much more data points in figure 4. Ergo indicating that our code needs improvement. Before resulting to the recall @K technique, we tried another evaluating technique that looked at the predicted labels vs actual labels on a graph. However we unable to get that functioning and decided to move on to a different method. Figure 6 was the first real plot that showed any sort of clustering, we initialized our k clusters to three for this trial. Despite the small milestone figure 6 signified, it still did not display all the points as we wanted. Figure 7 was an unsuccessful plot, which did not plot the clusters individually. Figures 8 and 9 are from our most recent results and we initialized our number of cluster classes to ten for both of these experiments. Figure 8 is graph of our most recent k means code without applying t-SNE to our data points before clustering them. Figure 9 is a graph of our data points after having the t-SNE applied to them and running our k means code. Despite being unable to get the recall @K to work, we were able to see that the points in figure 8 were closer together and therefore more clustered.

Fig. 3. Images from mnist dataset after being processed through our auto-encoder.

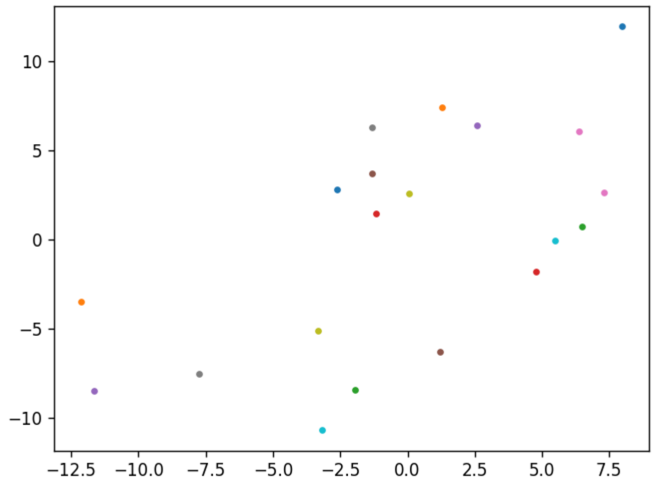

Fig. 4. Plot of points after applying t-SNE.

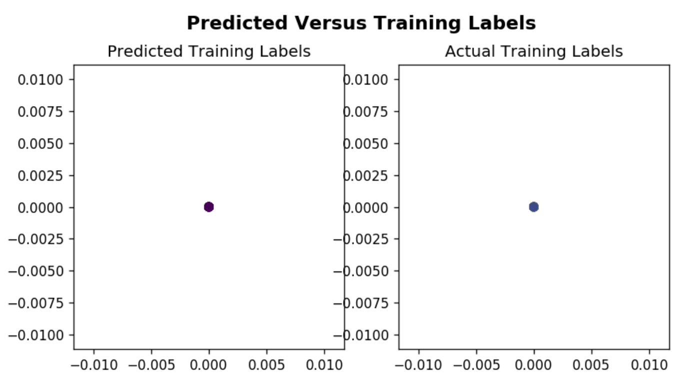

Fig. 5. A plot that was supposed to show our predicted vs actual training labels.

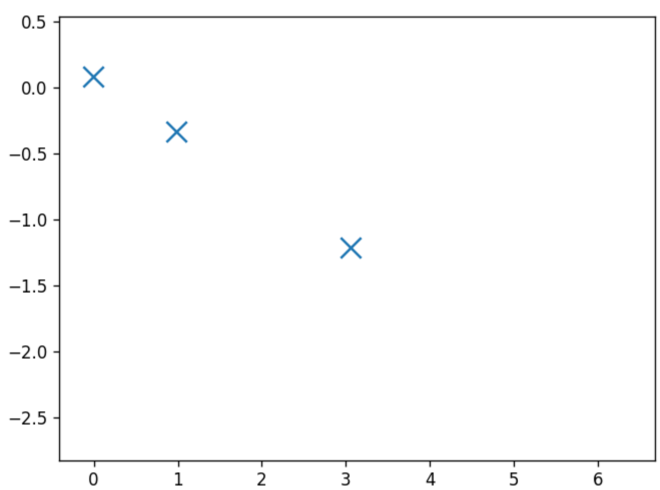

Fig. 6. Plot of the one of our trial k means algorithms using mnist data set.

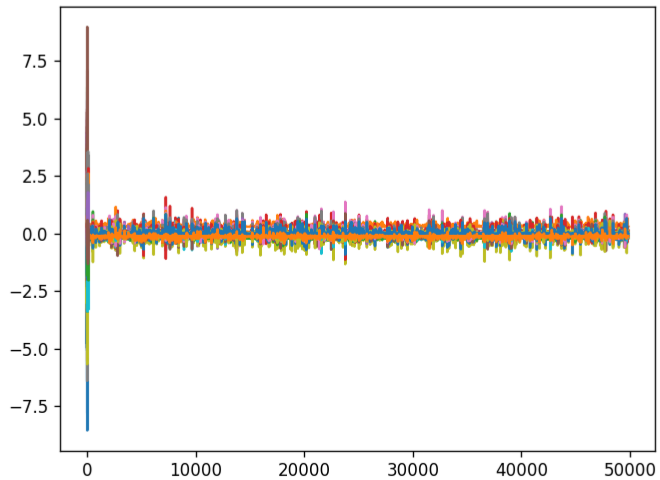

Fig. 7. Plot of the one of our trial k means algorithms using mnist data set.

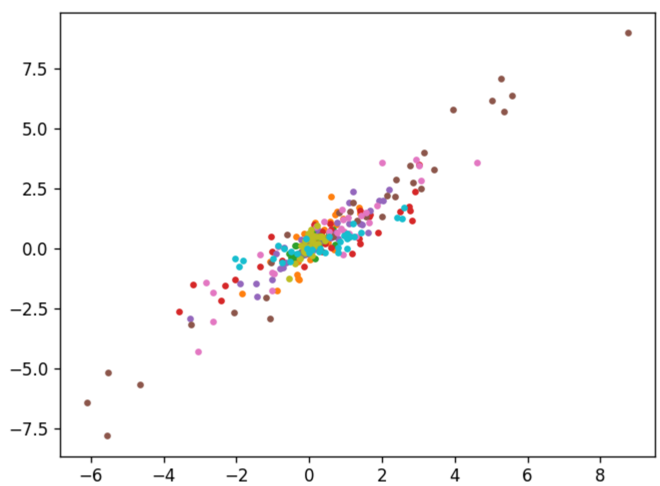

Fig. 8. K means cluster of our data before applying the t-SNE component to our data points

VI. FUTURE WORK

There are several objectives which need to be completed for this research project. The matter of highest concern is to make a more robust k means algorithm. As of right now our clusters are all quite close together with quite a few outliers, which means that our results need to be improved. In order to get more uniform clusters, our accuracy measurement code needs to be properly implemented. We hope to have our recall @ K evaluator working soon, however, as of right now, our code does not properly measure accuracy. In addition, the radial basis function solver needs to created so that we are actually able to compare its performance against our very own auto-encoder based model. There are three main data sets which we will be working with. Currently, we have used the mnist data

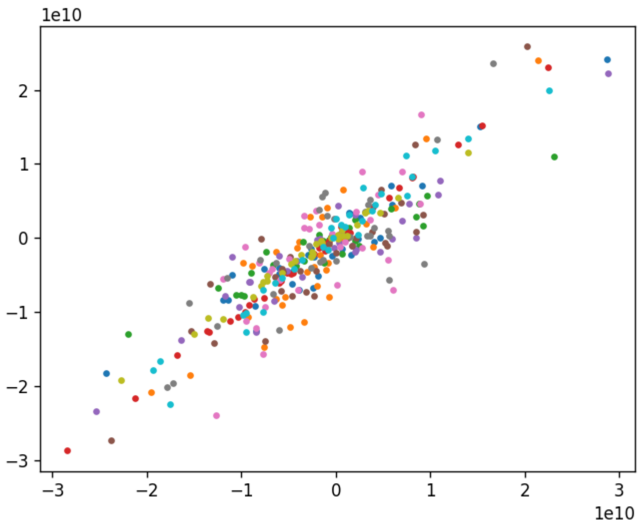

Fig. 9. K means cluster of our data before applying the t-SNE component to our data points

set of handwritten digits on our auto-encoder model. Once our model is fully up and running with the mnist data set, then we will run the cifar data set and the mass buildings data set on our model and the radial basis function solver. Lastly, we will explore the possibility of increasing our auto-encoder channels. This in turn may allow for better quality of our training images, making it easier to cluster and classify testing images.

VII. Conclusion

The objective of this task is to establish a mechanism that will provide for optimal image segmentation. By breaking down images into layers using an auto-encoder and then applying a k-means clustering algorithm to the dimensionally reduced images, we aspire to strengthen the link between neural networks and image segmentation. If this experiment is successful, the outcome will produce well defined clusters and highly accurate classification of images. In turn, this would become another way to improve current knowledge and procedures for future and present problems dealing with image segmentation.

Acknowledgment

This research was primarily funded and supported by the Research Experience for Undergraduates program at the University of Colorado Colorado Springs Machine Learning in Natural Language Processing and Computer Vision department under grant No. 1659788 from the NSF and grant No. 1359275 from the NSF.

References

[1] Bengio, Yoshua. Learning Deep Architectures for AI. Now Publishers Inc, 2009.

[2] Benson, C. C., V. L. Lajish, and Kumar Rajamani. "Brain tumor extraction from MRI brain images using marker based watershed algorithm." In Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference on, pp. 318-323. IEEE, 2015.

[3] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[4] Gong, Maoguo, Linzhi Su, Meng Jia, and Weisheng Chen. "Fuzzy clustering with a modified MRF energy function for change detection in synthetic aperture radar images." IEEE Transactions on Fuzzy Systems 22, no. 1 (2014):

[5] Kapoor, Dimple, and R. Kashyap. "Segmentation of Brain Tumor from MRI Using Skull Stripping and Neural Network." (2016).

[6] Kim, Jiwon, Jung Kwon Lee, and Kyoung Mu Lee. "Accurate image super-resolution using very deep convolutional networks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[7] Koltun, Philipp Krhenbhl Vladlen. "Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials." (2011).

[8] Kumar, S., et al.: Skull stripping and automatic segmentation of brain MRI using seed growth and threshold techniques pp. 422426 (2007)

[9] Le, Quoc V. "A Tutorial on Deep Learning Part 2: Autoencoders, Convolutional Neural Networks and Recurrent Neural Networks." Google Brain (2015).

[10] Mao, Xiao-Jiao, Chunhua Shen, and Yu-Bin Yang. "Image restoration using convolutional auto-encoders with symmetric skip connections." arXiv preprint arXiv:1606.08921 (2016).

[11] Meyer, Benjamin J., Ben Harwood, and Tom Drummond. "Nearest Neighbour Radial Basis Function Solvers for Deep Neural Networks." arXiv preprint arXiv:1705.09780 (2017).

[12] Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. Pattern Analysis and Machine Intelligence, IEEE Transactions on 12(7), 629639 (1990)

[13] Radhakrishna, Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. "Slic superpixels." Technical Report 149300, EPFL (2010).

[14] Ray, Siddheswar, and Rose H. Turi. "Determination of number of clusters in k-means clustering and application in colour image segmentation." In Proceedings of the 4th international conference on advances in pattern recognition and digital techniques, pp. 137-143. 1999.

[15] Schmidhuber, Jrgen. "Deep learning in neural networks: An overview." Neural networks 61 (2015)

[16] Stutz, David. "Understanding convolutional neural networks." In Seminar Report, Fakultt fr Mathematik, Informatik und Naturwissenschaften Lehr-und Forschungsgebiet Informatik VIII Computer Vision. 2014.