Content Based Image Resizing Using Deep Auto Encoder

Aman Kumar Sinha, Jay Gupta, Pritish Vilin Zunke
This work was supported by the Vellore Institute of Technology under the super vision of Dr. Geetha S

Abstract - This document grives an insight into the working of Content Based Image Resizing Using Deep Convolutional Auto Encoder. A comparative study is performed between energy map produced using the sobel operator and the auto encoder and comparision between their respective resized images is performed. To perfrom the rezising operations on an image, firstly generate a energy map for the image using the auto encoder-decoder convolutional neural network this approach is an extension of seam carving algorithm .The proposed method is built upon the concept of seam carving. BSDS 500 dataset was used to train and build the Deep Convolutional Auto Encoder.

1. INTRODUCTION

Now internet technology and rapid daily development there are various multimedia techniques, numerous images available on the internet are not frequently used by people. Due to increased use of a wide variety of displays devices such as computers, mobile phones and televisions. therefore, each one having a unique size and aspect ratio. If The resulting image is not scaled to the aspect ratio of the display devices, the image is distorted and this visual interpretation is inadequate.

Through seam carving we use a recursive approach to remove a path in the image where the path represents minimum cost path based on pixel values present in the particular path. Simillarly paths can be added to the image to essentially increase the dimentions of the image.

To carve the image two seams must be found one in each of the two dimentions i.e. one seam in horizontal direction and the other in the vertical direction, by neglecting any one of the directions we are basically resizing the image in only one of the directions. Choosing seam in one or more directions depends on a case to case basis, every application has a different requirement when it comes to resizing images.

Image features that remain on being resized completely depend upon the energy map generated, a good energy map will result in a good resized image where the important features of the image is retained and irrelevant features such as the background is removed.

For this paper we have used BSDS500(Berkeley Segmentation 500) which is used as a standard benchmark for contour detection and contains object contors, interior boundaries, background boundaries. This data set contains 500 images with carefully annotated energy map generated by us.

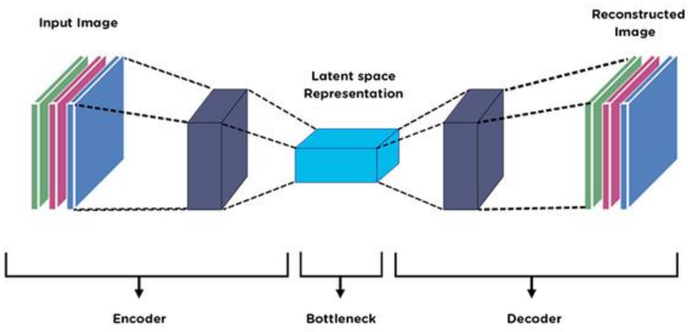
2. DEEP CONVOLUTIONAL AUTOENCODERS

Autoencoders are a type of ANN(Artificial Neural Network) which is used to produce a target output from the given input. When an image has to under go the encode-decode process it is first converted into a latent-space repesentation which is called bottelneck, later this is used to reconstruct the image from the bottleneck. An autoencoder is data-specific i.e. they can compress data only on the basis of their model, therefore right training of model is crutial in this scenario. They provide lossy compression which means that the output will be degraded compared to the original inputs.

Autoencoders learns by itself from the data that is provided hence making it easy to train model where is to be tarined for a specific target.

Autoencoders comprise of two basic components i.e. Encoder and Decoder. Through encoders input data in this case an image is compressed to a latent-space representation. The decoder later is used to decompress the image from that latent space back to the required image, here lossy reconstruction of image takes place.

The use of autoencoders is prominent as it is easy to share/transfer data in a latent-space representation which can be later reconstructed from the latent-space. The task of an autoencoder is to train/build a model in such a way that bottelneck will learn essential features of the image. By giving bottelneck image we are basically giving the autoencoder smaller dimentions of the input image what helps the autoencoder understand essential features of the image.



Deep Convolutional Autoencoder Architecture

Denoising of imput image, Dimentionality reduction of image are the two most prominent applications of autoencoders. With appropriate cnstraints autoencoders can give results that re better when compared with principle component analysis and other techniques.

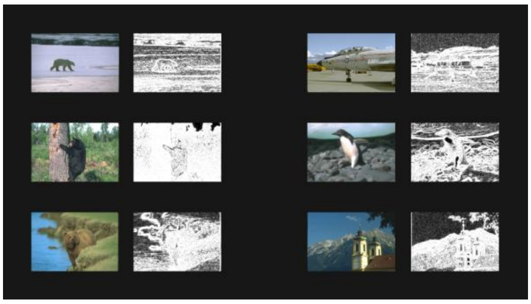
3. ENERGY MAP GENERATION

Table 1: Structure of layers, filter, stride for Deep Convolutional Autoencoder

Autoencoder	Unit Level	Conv Layer	Stride	Filter
Encode	Level 1	Conv2D Conv2D Maxpooling	1 1 -	3x3 3x3 -
Encode	Level 2	Conv2D Conv2D Maxpooling	2 1 -	3x3 3x3 -
Encode	Level 3	Conv2D Conv2D Maxpooling	2 1 -	3x3 3x3 -
Encode	Level 4	Conv2D Conv2D Maxpooling	2 1 -	3x3 3x3 -
Encode	Level 5	Conv2D Conv2D	2 1	3x3 3x3
Decode	Level 6	Upsampling2D	-	-
Decode	Level 7	Deconv2D Deconv2D Upsampling2D	1 2 -	3x3 3x3 -
Decode	Level 8	Deconv2D Deconv2D Upsampling2D	1 2 -	3x3 3x3 -
Decode	Level 9	Deconv2D Deconv2D Upsampling2D	1 2 -	3x3 3x3 -
Decode	Level 10	Deconv2D Deconv2D Deconv2D	1 2 1	3x3 3x3 1x1

The above table represents the the configuration of the Deep convolutonal auto encoder, the proposed algotihm uses 5 encoding layers and 5 decoding layers, to train the model. Convolutional layer is used to create a convolutional kernel that is convolved with the given input to give tensor of outputs, Maxpooling layer is also used to as a dimentionality reduction layer to extract the most relevant/significant features from the is image, since after every encodig layer executes it is important to get increase the size of the image in the decoding layer as the final image needs to be an energy map of size 256x256 for this reson we use the upsampling layer to scale up the values.In the decodig phase along with upsampling layer we also use transpose of convolutional layer to increase the size of image and get the final output of 256x256. The model is compiled using adam optimiser which is an extension of stocastic gradient descent method and is used to update the Deep convolutional autoencoder networks weights more efficiently , it is a combination of properties from AdaGrad and RmsProp algorithm that results in a algorithm capable of handling sparse gradients on noisy problems, it also uses momentum and adaptive learning rates to converge faster, for this model the learning rate was set as 0.0001 . A batch size of 50 and 10 epochs were used while traning the model.

Figure: 1: Structure of layers, filter, stride for Deep Convolutional Autoencoder



The coloured images to the left represent the original prictures from the BSDS 500 dataset and and the images towards the right represent the energy map obtained using solble operator which is used to train the Deep convolutional auto encoder model .

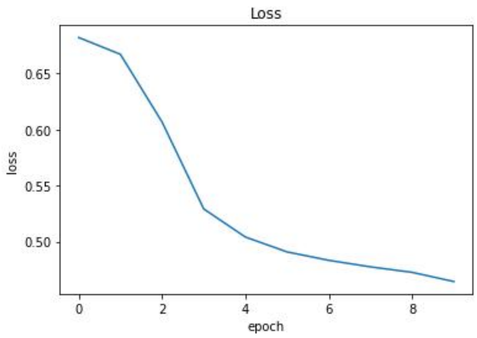
4. SEAM CARVING

The proposed method uses seam carving algorithm which was developed by Shai Avidan and Ariel Shamir, The algoritm is responsible to search for seams or paths of least importance that is seams/paths with minimum cost and remove such seams/paths to reduce the size of the image. This process of removal is done twice once in the horizontal direction and once in the vertical direction. Simillarly to increase the size of the image we insert the seams/path with minimum cost. The number of seams that need to be removed depends on the dimention of the original image and the factor by which the image needs to be resized . Finding the seams can be done through algorithms like Dijakstra's algorithm, graphcut algotiyhm, dynamic programming, or other greedy algorithms. Through seam carving we achieve image retargeting which helps in displaying images without distortion.

RESULTS

The experimental results obtained were as follows:

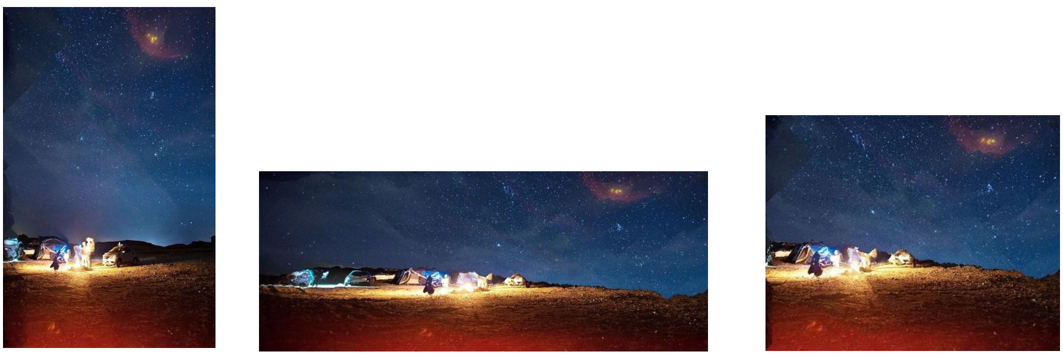
Figure 2: Loss plot for the built model



The above represented figure shows the loss plot obtained during training/building the model, the graph represents an almost linear decrease in the the loss.

Figure 3: Test image 1

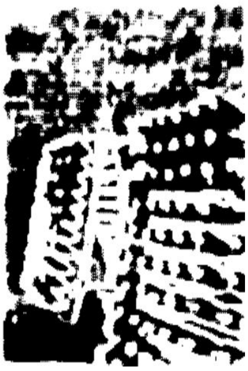
Initial input **Energy map for column-wise** **Energy map for row-wise**

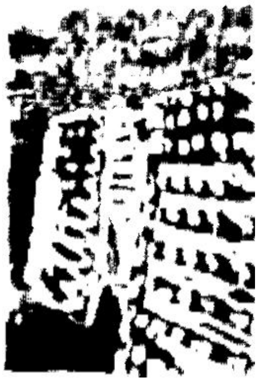


Seam-coulmnwise **Seam-rowwise** **Final output**

Figure 4: Test image 2


Initial input


Energy map for column-wise


Energy map for row-wise


Seam-coulmnwise


Seam-rowwise


Final output

Figure 5: Test image 3


Initial input


Energy map for column-wise


Energy map for row-wise

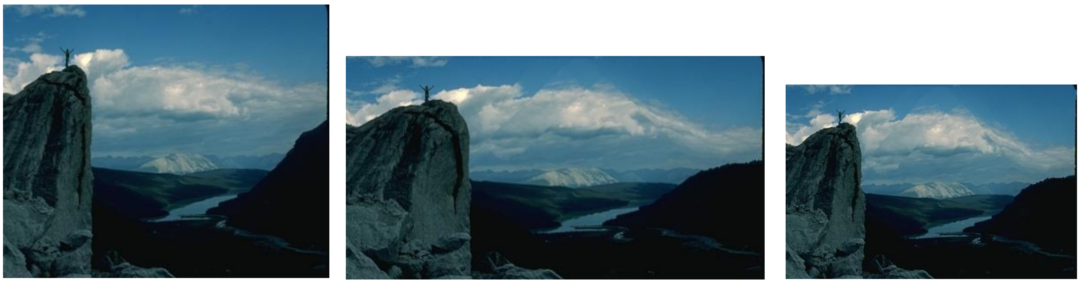

Seam-coulmnwise


Seam-rowwise


Final output

Figure 6: Test image 4

Initial input | Energy map for column-wise | Energy map for row-wise



Seam-coulmnwise | Seam-rowwise | Final output

5. CONCLUSION

The final output images were scaled down by a factor of 20% , the scaling was done column wise first and then for row wise, the final output is the combination of the two. Two energy maps were generated in the entire process, the first energy map was generated for column wise seam carving, and the second energy map is used for row wise seam carving. The model was trained on 500 images and the processing was done on a batch size of 50 images for 10 epochs, the current model produces substancial results when we sacle down the images by a factor of 0 to 40% but from scaling factor of greater than 50% we find that the images start loosing a few key features and images start looking a little distorted. To improve this drawbacks, we need to incerase the number of images in the dataset or increase the number of epochs. This solution is not a viable one as of now, since there is a lack of hardware resources and computational power for proper training of the model.

6. REFERENCES

[1] Avidan, S. and Shamir, A., 2007. Seam carving for content-aware image resizing. In *ACM SIGGRAPH 2007 papers* (pp. 10-es).

Song, E., Lee, M. and Lee, S., 2018. CarvingNet: Content-guided seam carving using deep convolution neural network. *IEEE Access* , *7* , pp.284-292.