Seam Carve Detection Using Convolutional Neural Networks

Mehtab Iqbal[1], Lei Chen[1](✉), Hengfeng Fu[2], and Yun Lin[2]

[1] College of Engineering and Computing, Georgia Southern University, Statesboro, GA, USA
lchen@georgiasouthern.edu

[2] College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China

Abstract. Seam carving is a form of content-aware image modification. This modification can vary from resizing to clipping of content within an image. This can be easily used to alter images to achieve steganographic goals or the propagation of misleading information. Deep learning, particularly Convolutional Neural Networks have become prolific in today's image-based intelligent systems. However, it has been found that convolutional networks specialized for image classification tend to perform poorly for steganalysis—specifically seam carving. In this paper, we propose a convolutional neural network architecture which is able to learn the nuances of seam carved images.

Keywords: Steganalysis · Seam carving · Convolutional Neural Networks

1 Introduction

Steganography is a form of covert communication which relies on utilizing a data container to hide messages [1]. With social media and their prolific use, channels for hidden communications are abundant [2]. An image speaks a thousand words they say. With the pervasiveness of smartphones, using images and image-based social media is very common [2]. Most of these images are in the compressed JPEG format [2].

These JPEG images can easily be used as containers to hide information. They may also be and are often used as secret message carriers. This embedded information in an image is neither visible to the viewer, nor is it detected by firewalls. Furthermore, it is very difficult to detect if any alteration has been made to an image without the original image to compare with. As a result, alongside steganography, content-aware image re-targeting algorithm such as Seam Carving has gained a lot of popularity [3].

Seam Carving relies on minimizing the energy cost of a seam through dynamic programming [4] to alter an image. It is a very efficient algorithm for resizing an image by removing the low energy content of the image [5]. It is often used to remove objects from an image [6]. This removal of objects from images can have a serious impact on the semantic value of an image [6] by altering the overall semantic content of the whole image, which is even more troublesome given the nature of information sharing in today's world—highly image-based through use of social media and the internet. There have been many algorithms and models to detect seam carving, for example, Liu [7]

S. Liu and G. Yang (Eds.): ADHIP 2018, LNICST 279, pp. 392–407, 2019.
https://doi.org/10.1007/978-3-030-19086-6_44

have used re-compression techniques to detect seam carving, and Ke et al. [3] observed the seam patterns by seam carving the image again, in order to determine seam carved effects in an image.

Although many novel processes and algorithms to detect seam carving exits, deep learning has only recently started being used in steganalysis [8]. This is because steganalysis requires a detection of stego-noise, which is a very weak signal and visual processing networks are not geared towards detecting them [9]. With increasing research in deep learning based steganalysis, in particular, the use of modified convolutional neural networks, [8] and the advances in deep learning frameworks [10] it is becoming increasingly possible to train models for use in other applications. Some frameworks such as Caffe [11], Theano [12], Tensorflow [13], etc. have made available deep learning libraries that leverage the capabilities of mobile devices, allowing mobile applications to load trained models and utilize it towards classification problem without the cost of transferring a large amount of data back to the server for processing [10].

Using these advancements, whilst keeping in mind, the limitation of a portable device, we propose a Tensorflow based convolutional model that can be sufficiently trained so that the model can be loaded and made portable for use in various applications. To achieve this, we would look into local binary patterns as used by Yin et al. [6], and use the structural design proposed by Xu et al. [8] which utilizes a modified convolutional layer for feature extraction.

2 Literature Review

2.1 Steganography and JPEG Images

The mechanism of sending secret messages by hiding them in innocuous medium consequently making the communication invisible is known as Steganography [14, 15]. Steganography hides the very existence of secret messages and masks the presence of communication by making the true message not discernible to the observer [16, 17]. While steganographic methods strive for high security and capacity, usually these techniques are not concerned with robustness [18]. Several techniques of image steganography include spatial domain, transform domain-based methods and spread spectrum method [19, 20]. Images that have hidden messages or the carrier images in steganography are called the cover images [19].

Although cover images can have many formats, the most popular choice for steganographers are digital files, that can compress images with a small loss of perceptual quality [1]. In addition, the tools used for steganography encompasses bit-wise methods that can apply a least significant bit (LSB) insertion and noise manipulation [21]. Such tools are known as image domain tools [21]. JPEG is an image type that can ensure minimum data loss through direct manipulation and recovery by using the Discrete Cosine Transform (DCT) [21]. In JPEG images, information is hidden by modulating the rounding choices of the DCT coefficients thus making detection of such messaged difficult [14]. To combat these issues and the increasing popularity of steganographic methods, a new field of Steganalysis is established [20]. This technique relies on the changes in the statistical characteristics of an image to detect the embedded data [20].

2.2 Seam Carving

Seam carving is a technique used towards content-aware scaling, in other words, pixels on the least significant seams are removed or inserted, in order to alter the size of an image [4]. This is achieved by defining an energy function based on the gradient in order to identify the seams containing the lowest energy [22, 23]. For an image I of dimension n × m, the vertical seam is defined as

$$s^x = \left\{s_i^n\right\}_{i=1}^n = \{(x(i), i)\}_{i=1}^n = s.t\,\forall i,\ x(i-1)| \leq 1 \tag{1}$$

Where x maps pixels such that, $x : [1, \ldots, n] \rightarrow [1, \ldots, m]$ in other words, i denotes the row coordinate, and the corresponding column coordinate is given by $x(i)$. Similarly, the horizontal seam is given by the equation

$$s^y = \left\{s_j^n\right\}_{j=1}^n = \{(y(j), j)\}_{j=1}^n = s.t\,\forall j,\ x(j-1)| \leq 1 \tag{2}$$

Where y maps pixels $y : [1, \ldots, m] \rightarrow [1, \ldots, n]$. An image can be altered by modifying the seams containing the least energy seam. The energy function for a vertical seam is denoted by the energy function $E(s)$.

$$E(s) = \sum_{i=1}^{n} e((x(i), i)) \tag{3}$$

Where the energy function for each pixel is given by $e(I)$.

$$e(I) = \left|\frac{\partial}{\partial x} I\right| + \left|\frac{\partial}{\partial y} I\right| \tag{4}$$

Similarly, the parameters just need to be flipped and s^y used instead of s^x in order to calculate the least every horizontal seam. Finally, the lowest energy seam can be calculated by minimizing the energy function which can be achieved through dynamic programming [6] and a minimum energy matrix M can be build using the relationship below:

$$M(i,j) = e(i,j) + min(M(i-1, j-1), M(i-1, j), M(i-1, j+1)) \tag{5}$$

Figure 1 compares image resizing using Seam Carved technique versus reshaping. Furthermore, Fig. 2 shows how seam carving can be used to remove objects from images [24].

2.3 Steganalysis and Detection of Seam Carving

The technique for detecting and analyzing files that are potential carrier files and have hidden data using Steganography is called Steganalysis. The objectives of this measure can be three levels: detecting, extracting and disabling or destroying hidden messages.

Fig. 1. Seam carving vs regular resizing

Fig. 2. Seam carving to remove the object from the image

There are several ways of performing steganalysis on carrier files including the Raw Quick Pair (RQP) technique, Regular-Singular analysis (RS) technique, Histogram Characteristic Function (HCF) technique [19].

When there is a change to close color pairs on high-color images, it can indicate that the image has an embedded message. The raw quick pairs (RQP) technique is based on an observation and assumption. When there is an observed change in colors close to the pairs on high-color, it can indicate that the image has an embedded message. This technique also assumes that the total number of pixels is significantly larger than the number of unique colors in the cover image [25]. Although this technique shows the existence of a message, it cannot calculate the length of the messages. Its limitations include the cover image to have less than 30% unique colors of the total pixel [26] and neither can this technique be applied to grayscale images, as they have less than 256 colors and is not enough to reflect changes in an insertion operation.

The Regular Singular (RS) analysis technique is built on the observation that randomizing LSB of the images influences its smoothness [27], whereas the Histogram Characteristic Function (HCF) is based on investigating the characteristics of image histograms and the effect on histograms caused by embedding secret images. Regular Singular analysis technique both finds and calculates the length of the message hidden in an image. However, the complexity time of RS is O (n), where n is the number of pixels in an image.

Certain improvements were made to the HCF technique [28, 29]. The first improvement calibrates by down-sampling the images and the second technique combines two adjacent pixels as opposed to averaging four adjacent pixels and is known as alternative calibration. Adjacency histogram, another improvement made to this technique can detect grayscale images [28]. To make the grayscale histogram parser, the two-dimensional adjacency histogram is used [26]. This histogram uses the pixel intensity of two adjacent pixels as one data point. There is no one size fit all solution for steganalysis of any sort, this is especially true for seam carving. Research on seam carving forensics has been done since 2009 [3]. Sarkar et al. [5] theorized that if enough seam is changed within an image, then the inter-pixel correlation and co-occurrence matrix should undergo sufficient change. This change is expected to be reflected in the local block-based DCT coefficients of the JPEG image [30]. They utilized a Markov random process in order to develop a probability matrix to represent the process and trained an SVM using 50% of the data. They achieved about 80% accuracy for their work.

Additionally, Fillion and Sharma [31] proposed statistical features that include bias of energy distribution, the dispersal of seam behavior, and the affection of wavelet absolute moments. Their model managed to attain an accuracy of up to 91.3% for as low as 20% seam-carved images. Ryu et al. [32] trained their SVM model using average column energy, average row energy, average energy, max seam. Their model achieved accuracy between 71.52% and 93.5% but failed to detect object removal.

Towards the detection of content-aware alteration in JPEG images, Qingzhong Liu [33, 34], merged shift-recompression based features in the spatial domain, and neighboring joint density in DCT domain together. Wei et al. [35] divided images into 2×2 mini-squares with pairing 2×3 candidate patches to observe possible effects of seam carving. Then, taking into account the patch transition probability they extracted a 252 dimension feature set which they used to train an SVM classifier, resulting in up to 95.8% accuracy on 20% seam carved images. Yin et al. [6], extracted the local binary pattern, generally used for texture classification, and reached an accuracy of 97% in best cases at 21% seam carved images.

2.4 Convolutional Neural Networks and Steganalysis

Convolutional neural networks (CNN) have been used very widely in computer vision and has made many large achievements [8]. However, steganalysis and visual processing for artificial intelligence are very different tasks [9]. Qian et al. [9] have tested several visual processing CNNs towards steganalysis and the consensus is that they did perform up to the task. They also, in their 2015 paper [9] proposed a modification for the convolutional layer in a CNN to detect stego-content. They called this Gaussian-Neuron CNN (GNCNN), and it relied on a Gaussian function as the activation function.

$$f(x) = e^{-\frac{x^2}{\sigma^2}} \tag{6}$$

where, σ determines the width of the curve.

According to them, this function is supposed to generate a significant positive response when the input intervals are small [9]. Xu et al. [8] proposed a whole network architecture based on the GNCNN and introduced batch normalization prior to a $\tanh(x)$ activation function.

They also used a High Pass Filter (HPF) on the image before using it as an input in their neural network. After three stages of convolution, they activated the last convolutional layer with a linear rectifier (ReLu) function, followed by a fully connected layer and Softmax for classification. They tested their architecture on S-Uniward [36], and HILL [37] utilized steganography. They managed to achieve between 58.44% and 79.24% accuracy on HILL, and 57.33% and 80.24% on SUNIWARD.

Recently, Sedighi and Fridrich [38] introduced the use of histogram layer into a convolutional model in order to achieve remarkable learning again S-UNIWARD based stego images. In fact, it is convolutional network is increasingly being used to tackle steganalysis tasks.

3 Methodology

3.1 Data

We collected a dataset from Sam Houston State University's [39] image database, which contains a set of 1000 images. Five hundred of these are untouched JPEG images and the other five hundred were manipulated versions of the images using seam carving at the quality of 75. All original images are everyday pictures of dimensions 1234×1858, or 1858×1234. The seam carved images ranges from resizing in either horizontal, and/or vertical direction, removed content, or other forms of modifications. Figures 3 and 4 shows some samples of the images from the dataset. Figure 3 contains the original images, and Fig. 4 show the corresponding seam carved images. Some changes are obvious, but others contain subtle signal changes, not visually perceptible.

We excluded all images that were only resizing of the original image, or in other words, if there were dimensional alteration between the original and the seam carved image, we excluded it from the dataset. There were a total of 8 such images and that reduced our dataset size to 992 images. Our model constrained us to train on smaller resized versions of the image. Excluding resized images from our dataset eliminates ambiguity in labeling when we resize our images for uniformity. This is explained in more detail in the training section of this paper.

3.2 Feature Exploration

We considered a few random samples to compare and explore the differences within the image pair, to determine the feature differences. We started with the images shown in Fig. 5.

First, we generated histograms for the images and then compared the histograms of the two images to find subtle differences in the shape. Since we were interested in the luminance distribution of the images, we started by converting the image from the RGB color space to grayscale. Also, in order to capture all pixel values, we did not threshold but used 256 bins as seen in Fig. 6. Then, we decided to see if each color channel elucidated more information. We treated each channel similar to our grayscale image

Fig. 3. Original image preview

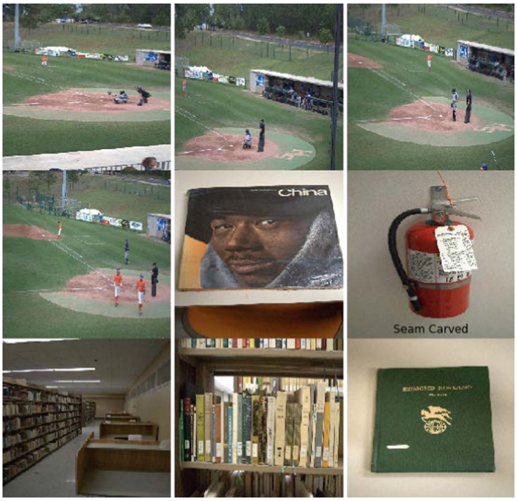

Fig. 4. Seam carve image preview

and superimposed all the resulting histogram data onto a single graph as seen in Fig. 7. Although subtle, we see a change in direction in the red and green channel near the center peak.

Next, we wanted to explore how each pair of color was distributed among each pixel. For each pairwise combination of channels, we used 32 bins to see the distribution. The outcome is depicted in Fig. 8, and this set of data was yet another set of feature used in the model.

Fig. 5. Sample from the dataset

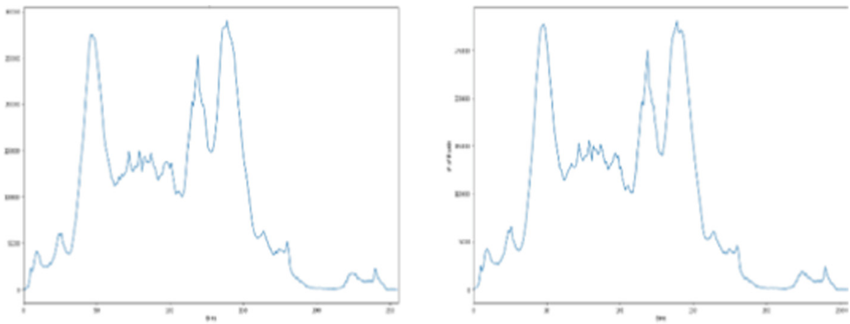

Fig. 6. Grayscale histogram showing luminance distribution between the original (left) and seam carved (right) images

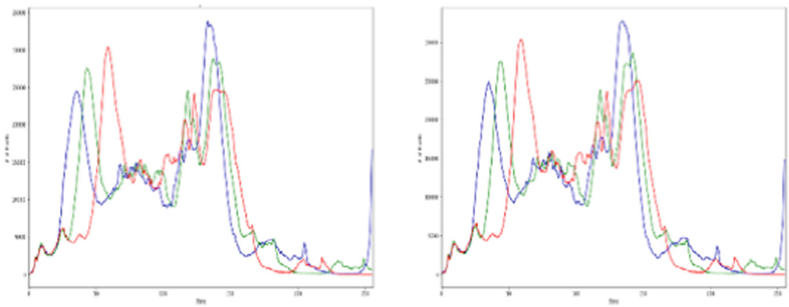

Fig. 7. Color distribution histogram - original (left), and seam carved (right) (Color figure online)

Finally, we also considered the use of all three channels to determine a 3-dimensional histogram and included it in our model. We will discuss all the parameters in the next section on the architecture of our network.

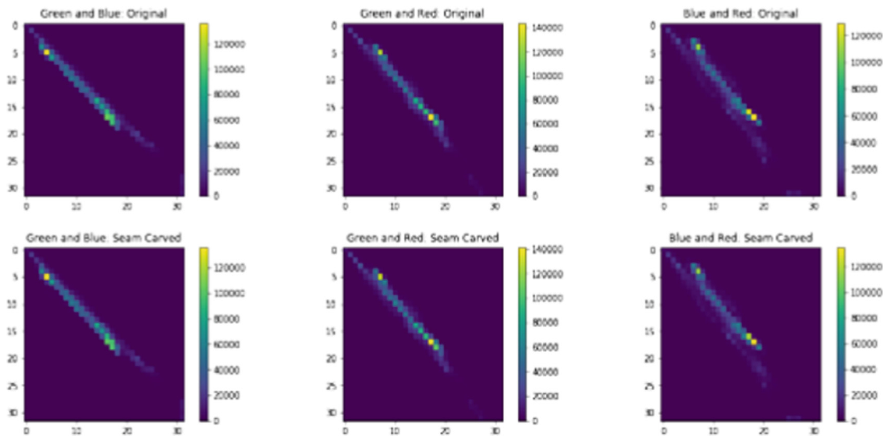

Fig. 8. 2D histogram comparison between the original image (above), and seam carved image (below)

3.3 Image Processing

In order to train our model, we first extracted features from the image or processed them for input. We first resized all our images to 256 $\times$ 256 for ease of computation. We then extracted all the histogram features as explained in the feature exploration section and flattened them to vector matrices. The shape of the 2-dimensional color matrix is (, 768), the shape of the superimposed color channel matrix is (, 3072), and for the 3-dimensional histogram matrix is (, 512). Once we had our vectors, we calculated the local binary representation (LBP) of each channel of an image, and the result example is shown in Fig. 9. Once we calculated the LBP, we stitched the channels back together to feed into our convolutional model.

Fig. 9. LBP representation of each channel on the original image

Finally, for our parallel network, we convolved our image with a High Pass Filter (HPF), using the Gaussian high pass filter with a sigma value of 1. The result of the HPF operation is shown in Fig. 10.

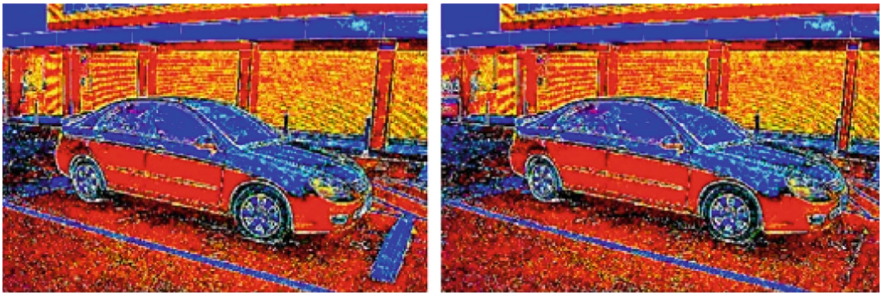

Fig. 10. Gaussian high pass filter result for the original image (left) compared with the seam carved image (right)

The next section explains in detail the network architecture, and how we used our three inputs to train our model.

3.4 Structure and Architecture of Network

The neural network model consists of two parallel convolutional networks, merged with our histogram data that leads to a Softmax activated classifier as depicted in Fig. 11.

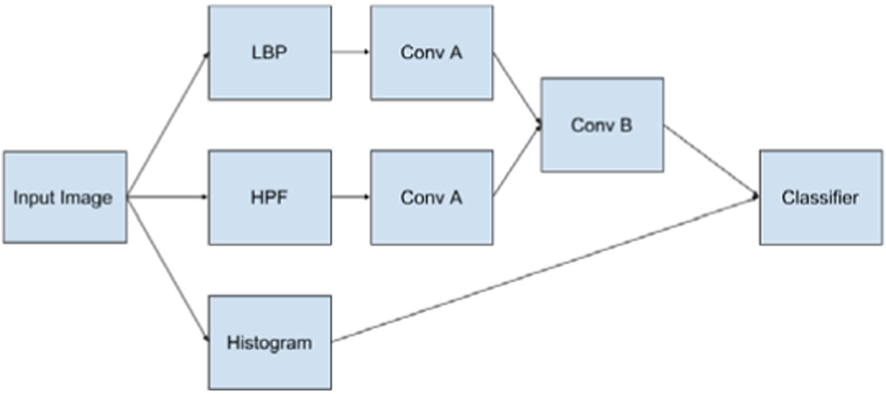

Fig. 11. Overall architecture

Each input image is processed as outlined in the image processing section of this paper. The LBP and HPF images each are used as input into two parallel identical convolutional neural networks (Conv A), whose architecture is shown in Fig. 12. Conv A is a feature learning stage of the neural network, where each convolutional layer is followed by a batch normalization layer, which is then activated by the tanh(x) activation function, then average pooled using a 5×5 kernel, traversed using stride size of 2.

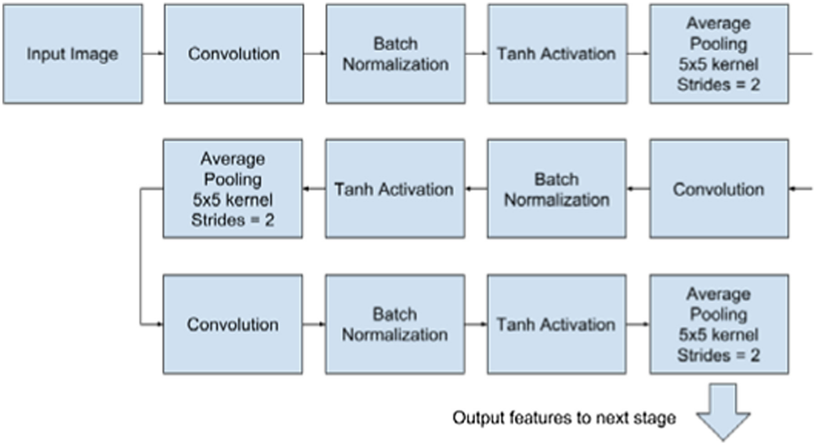

Fig. 12. Conv A, the convolutional feature learning architecture

The output of each parallel Conv A is concatenated and further trained through a secondary smaller network (Conv B). Conv B is depicted in Fig. 13. Conv B consists of a single convolutional layer activated by tanh(x) followed by a Max Pooling layer, which is then connected to a fully connected layer (Dense) by 512 neurons to the classification layer shown in Fig. 14.

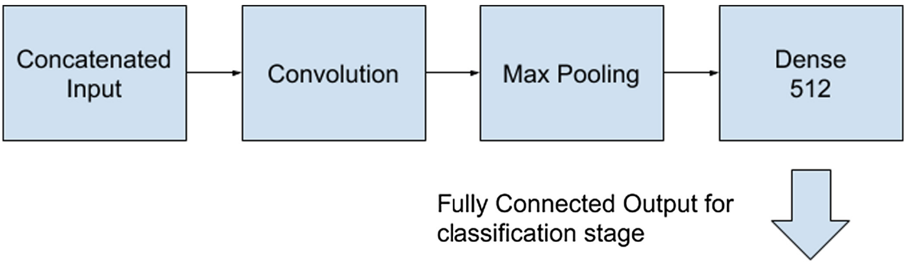

Fig. 13. Conv B, preparing the convolutional data for classification

The classification layer (Fig. 14) contains 6 fully connected layers (Fig. 14 shows a summary of the connections, since all connections would not depict very well in image), in reducing number of neurons per layer. The first layer contains 512 neurons, following by 256 neurons, and then 128, 64, 32 respectively. Finally, the last layer consists of binary neurons activated by the Softmax function to classify into our expected classes of 0, and 1, representing "Unaltered", and "Seam Carved" respectively.

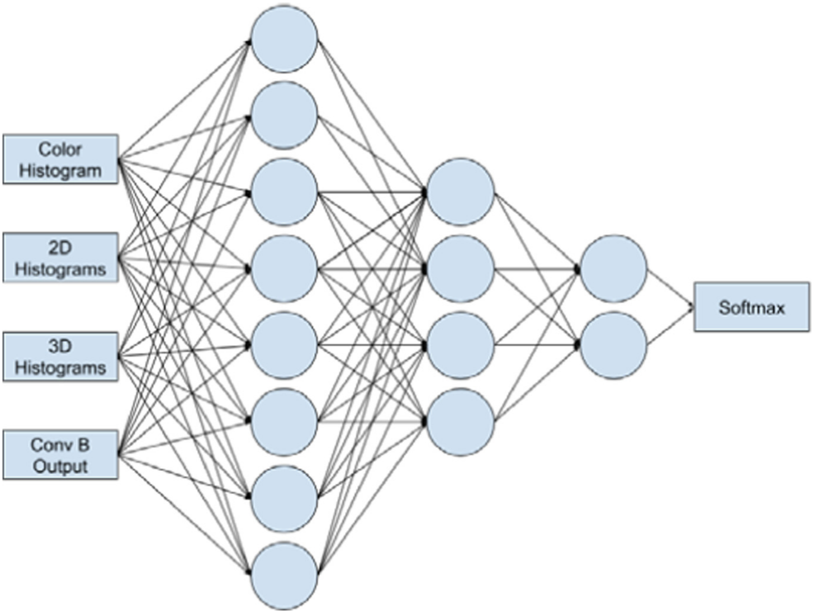

Fig. 14. Classification layer of the network

3.5 Training

For our training setup, we used a variant of the Adam optimizer [40, 41], with an initial learning rate of 0.0003 and no decay. We divided our data into random 80% for training and 20% for validation. The training was done over 90 epochs but stopped early at 43 epochs. Figure 15 shows the validation accuracy, and loss graph from our training.

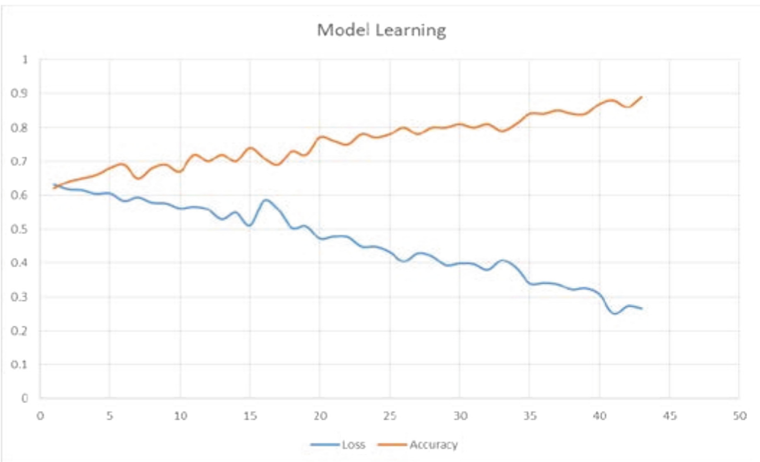

Fig. 15. Training validation accuracy and loss

However, before training our entire model, we experimented with training each part of our model. We connected our Conv A (Fig. 12) to the classification layer (Fig. 14) and used images without any filtering or energy extraction.

We then tested with only the LBP image as input. Followed by the use of the Gaussian HPF image as input. To compare our model. The results of each experiment are discussed in the next section.

4 Results

Our model achieved a validation accuracy of 89% at a 26.7% loss. However, during training, the model did achieve up to 94.4% accuracy at some stage at a loss of 16.4% but it did not sustain for too long. The convolutional layer on its own with either an LBP or a Gaussian HPF image achieved to achieve a good learning curve and accuracy of up to 73% on the validation data.

However, images without the appropriate filter did not perform very well. Seam carving produces very subtle changes when retargeting an image. Typical visual processing models such a convolutional neural network is not able to learn any significant features which allow the model to generalize. In fact, when the image without any processing was used to train our Conv A, the model failed to learn altogether with loss ratio rising above 1.5 with an accuracy of around 48%, which, for a binary classification is random. The complete model, however, along with both the LBP and HPF image, produced remarkable results, yield an accuracy of 89%.

A crucial point to note is how the addition of the flat histogram information gave the model a boost in accuracy and allowed it to learn much faster. Finally, from the results there seem to exist a significant correlation between the 3-dimensional histogram and the seam carved image.

5 Conclusion

Seam carving is a popular content-aware image retargeting algorithm and is sometimes used for nefarious purposes. Previous studies have used many sophisticated processes, ranging from signal processing to Markov probability distribution in order to detect seam carved images with very high accuracy. In this paper, we wanted to leverage some filtering and convolving kernels produced by other researches to enable the training of a deep neural network.

We were particularly interested in using deep learning libraries available for smartphone applications so that the model can portable in mobile devices. To this goal, we have managed to train a Tensorflow based neural network model, which can classify seam carved images which contain information addition or removal. Further research may include simplification of the model by investigating the hyper-parameters to reduce training time and make the model more adaptable.

References

1. Chen, M., Sedighi, V., Boroumand, M., Fridrich, J.: JPEG-phase-aware convolutional neural network for steganalysis of JPEG images. In: Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security, pp. 75–84. ACM, New York (2017)
2. Holub, V., Fridrich, J.: Low-complexity features for JPEG steganalysis using undecimated DCT. IEEE Trans. Inf. Forensics Secur. **10**, 219–228 (2015)
3. Ke, Y., Shan, Q., Qin, F., Min, W., Guo, J.: Detection of seam carved image based on additional seam carving behavior. Int. J. Signal Process. Image Process. Pattern Recognit. **9**, 167–178 (2016)
4. Avidan, S., Shamir, A.: Seam carving for content-aware image resizing. ACM Trans. Graph. (TOG), 10 (2007)
5. Sarkar, A., Nataraj, L., Manjunath, B.S.: Detection of seam carving and localization of seam insertions in digital images (2009)
6. Yin, T., Yang, G., Li, L., Zhang, D., Sun, X.: Detecting seam carving based image resizing using local binary patterns. Comput. Secur. **55**, 130–141 (2015)
7. Liu, Q.: An approach to detecting JPEG down-recompression and seam carving forgery under recompression anti-forensics. Pattern Recognit. **65**, 35–46 (2017)
8. Xu, G., Wu, H.Z., Shi, Y.Q.: Structural design of convolutional neural networks for steganalysis. IEEE Signal Process. Lett. **23**, 708–712 (2016)
9. Qian, Y., Dong, J., Wang, W., Tan, T.: Deep learning for steganalysis via convolutional neural networks. In: Media Watermarking, Security, and Forensics, p. 94090 J. International Society for Optics and Photonics (2015)
10. Alzantot, M., Wang, Y., Ren, Z., Srivastava, M.B.: RSTensorFlow: GPU enabled TensorFlow for deep learning on commodity android devices. In: Proceedings of the 1st International Workshop on Deep Learning for Mobile Systems and Applications, pp. 7–12. ACM, New York (2017)
11. Jia, Y., et al.: Caffe: convolutional architecture for fast feature embedding. In: Proceedings of the 22Nd ACM International Conference on Multimedia, pp. 675–678. ACM, New York (2014)
12. Bergstra, J., et al.: Theano: deep learning on GPUs with python. In: NIPS 2011, BigLearning Workshop, Granada, Spain, pp. 1–48. Citeseer (2011)
13. Abadi, M., et al.: TensorFlow: a system for large-scale machine learning. In: OSDI, pp. 265–283 (2016)
14. Anderson, R.J., Petitcolas, F.A.P.: On the limits of steganography. IEEE J. Sel. Areas Commun. **16**, 474–481 (1998)
15. Ge, H., Huang, M., Wang, Q.: Steganography and steganalysis based on digital image. In: 2011 4th International Congress on Image and Signal Processing, pp. 252–255 (2011)
16. Lu, T.-C., Chang, C.-C.: Lossless nibbled data embedding scheme based on difference expansion. Image Vis. Comput. **26**, 632–638 (2008)
17. Lin, C.-C., Hsueh, N.-L.: A lossless data hiding scheme based on three-pixel block differences. Pattern Recognit. **41**, 1415–1425 (2008)
18. Farid, H.: Detecting steganographic messages in digital images. Report TR2001-412 Dartmouth College, Hanover NH (2001)
19. Qian, T., Manoharan, S.: A comparative review of steganalysis techniques. In: 2015 2nd International Conference on Information Science and Security (ICISS), pp. 1–4 (2015)
20. Sabeti, V., Samavi, S., Mahdavi, M., Shirani, S.: Steganalysis and payload estimation of embedding in pixel differences using neural networks. Pattern Recognit. **43**, 405–415 (2010)

21. Johnson, N.F., Jajodia, S.: Steganalysis of images created using current steganography software. In: Aucsmith, D. (ed.) IH 1998. LNCS, vol. 1525, pp. 273–289. Springer, Heidelberg (1998). https://doi.org/10.1007/3-540-49380-8_19
22. Dong, W., Zhou, N., Paul, J.-C., Zhang, X.: Optimized image resizing using seam carving and scaling. In: ACM SIGGRAPH Asia 2009 Papers, pp. 125:1–125:10. ACM, New York (2009)
23. Rubinstein, M., Gutierrez, D., Sorkine, O., Shamir, A.: A comparative study of image retargeting. In: ACM SIGGRAPH Asia 2010 Papers, pp. 160:1–160:10. ACM, New York (2010)
24. Seam Carving—skimage v0.15.dev0 docs. http://scikit-image.org/docs/dev/auto_examples/transform/plot_seam_carving.html#id2
25. Fridrich, J., Long, M.: Steganalysis of LSB encoding in color images. In: 2000 IEEE International Conference on Multimedia and Expo, ICME2000, Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No. 00TH8532), vol. 3, pp. 1279–1282 (2000)
26. Nissar, A., Mir, A.H.: Classification of steganalysis techniques: a study. Digit. Signal Process. **20**, 1758–1770 (2010)
27. Fridrich, J., Goljan, M., Du, R.: Detecting LSB steganography in color, and gray-scale images. IEEE Multimed. **8**, 22–28 (2001)
28. Ker, A.D.: Steganalysis of LSB matching in grayscale images. IEEE Signal Process. Lett. **12**, 441–444 (2005)
29. Ker, A.D.: Resampling and the detection of LSB matching in color bitmaps. In: Security, Steganography, and Watermarking of Multimedia Contents VII, pp. 1–16. International Society for Optics and Photonics (2005)
30. Shi, Y.Q., Chen, C., Chen, W.: A Markov process based approach to effective attacking jpeg steganography. In: Camenisch, J.L., Collberg, C.S., Johnson, N.F., Sallee, P. (eds.) IH 2006. LNCS, vol. 4437, pp. 249–264. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-74124-4_17
31. Fillion, C., Sharma, G.: Detecting content adaptive scaling of images for forensic applications. Presented at the February 4 (2010)
32. Ryu, S.-J., Lee, H.-Y., Lee, H.-K.: Detecting trace of seam carving for forensic analysis. IEICE Trans. Inf. Syst. **97**, 1304–1311 (2014)
33. Liu, Q., Cooper, P.A., Zhou, B.: An improved approach to detecting content-aware scaling-based tampering in JPEG images. In: 2013 IEEE China Summit and International Conference on Signal and Information Processing, pp. 432–436 (2013)
34. Liu, Q., Li, X., Cooper, P.A., Hu, X.: Shift recompression-based feature mining for detecting content-aware scaled forgery in JPEG images (2012)
35. Wei, J.-D., Lin, Y.-J., Wu, Y.-J.: A patch analysis method to detect seam carved images. Pattern Recognit. Lett. **36**, 100–106 (2014)
36. Holub, V., Fridrich, J., Denemark, T.: Universal distortion function for steganography in an arbitrary domain. EURASIP J. Inf. Secur. **2014**, 1 (2014)
37. Li, B., Wang, M., Huang, J., Li, X.: A new cost function for spatial image steganography. In: 2014 IEEE International Conference on Image Processing (ICIP), pp. 4206–4210. IEEE (2014)
38. Sedighi, V., Fridrich, J.: Histogram layer, moving convolutional neural networks towards feature-based steganalysis. Electron. Imaging. **2017**, 50–55 (2017)
39. Liu, Q., Chen, Z.: Improved approaches with calibrated neighboring joint density to steganalysis and seam-carved forgery detection in JPEG images. ACM Trans. Intell. Syst. Technol. **5**, 63:1–63:30 (2014)

40. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. ArXiv14126980 Cs (2014)
41. Reddi, S.J., Kale, S., Kumar, S.: On the convergence of adam and beyond (2018)