Digital Image Colorization, Style Transfer and Image Inpainting using Deep Learning

Shivani Dere[1], Anurag Chaudhari[1], Adarsh Laddha[1], Yashaswini Deora[1], Dhanalekshmi Yedurkar[3]

[1]Computer Science and Engineering, MIT School of Engineering, MIT-ADT University, Pune, India
[3]Professor, Dept. of Computer Science and Engineering, MIT School of Engineering, MIT-ADT University, Pune, India

---***---

Abstract - *In the era, where colors and style fascinate everyone, more emphasis is given on aesthetics and beauty. This research paper proposes a deep learning method based on Convolutional Neural Network (CNN) to develop an application for converting images into artistic style, colorization of the image, and inpainting of image. The proposed method combines all the three applications into a single web-based application termed as Neuron. Here, colorization is performed by CNN, image inpainting is obtained by Generative Adversarial Network (GAN), style image is generated by Neural Style Transfer (NST) techniques. We trained the distinct models for all three applications and produced qualitative and quantitative comparisons with other traditional approaches to endorse this approach.*

***Key Words*: CNN, Deep Learning, Neural Style Transfer, Image Colorization, Image Inpainting, GAN, RELU, feed forward network**

1.INTRODUCTION

Currently, image processing using a deep learning technique is an emerging area and is gaining greater popularity especially in improving the quality of digital images. It has also showcased a greater performance in feature extraction and classification. The important attributes of a digital image are saturation, brightness, contrasts, sharpness. This information has the ability to make a picture or ruin its characteristics. Hence, these characteristics need to be well preserved without degrading the image quality. Conventional techniques rely on human interaction by utilizing countless color scribbles, or by mingby performing segmentation techniques.

Transferring Style from one image to the other is one of the major concerns in texture transfer. In texture transfer, the main motive is to combine a texture feature from an input image to the expected image. This is done by preserving the semantics of the target image. Various methods have been proposed by employing non-parametric algorithms. They synthesize the real like natural textures by resampling the pixels of a designated source texture

Inpainting is a procedure in which is used to recover the lost fragments of an image and to recreate them. Image inpainting is applied for restoring old images, damaged films, and to edit an image in order to eliminate undesired image content. Various researchers have presented several techniques for addressing these difficulties.

Richard Zhang et al. proposed a fully automated way to produce lively and real like images

The method utilizes a feed-forward type of CNN. The authors evaluated the algorithm by employing a colorization Turing test. The results showed that the technique presented significantly improved in comparison with the traditional method.

Leon A. Gatys et al. proposed an algorithm to generate images of high quality that combine the content of an arbitrary image with the appearance of different well-known artistic images. The author proposed a new understanding of the deep image depictions learned by Convolutional Neural Networks and signified their potential for high-level image synthesis and manipulation.

Currently, deep learning and neural networks have obtained a lot of recognition among researchers in the area of image processing. CNNs have proved to be a successful method in image recognition, color recognition, image sharpening and restoration, pattern recognition, and image generation. Any image layer has useful data regarding input images at distinct levels. While every layer is generated due to applying several image filters, each layer of the input image abstracts any particular feature to the preceding layer.

2. LITERATURE REVIEW

Previous work in image inpainting involves training the convolution neural network to predict the missing pixels and another restores the missing part of the image on patch level to get a restored image but, these techniques look in background pixels to fill missing patches in the background Hence, they are unable to fill missing pixels of complex structures such as faces, objects. Jiahui Yu et.al presents a two-stage feed-forward generative neural network in which the First stage consists of filling the rough values in missing pixels and second is the contextual attention layer. The key idea behind the contextual stage is to fill missing patches of image on the basis of known patches.

They introduce style augmentation in this paper, a new type of data augmentation based on random style transfer for increasing the robustness of Convolutional Neural Networks (CNN) in classification and regression applications. During training, texture, contrast, and color are randomly generated but shape and semantic content are preserved. This is accomplished by performing style randomization via an arbitrary style transfer network. instead of constructing target style embeddings from a style image, they sample them from a multivariate normal distribution. Working with only styled pictures reduces accuracy, therefore accuracy should be balanced if only styled images are used. [9]

In Poisson denoising, this research suggests a deep learning demising network that outperforms established benchmark algorithms statistically significantly. The demising network has the potential to yield statistically significant results. Future studies could look into whether the network can learn to distinguish between different types of noise. for example, Gaussian noise or a random noise with undetermined properties. [16]

Here Instead, they present a fully automated data-driven strategy for coloring grayscale photos in this work. We used deep learning to create a model that can predict colors in a greyscale image. Image datasets and deep Convolutional Neural Networks are used in this method (CNN). The model is trained using the training dataset, and it is then tested using test data images. The pixel deviation from the original photos is calculated, and the results are compared. The pixel departure from the original images is also minimized using an efficient technique. [2]

In this study, a new approach for colorizing grayscale photographs is proposed. By transferring colors from a reference (source) color image to a destination grayscale image, the suggested technique colorizes grayscale images. A feed forward artificial neural network (ANN) is built and trained by mapping pixels from a grayscale space (grayscale version) into the color space of a reference (source) color image with a mood similar to the destination grayscale image. [17]

This research presents a deep neural network that predicts pixels in an image in two spatial dimensions consecutively. Our method encodes the entire collection of dependencies in the image by modelling the discrete probability of the raw pixel values. Given that these models improve as they grow in size and that there is virtually limitless data to train on. We rely on additional computation and larger models to improve the results even further. [18]

The usage of partial convolutions, where the convolution is masked and renormalized to be conditioned on just valid pixels, is proposed in this study. As a part of the forward pass, we also provide a technique to automatically build an updated mask for the next layer. For irregular masks, our model outperforms existing techniques. If the image's defects are huge, the model will struggle to perform well on it. [21]

There are certain drawbacks, such as the model's accuracy could be improved.

Here, the paper presents three distinct techniques of Image processing namely, Image Colorization, Neural Style transfer and Inpainting of an image. Coloring a grayscale image is a tedious task to be performed manually. Hence, this research paper targets on finding a solution based on deep learning.

2. Proposed Methodology

This research paper aims to propose a method based on deep learning using CNN for image inpainting, styling, and colorization. This can be attained through a website that combines these models. The webpage serves as a user-friendly graphical interface. The overall block schematic is shown in figure 1.

1)Client-Side
The client-side displays the entire web application to the user side. It accesses the information available on the server. Here, the client requests the server to select/upload an image among the three features such as image styling, inpainting, and colorization. The commonly used methods in HTTP protocol are the GET request and the POST request. The GET request is used to access the data from a particular database and the POST request is used to accept the data on the client-side. These techniques are utilized to facilitate users to select the appropriate applications such as Style Transfer, Image inpainting, and Image Colorization. The request from the client to the web server is directed via the Internet.

2)*Web Server*: Requests sent by the client are received by the Web server, it stores all the information related to the request and forwards it to the respective domain name. This intercommunication is performed by using Hypertext Transfer Protocol (HTTP). All the requested pages, uploaded images, etc. are stored in the webserver database.

3)*Application Server:* The application server receives the request from the webserver and performs the task requested by the client. After performing the specified task, the image is directed to the database service for storage.

4)*Database Server*: Every form of information both in the unprocessed and processed form is fetched by the client and are stored in the database. The information can be in the form of images and login credentials. The application server fetches this data and further processes it and stores it back in the database. Lastly, the application server again retrieves the processed information from the database and sends them to the web server. In the web server, the data gets stored further. These details are again forwarded back to the client via the internet.

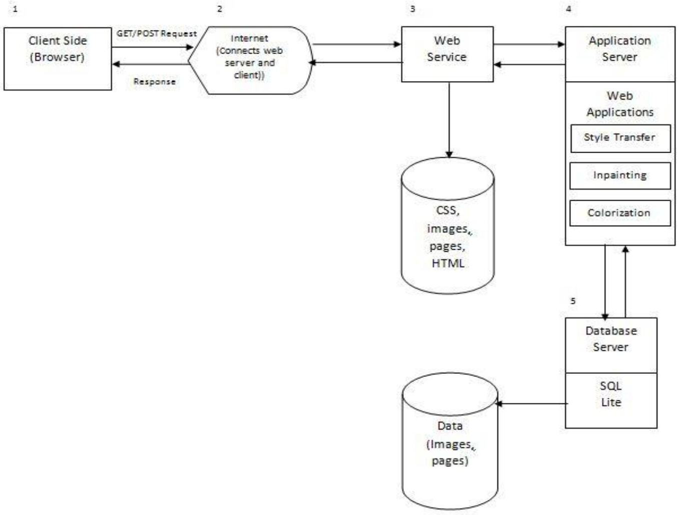

Fig - 1: Process flow of the proposed approach. This section describes the block-wise process in detail.

A. Colorization Model

Colorization of images refers to transforming black and white images to colored images, wherein the model produces colorized images that constitute semantic colors and tones from the input provided. Deep learning techniques such as Convolutional Neural Network (CNN) to obtain the desired colored images are applied. It is observed that CNNs are good at learning patterns from pictures and also map them with object classes. CNN is one of the basic architectures used for image processing based on the human brain's connectivity of neurons. The different layers of a CNN architecture consist of the Convolution layer, Pooling layer, fully connected layer. According to Zhang et.al which is similar to the vgg16 architecture, this prototype proposes it without the pooling layer and fully connected layer.

Steps:

1. The input grayscale image passed to the model is firstly resized into 224 x 224 which is necessary because there's a need to decrease the total number of pixels. Let this resized image be considered as X. Refer Fig 2 for this step.

2. The CNN architecture for colorization is different compared to the basic CNN architecture since there are no pooling layers and fully connected layers. Every block over here has 2-3 convolutional layers. The convolutional layers are then followed by RELU (Rectified Linear Unit) and accompanied by the Batch Normalization layer. In this step, all changes acquired in the resolution are obtained through either spatial down sampling or up sampling between convolutional blocks. L represents a lightness channel that encodes intensity information.

3. The resized image is then passed through the convolutional blocks and eventually through the neural networks due which the image gets transformed into $\hat{Z}$. Let the transformation be set down to G so the mathematical representation of this will be, $\hat{Z} = G(X)$. Dimensions of $\hat{Z}$are $H \times W \times Q$. Where $H$and $W$are the height and width obtained from the output of the final convolutional layer. $\hat{Z}$ consists of a vector with$Q$ values, where Q constitutes probabilities of pixels for each class for $H \times W$ pixels.

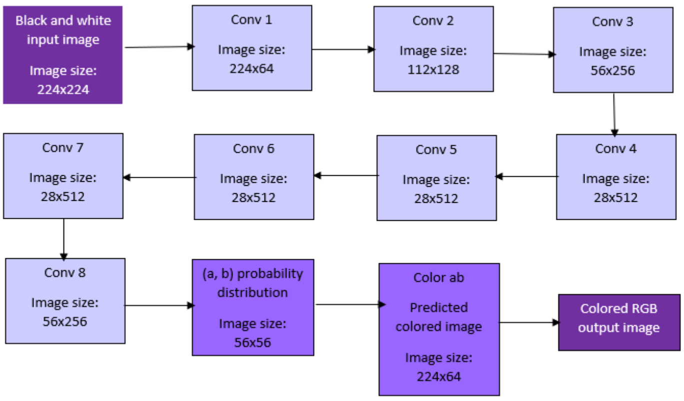

Fig - 2: CNN architecture for colorization

4. Since the final output is an RGB color image that has 3 channel color space where the information of these channels is encoded in a and b. Where a refers to the green-red component channel and $b$refers to the blue-yellow component channel. The L channel which refers to lightness and encodes intensity data only. So, the next step involves recovering the whole image from the $\hat{Z}$. Here we find the single pair of abchannel values for $\hat{Z}h, w$ which is the probability distribution. Therefore, the ab pair in correlation to the annealed mean of the probability distribution $\hat{Z}h, w$is depicted in terms of annealed mean as $\hat{Y}h, w$, where annealed mean is interposing in between the mean and mode evaluation to obtain a quantity.

5. To acquire the colored image, it is up sampled to actual image size and appended with the lightness channel L. The next step is the multinomial loss function. Here, the model transforms all the color

images into corresponding Z values in the training data processing. We invert the mapping of H,using $Z = H^{-1}(Y)$. The standard cross-entropy loss is found using,
$L(\hat{Z}, Z) = -1/HW \sum_{h,w} \sum Zx_{h,w,q} \log(\hat{Z}_{h,w,q})$

6. The final step is color rebalancing since the loss function gives dull colors. Therefore, the loss function is replaced by,
$L(\hat{Z}, Z) = -1/HW \sum_{h,w} v(Z_{k,w}) \sum_q Z_{k,w,q} \log(\hat{Z}_{h,w,q})$ to obtain vibrant colors.

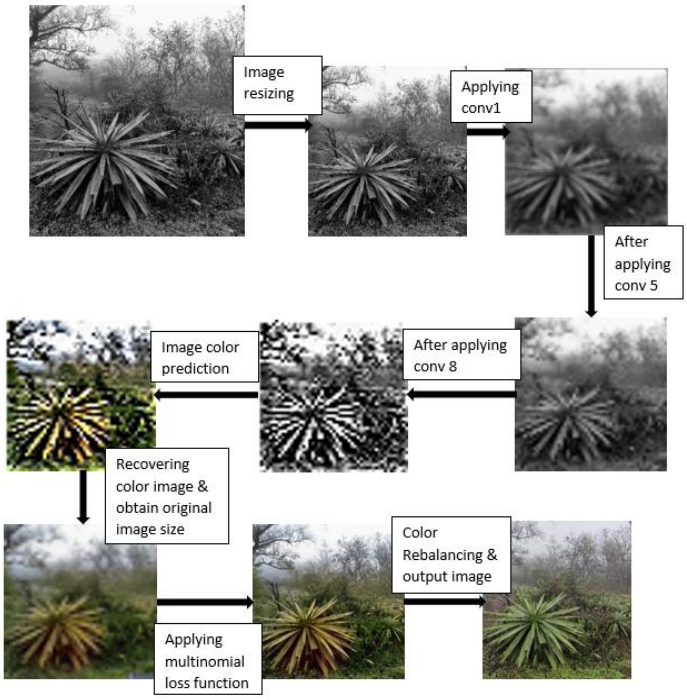
Image resizing
Applying conv1
After applying conv 5
After applying conv 8
Image color prediction
Recovering color image & obtain original image size
Applying multinomial loss function
Color Rebalancing & output image

Fig – 3: Process of Image transformation from grayscale to colorized image.

B. Neural Style Transfer

Since deep neural networks require a large dataset to train, we tried to overcome this by employing data augmentation techniques such as the addition of noise, flipping, zooming, elastic deformation, and rotation of the image. But it was found that this technique can only perform rotation and scale invariance. It is unable to achieve texture and illumination variation details, and extract complex features. In order to address these difficulties, Neural Style Transfer is used. NST provides a way to transform training images by randomizing the texture, illuminations, and contrast without losing the semantics of images.

In NST, at first, the input image is read. It is followed by defining the content and style representations. Here, From the beginning of the input layers of the network, the first few layer activations depict low-level features such as edges and textures. As we step forward in the network, the last layers depict higher-level features such as object parts like leaves or poles. The content loss of an image is computed by the following equation,

Steps:

1. Visualize the input. Construct a function to input an image whose dimensions does not exceed 512 pixels

2. Define content and style representations. With the help of in-between layers, we can fetch the content and style representations. From the beginning of the input layers of the network, the first few layer activations depict low-level features such as edges and textures. Here the last layers depict higher-level features such as object parts like leaves or poles.

3. Build the model: For defining and building a model using the functional API, state the inputs and outputs:
   model = Model (inputs, outputs)

4. Calculate Style: The whole image can be depicted by the values of in-between feature maps. The style of the image can be expressed by the means and correlations across disparate feature maps. Compute a Gram matrix that has this knowledge by taking the outer product of the feature vector with itself at distinct locations and averaging its overall positions.

5. Setting up of loss function (*Content loss):* It is a function that expresses the distance from our input image a and the content image b. It can be computed by:

$$\sum_{content}^{1}(a,b) = \sum_{x,y}^{n}(F_{xy}^{l}(b) - P_{xy}^{l}(a))^2$$

Backpropagation is performed in the established way where content loss is minimized. Thus, altering the initial image until it results in a likely response as a definite layer as the actual content image. The Style cost is given by:

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{x,y}(S_{xy}^{l} - T_{xy}^{l})^2$$

where S^{l}_{ij} and T^{l}_{ij} are the respective style representation in layer l of input image i and style image c Nl shows the number of feature maps, each of size Ml=h∗wi (where h is height and w is the width). Therefore, the overall style loss across each layer is: $L_{style}(i,c) = \sum_{l \in L} r_l E_l$

where we weigh the contribution of each layer's loss by some factor wl. In this model, weight each layer equally: $r_l = \frac{1}{||L||}$

C. Image Inpainting

Inpainting is the technique of modifying an image/picture in an unnoticeable form, there are many aims and applications of inpainting, like from the repairing of detriment paintings and photographs to the removal/replacement of selected parts of the image.

With a contextual attention layer, we deploy a unified feed-forward generative network. There are two phases to our training network. To find out the missing contents, the first phase uses a dilated convolutional network trained with reconstruction loss. The integration of contextual attention is the second phase. The model used raw image and its matched masked image as a pair for training. A 256 × 256 image with a missing section sampled randomly during training is submitted to a network, and the trained model can handle images of various sizes with many missing parts.

A two-phase rough-to-fine network design is used in the training network, with the first network making an initial rough prediction and the second network using the rough prediction as inputs to predict the better output. The reconstruction loss is explicitly learned in the rough network, while the reconstruction and GAN losses are explicitly trained in the refining network.

To generate missing spots/pixels, the contextual attention layer learns where to borrow or copy characteristic information from familiar backdrop spots/pixels. Two parallel encoders are used to integrate the contextual attention module (contextual attention layer + Diluted Convolutional layer). The first encoder employs layer-by-layer dilated convolution to focus on the contents, while the second encoder tries to pay attention to backdrop features of interest. To get inpainting result we merged the above 2 parallel encoders into a single decoder by aggregating the output feature of 2 encoders and fed them into the single decoder

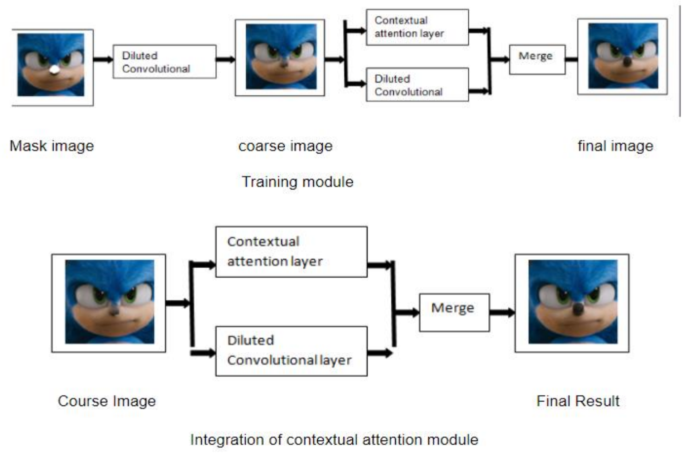

Fig – 4: Process of image transformation in image inpainting

4. Experimentation and Result

A. Neural Style Transfer

In this model for testing, input images can be of any shape, and the model will give an output as a Transferred Image of the same size as that of the input. After the generation of the transferred Image, we evaluate the Neural Style Transfer model as a method for data augmentation on the classification accuracy of the VGG network and compare its performance with traditional data augmentation methods.

Style Name	Result
None	0.8284
Flipping	0.8124
Flipping and Rotation	0.7834
Style Transfer	0.8426
Combined Traditional + Style Transfer	0.8648

Table - 1: Style Transfer Augmentation Results

Experimentation was done only using transferred Images to test classification accuracy and further combined both approaches to check any significant changes in classification accuracy. Observation said that a very remarkable increase in accuracy performance when we are using combined techniques for data augmentation.

Fig - 5: Output of Style Transferred Images

B. Image Inpainting

We evaluate the Image Inpainting model on test images having the size of 512 x 680 for consistency of testing and reported performance measure in terms of Peak signal-to-noise ratio (PSNR) and Structural Similarity Index (SSIM).

Test Input	PSNR	SSIM
Image 1	76.36 dB	0.95
Image 2	77.14 dB	0.96
Image 3	77.75 dB	0.98
Image 4	73.58 dB	0.95
Image 5	76.87 dB	0.97

Table - 2: Performance measure in terms of PSNR and SSIM on selective test Images for Image Inpainting.

Here, we got a high PSNR value for sample test images which shows that the quality of in painted Images is better. Similarly, we reported a huge similarity between the original and in painted Images.

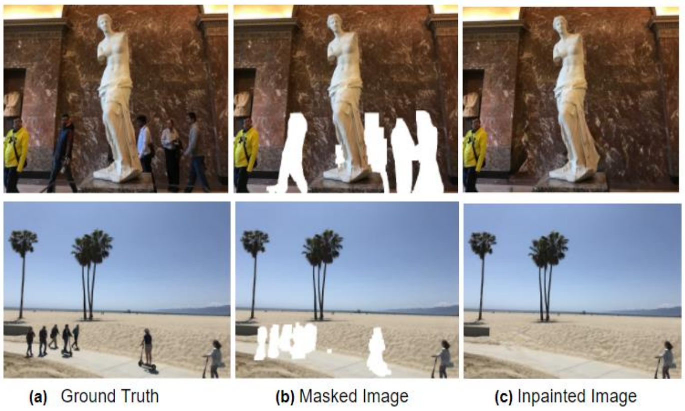
(a) Ground Truth (b) Masked Image (c) Inpainted Image

Fig - 6.: Examples of in painted Images

C. Image Colorization

We evaluate the Image Colorization model on test images of the same size to maintain test consistency and the model gives the output as a colorized image of the shape same as the shape of the input image and reported performance measure in terms of Peak signal-to-noise ratio (PSNR).

Test Input	PSNR
Image 1	71.60 dB
Image 2	70.50 dB
Image 3	70.71 dB
Image 4	70.44 dB
Image 5	71.94 dB

Table - 3: Performance measure in terms of PSNR on selective test Images for Image Colorization.

Here, a high PSNR value for sample test images which shows the quality of Colorized Images.

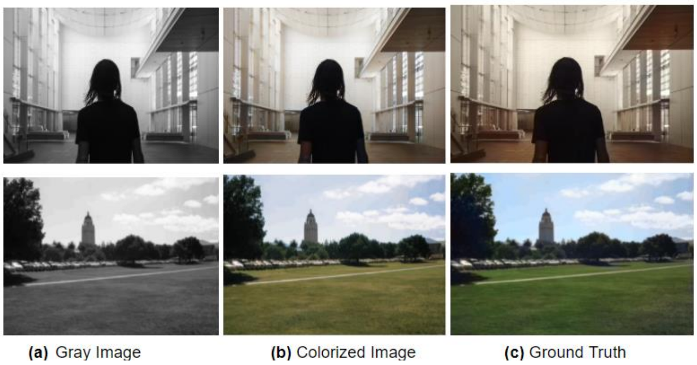
(a) Gray Image (b) Colorized Image (c) Ground Truth

Fig - 7: Examples of Colorized Images

3. CONCLUSIONS

Hence in this paper, we have proposed a method that combines all these three techniques i.e., Image colorization, Neural Style Transfer, and inpainting of images into a single web-based application. In the Neural Style Transfer approach, we presented style transfer as the image data augmentation technique and showed that this approach for data augmentation improves performance significantly. Similarly, we reported performance measures for the Image Inpainting and Colorization model by comparing the PSNR or SSIM readings of the test images. As future work, we plan to extend our website to serve and sell Stylized Images as paintings and develop a very high dimensional inpainting and colorization application.

REFERENCES

[1] R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2016, pp. 649–666.

[2] A. Patil, A. Save, V Patil, and V. Dsouza, “Coloring Greyscale Images using Deep Learning” in Proc. of the IRJET, vol. 6, Jul. 2019, pp. 801–803.

[3] B. Hariharan, P. Arbelaez, R. Girshick, and J. Malik. Hyper- ´ columns for object segmentation and fine-grained localization. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 447–456, 2015.

[4] L. A. Gatys, A. S. Ecker, and M. Bethge, “Image Style Transfer Using Convolutional Neural Networks”, pp. 2414-2423.

[5] Cheng, Z., Yang, Q., Sheng, B.: Deep colorization. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 415–423

[6] Dahl, R.: Automatic colorization. In: http://tinyclouds.org/colorize/. (2016)

[7] Charpiat, G., Hofmann, M., Sch¨olkopf, B.: Automatic image colorization via multimodal predictions. In: Computer Vision–ECCV 2008. Springer (2008) 126–139

[8] Ramanarayanan, G., Ferwerda, J., Walter, B., Bala, K.: Visual equivalence: towards a new standard for image fidelity. ACM Transactions on Graphics (TOG) 26(3) (2007) 76

[9] Jackson, Philip and Atapour-Abarghouei, Amir and Bonner, Stephen and Breckon, Toby and Obara, Boguslaw (2019) 'Style augmentation: data augmentation via style randomization.', IEEE/CVF Conference on Computer Vision and Pattern Recognition, Deep Vision Long Beach, CA, USA, 16-20 June 2019

[10] Xu Zheng1,2, Tejo Chalasani2, Koustav Ghosal2, Sebastian Lutz2 and Aljosa Smolic, “STaDA: Style Transfer as Data Augmentation” (2019)

[11] X. Bouthillier, K. Konda, P. Vincent, and R. Memisevic. Dropout as data augmentation. arXiv:1506.08700, 2015.

[12] T. Q. Chen and M. Schmidt. Fast patch-based style transfer of arbitrary style. In Workshop in Constructive Machine Learning, 2016.

[13] D. C. Ciresan, U. Meier, L. M. Gambardella, and J. Schmidhuber. Deep big simple neural nets excel on digit recognition. Neural Computation, 22(12):3207–3220, 2010.

[14] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In Int. Conf. Artificial Intelligence and Statistics, pages 215–223, 2011.

[15] Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In Proceedings of COMPSTAT’2010, pages 177–186. Springer.

[16] Po-Yu Liu, Prof. Edmund Y. Lam, Image Reconstruction Using Deep Learning, arXiv:1809.10410 , 2018

[17] Yazen a. Khalil& peshawa j. Muhammad ali, A proposed method for colorizing grayscale images, International Journal of Computer Science and Engineering (IJCSE), ISSN 2278-9960, Vol. 2, Issue 2, May 2013, 109-114

[18] Aaron van den Oord, Nal Kalchbrenner, Koray Kavukcuoglu, Pixel Recurrent Neural Networks, arXiv:1601.06759**, 2016**

[19] Caruana, R., Lawrence, S., and Giles, C. L. (2001). Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. In Advances in neural information processing systems, pages 402–408.

[20] Chalasani, T., Ondrej, J., and Smolic, A. (2018). Egocentric gesture recognition for head mounted ar devices. In Adjunct Proceedings of the IEEE International Symposium for Mixed and Augmented Reality 2018 (To appear).

[21] Chen, Y.-L. and Hsu, C.-T. (2016). Towards deep style transfer: A content-aware perspective. In BMVC.

[22] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and Locally Consistent Image Completion. ACM Trans. Graph. 36, 4, Article 107 (July 2017), 14 pages. DOI: http://dx.doi.org/10.1145/3072959.3073659

[23] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, Thomas S. Huang, “Generative Image Inpainting with Contextual Attention’, 2018

[24] Guilin Liu, Fitsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, Bryan Catanzaro, “Image Inpainting for Irregular Holes Using Partial Convolutions”, 2018

[25] Connelly Barnes, Eli Shechtman, Dan B. Goldman, and Adam Finkelstein. 2010. The Generalized Patchmatch Correspondence Algorithm. In European Conference on Computer Vision. 29–43.

[26] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. 2000. Image Inpainting. In ACM Transactions on Graphics (Proceedings of SIGGRAPH). 417–424.

[27] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. 2003. Simultaneous structure and texture image inpainting. IEEE Transactions on Image Processing 12, 8 (2003), 882–889.

[28] A. Criminisi, P. Perez, and K. Toyama. 2004. Region Filling and Object Removal by Exemplar-based Image Inpainting. IEEE Transactions on Image Processing 13, 9 (2004), 1200–1212.

[29] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. 2012. Image Melding: Combining Inconsistent Images using Patch-based Synthesis. ACM Transactions on Graphics (Proceedings of SIGGRAPH) 31, 4, Article 82 (2012), 82:1–82:10 pages