Neural Style Transfer

Stefan Wezel

Abstract: *Introduced by Gatys* et al. *(2015), the field of neural style transfer (NST) has not only evolved rapidly but also allowed for insights into the processes inside neural networks as well as into human perception. Various methods, ranging from image-based to model-based approaches have been introduced to alleviate the weaknesses of the original formulation. Here, we recapture the idea behind Gatys* et al. *(2015)'s algorithm and give an overview of methods, used in the current field of neural style transfer.*

Contents

1 Introduction

Art has played an important role in human culture throughout most of its history [Carroll (2004)]. Despite this, little is known about what the deciding neurological and psychological factors are for making us perceive something as aesthetic. In the past, the relationship between neurological science, psychology, and artificial intelligence has proven to be reciprocally beneficial. Especially the field of machine learning has yielded astonishing results, partially inspired by neurological foundations. On computer vision tasks, Convolutional Neural Nets have already outperformed humans, thus giving the impression that they may almost rival the perceptive prowess of the human visual cortex [Geirhos *et al.* (2017)].
However, in Style Transfer, traditionally a subfield of non-photorealistic rendering, Neural Networks have yet to prove their capacity. Gatys *et al.* (2015)'s work shows that they can be used to transfer arbitrary styles to any given content image. Besides the visually astonishing results, their work also gives insights into the creation and perception of artistic images.

2 Non-neural Style Transfer

Before Neural Networks were applied to Style Transfer, a popular approach to the problem existed in image-based artistic rendering [Kyprianidis *et al.* (2012)]. This area of research can be subdivided into multiple directions. **Linear Transformations**, i.e. Filters designed for image processing are used to create stylized images by Winnemöller *et al.* (2006) and Tomasi and Manduchi (1998) among others.
Another popular approach, **Stroke-Based Rendering** typically starts with a photograph on which then strokes are placed iteratively in order to mimic a certain style. The placement of the strokes is optimized according to a given objective function which is some quantity that measures how similar the synthetic painting is to a specified image [Hertzmann (2003)]. Another technique is **Region-based Rendering** where an image is segmented into different regions. This enables a rendering algorithm to be sensitive to each region's specific content [Kolliopoulos (2005)]. Both these approaches do lack the ability to incorporate any arbitrary style. Therefore, Hertzmann *et al.* (2001) proposes to learn a transformation from source to target image in a supervised fashion with the **Example-Based Rendering** technique of Image Analogies. This, however, requires training data which may not be available.
While not having been designed for the goal of creating an artistic image, important contributions to style transfer research also come from **Texture Synthesis**. Early work there focused on pixel measurements [Julesz (1962)]. Later, filter responses played an important role in the work of Heeger and Bergen (1995) and Portilla and Simoncelli (2000). The use of summary statistics in Texture Synthesis can be viewed as a precursor to the

neural style transfer algorithm proposed by Gatys *et al.* (2015). There, however, not the statistics of an image, but rather the statistics of the latent representation of an image are used for measuring similarity in style.
Rather than using Descriptive Statistics, another branch of research exploits Markov Random Fields (MRF) in a **non-parametric modeling** approach to rendering stylized images. An MRF assumes that each pixel is characterized solely by the pixels in its spatial neighborhood [Jing *et al.* (2019)]. Efros and Leung (1999) find pixels in the texture image by finding pixels whose neighborhoods resemble each other and then replace them in the source image.
The topic of neural style transfer can also be linked to **Image Reconstruction** where, rather than encoding images into latent representation, the goal is to create an image from given information. An approach, proposed by Mahendran and Vedaldi (2015) is able to generate images by optimizing latent representations. Given random noise, the algorithm iteratively optimizes the image up to a point where its latent representation matches that of those, generated by a convolutional net. This is computationally expensive because a new training process is required for every image. In order to generate images faster, Dosovitskiy and Brox (2016) propose to train a generative model to produce an image. Then, after a training stage, images can be computed in real-time, thus shifting the computational effort.

3 Original Algorithm of Neural Style Transfer

Using summarizing statistics from the feature representation of CNNs, Gatys *et al.* (2015) propose an image-based approach to Style Transfer.

3.1 Setting

Given two images, one responsible for the content, and one for style, the goal of neural style transfer is to create a stylized image that matches the content images content and the style images style. The stylized image is iteratively optimized according to two loss terms.

3.2 VGG-Net

Convolutional Neural Networks have proven to represent features of input images efficiently once trained for a task like image classification. An architecture that became popular after winning the ILSVRC localization task in 2014 is the VGG-net [Russakovsky *et al.* (2015)]. The VGG-net architecture (1) is characterized by stacking multiple convolutional layers, following a pooling operation. This is repeated multiple times up to one or more fully connected layers, which can serve as task-specific head Simonyan and

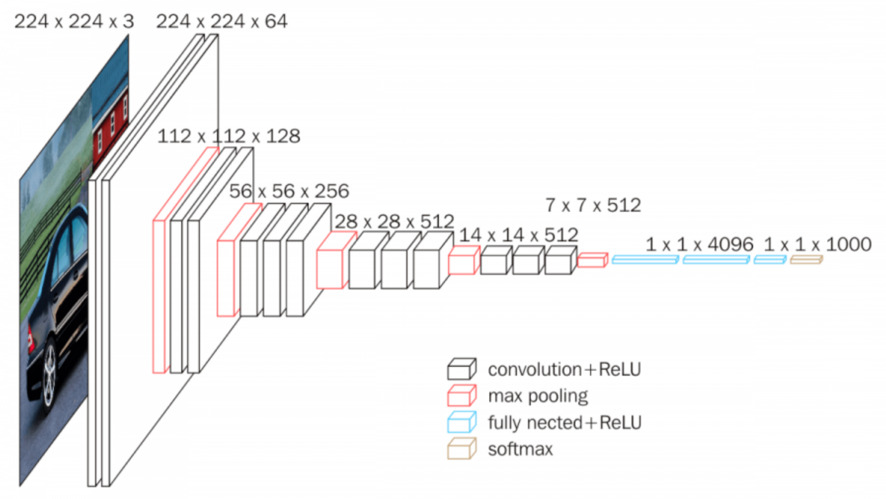

Figure 1: Architecture of the VGG net introduced by Simonyan and Zisserman (2014). Gatys *et al.* (2015) extract feature representations from different convolutional layers of such a model.

Zisserman (2014).
The latent representations from different convolutional layers are extracted by Gatys *et al.* (2015) and serve as the basis for obtaining summarizing statistics. These statistics are used to transform the input image.

3.3 Algorithm

The algorithm starts by extracting a content representation from the content image and style representation of the style image using the VGG-net. From the image, we want to optimize, we extract content as well as style. Then, the distance between the style representations and the content representations is measured. Each distance is a respective loss which gives us two quantities we can minimize. These two loss terms, style loss, and content loss are each multiplied with a small scalar and then added up, giving us our final loss term. Then, the gradient of the loss function for the current state of the input image is computed and backpropagated. However, only the input image is adjusted according to the gradient and none of the parameters of the VGG-net.

3.4 Content Representation and Loss

A convolutional layer l with n_l filters returns n_l feature maps which mark the positions of where each respective feature occurred. The height and width of these feature maps are determined by the input and stride used [LeCun *et al.* (1999)]. To extract content information, Gatys *et al.* (2015) propose to gather all feature maps of a layer in a feature Matrix

$F_l \in \mathcal{R}^{n_l \times m_l}$ where m_l equals the height times the width of the layers feature maps. Each feature map is vectorized and transposed, resulting in a feature vector which then becomes a column of the feature matrix. The level of abstraction of features in F_l is sensitive to the choice of l. Gatys *et al.* (2015) propose to use a layer that captures the shapes and layout of the input image rather than finer features such as texture information.

To measure the difference between an input image x and content image p, Gatys *et al.* (2015) propose to generate respective feature matrices F^l and P^l and measure mean square distance between these:

$$\mathcal{L}_{content}(p, x, l) = \frac{1}{2} \sum_{ij} (F^l_{ij} - P^l_{ij})^2 \tag{1}$$

3.5 Style Representation and Loss

The style of an image is invariant to the spatial arrangement of its features. Thus, the style representation should be spatially invariant as well. Since feature maps contain spatial information and therefore cannot be used to represent style, Gatys *et al.* (2015) propose to measure their co-occurrence in an image. For this purpose, they make use of the Gram matrix.

The Gram matrix of a set of vectors contains the inner product of all combinations of vectors from the set. In the case of neural style transfer, the set of vectors is the set of vectorized feature maps of a layer l. By calculating the inner product of each of the feature vectors, all spatial information is removed. Remaining is the information about how often the features appear in the same position. This gives us a spatially invariant representation of style by capturing what features correlate rather than where a feature appears.

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk} \tag{2}$$

Gatys *et al.* (2015) found that using multiple layers for the style representation produces visually more appealing results. So in order to measure similarity in style of input image x and style image a, the respective Gram matrices G^l and A^l are computed. The loss term is

$$L_{style}(a, x) = \sum_{l=0}^{L} w_l E_l$$

where

$$E_l = \frac{1}{4 n_l^2 m_l^2} \sum_{ij} (G^l_{ij} - A^l_{ij})^2$$

and w_l weights each layers contribution to the style loss.

3.6 Total Loss and Optimization

The total loss term is composed of the style loss and the content loss, each weighted by a factor.

$$\mathcal{L}_{total}(p, a, x) = \alpha \mathcal{L}_{content}(p, x) + \beta \mathcal{L}_{style}(a, x)$$

where α and β are hyperparameters that determine the style/content trade-off.
In order to compute gradients that can be backpropagated to the input image, style, and content loss need to be derived according to the layer's activation. For content loss, we can compute the derivative with

$$\frac{\partial \mathcal{L}_{content}}{\partial F^l_{ij}} = \begin{cases} (F^l - P^l)_{ij} & \text{if } F^l_{ij} > 0 \\ 0 & \text{if} F^l_{ij} < 0 \end{cases}$$

where as style loss can be differentiated as follows:

$$\frac{\partial E_l}{\partial F^l_{ij}} = \frac{1}{n_l^2 m_l^2}((F^l)^T(G^l - A^l))_{ij} \text{ if } F^l_{ij} > 0$$
$$0 \text{ if} F^l_{ij} < 0$$

4 Derivations and Alternative Approaches

Despite the field of neural style transfer being rather young, many alternatives, derivations, and improvements to Gatys *et al.* (2015) algorithm have been proposed. Generally, we can differ between two approaches. Image-based approaches iteratively optimize an input image to match the source image's content and style. This, however, requires a new training process for every new image. The many resulting backward passes are computationally expensive. Instead of adjusting an input image, **Model-based approaches** optimize a generative model that creates stylized images. Thus shifting the computational cost towards the training stage and making a real-time generation of stylized images possible. In only a few years, many ideas for both approaches have been proposed. In the following section, we will have a look at some of those ideas and briefly discuss their benefits and drawbacks.

4.1 Image-Based

Simonyan *et al.* (2013)'s algorithm for interpreting features learned by a convolutional neural net can be viewed as a precursor to neural style transfer. They optimize the input image to maximally activate a specified subset, i.e. a layer, feature map, or single neuron of a convolutional neural net. The previously discussed algorithm of Gatys *et al.* (2015)

then opened the field of neural style transfer with their image-based approach. The proposed optimization process, however, suffers from instability during training which often results in noisy stylized images. Risser *et al.* (2017) found that this is due to feature maps with different means and variances can still have a similar Gram matrix. To alleviate this, they calculate the distribution of feature activations of style image and input image representations and then compare the difference of the resulting distribution's histogram. This difference is then added as a third loss term. Besides the presented approaches, there exist many more and this is only a small subset of ideas developed in the last few years.

4.2 Model-Based

The field of model-based approaches can be subdivided into three groups which differ in their setting. In the **Single-style-per-model** setting, a generative model is trained to match one specific style. This could for example be achieved by creating a large dataset of input content and stylized target images with an image-based Algorithm and then train the generative model using on this dataset. Most of these approaches only differ in the network architecture used for the generative model [Jing *et al.* (2019)]. One such example, proposed by Johnson *et al.* (2016) is depicted in 3.
A more challenging setting is **Multiple-style-per-model** where one model is ideally able to create images in one of multiple pre-defined styles. A common idea for this is, to try and make only a subset of parameters responsible for one style and ideally be able to share parameters. One such approach was proposed by Chen *et al.* (2017), whose model learns representations for style and content individually. They use an encoder to learn content representations, a 'StyleBank' layer which contains specific filters for each style that can be chosen and a decoder that produces the stylized image, using the chosen filters from the 'StyleBank' layer. Images created by this approach are depicted in 4.
The category of model-based which matches the flexibility of image-based approaches the closest is **Arbitrary-style-per-model**. This setting is difficult and various, often quite different approaches exist. Chen and Schmidt (2016) find for each patch of feature representations from the content image the most similar patch in the representation of the style image and then replace it. The idea is to find representations that are responsible for style in the content image and then swap them with style information from the style image. Results from Chen and Schmidt (2016) are shown in 5.

5 Challenges and Future Directions

Despite the rapid progress of the only recently introduced field of neural style transfer, there are still many open challenges that have yet to be overcome.

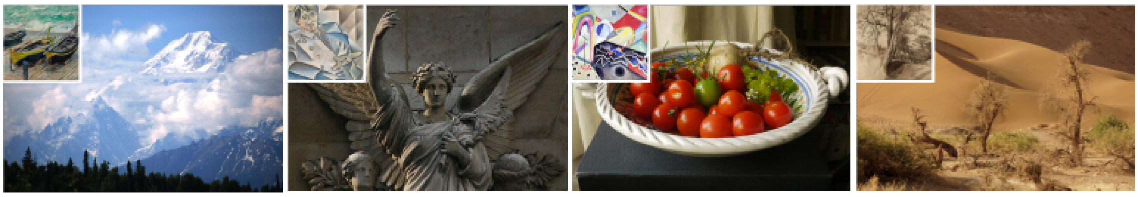

Figure 2: Content and style images.

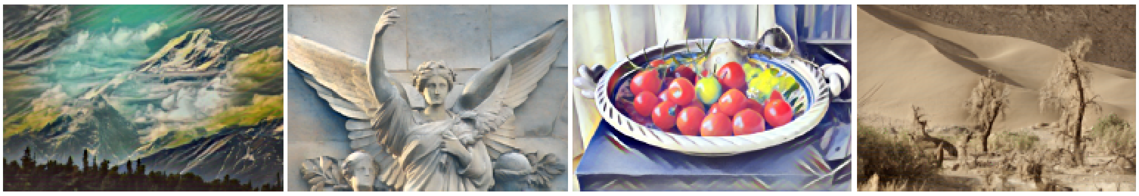

Figure 3: Johnson *et al.* (2016)

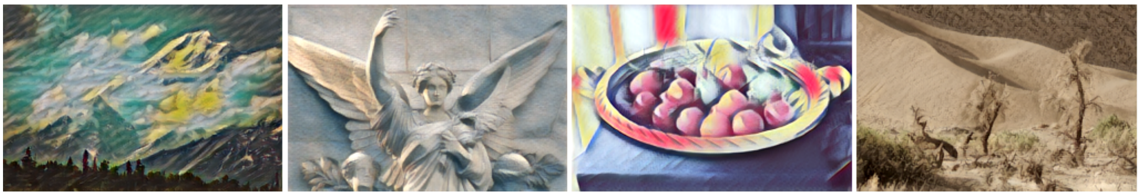

Figure 4: Chen *et al.* (2017)

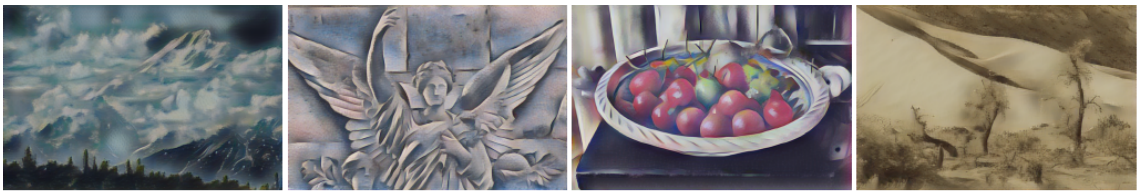

Figure 5: Chen and Schmidt (2016)

5.1 Evaluation

One key challenge is the evaluation of different models and algorithms. Since the result of neural style transfer is inherently subjective, coming up with a unified and agreed upon evaluation protocol is a difficult task. In recent publications, researchers often evaluated their novel techniques by showing side-by-side comparisons of their work next to existing techniques. Jing *et al.* (2019) propose to conduct a study with eight individuals that rate the results of different NST approaches. However, their ratings varied a lot, despite most of their subjects were of the same age and background. Also, such an approach is expensive as well as difficult to organize and replicate.

Jing *et al.* (2019) also propose to involve professionals, such as artists and designers into an evaluation process to measure performance according to agreed-upon aesthetic principles.

They also suggest that the NST research community would benefit from a benchmark dataset. Such benchmarks datasets have in the past caused large leaps in different areas of machine learning. A solution for this could be to use benchmark datasets from related areas, such as Non-photorealistic rendering. Meanwhile, NST is not only limited to image data, but is now also applied in video and 3D data [Huang *et al.* (2017), Kato *et al.* (2018)], so it would be necessary to agree on benchmark datasets for those categories as well.

5.2 Interpretability

Another important issue is the interpretability of models and algorithms used for neural style transfer. All of the covered approaches use black-box models like convolutional neural networks. Generally, it is not trivial to understand what features they learn and use for producing an output. In the case of an undesired or unsatisfying result, it is then difficult to see how changes made to the model or training process would affect the outcome. This then often results in a trial-and-error process.

Using models that learn **disentangled representations** could to some extend alleviate this problem. The definition of disentanglement in the context of machine learning is not agreed upon and the subject of debate [Higgins *et al.* (2018)]. Usually, it is assumed for a model that learns disentangled representation to encode the independent generating factors of data as separate dimensions in a latent representation thus creating interpretable representations that would ideally resemble real-world observable variables. This would not only make neural style transfer more interpretable but also more controllable and therefore more applicable because it could allow for tools, where different aspects of a certain style, such as stroke direction, brush size, lighting among many others could be controlled by a user.

However, the field of disentangled representation learning is challenging in itself. The mere possibility of unsupervised learning of disentangled representations is debated [Locatello *et al.* (2018)]. On the other side, learning disentangled representation in a supervised fashion may be unreasonable since it would require large datasets that would need to be variant to the many different aspects of a certain style.

6 Applications

To be applied in a broader set of real-world settings, neural style transfer has yet to overcome many of the discussed challenges.

An area where NST is already applied is **social media**. Apps like Prisma [Prisma-Labs (2020)] have made NST a popular way to create and share stylized photographs. By making NST available to a large number of users, Jing *et al.* (2019) hypothesis that NST research could use the perception of different stylized images in order to move towards a more quantitative evaluation. Mobile devices play a large role in social media but have

only limited computational resources. For more widespread use of NST on such edge devices, the developing community would have to come up with computationally less expensive methods.
An even more challenging setting for applied NST is that of **User-assisted Creation Tools**. For example, design tools, that help users integrate a certain style into their work could be able to speed up creation processes.
Such tools could be used in **Production of Entertainment Products**. Use of Computer Generated Imagery (CGI) has increased ever since its inception [Tucker (2007)]. Integrating certain styles into CGI is an elaborate process that requires programming shaders and the work of artists [Apodaca *et al.* (2000)]. Tools that could render given images in a specified style could alleviate some of the technical difficulties of this process and speed up production as well as make it more flexible since ideally styles could be manipulated after animating sequences and would be invariant to the underlying imagery.
Besides those discussed, it is very likely that there are many more approaches to which some of them may only emerge, once the field of NST itself has progressed even further.

7 Conclusion

The field of neural style transfer is, despite its very recent emergence interesting and fast developing. It motivates us to further engage with the processes inside deep neural networks. Gatys *et al.* (2015) hypothesize that it can even help us understand the underlying principles of what makes something aesthetically appealing to humans. That Convolutional Neural Networks excel in such an area further proves their power and versatility. It also shows that meaningful and good representations can be useful for downstream tasks, even ones previously not thought about. However, many open challenges remain as the field of neural style transfer matures.

References

Apodaca, A. A., Gritz, L., and Barzel, R. (2000). *Advanced RenderMan: Creating CGI for motion pictures*. Morgan Kaufmann.

Carroll, N. (2004). Art and human nature. *The journal of aesthetics and art criticism*, **62**(2), 95–107.

Chen, D., Yuan, L., Liao, J., Yu, N., and Hua, G. (2017). Stylebank: An explicit representation for neural image style transfer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1897–1906.

Chen, T. Q. and Schmidt, M. (2016). Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337*.

Dosovitskiy, A. and Brox, T. (2016). Generating images with perceptual similarity metrics based on deep networks. In *Advances in neural information processing systems*, pages 658–666.

Efros, A. A. and Leung, T. K. (1999). Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1033–1038. IEEE.

Gatys, L. A., Ecker, A. S., and Bethge, M. (2015). A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*.

Geirhos, R., Janssen, D. H., Schütt, H. H., Rauber, J., Bethge, M., and Wichmann, F. A. (2017). Comparing deep neural networks against humans: object recognition when the signal gets weaker. *arXiv preprint arXiv:1706.06969*.

Heeger, D. J. and Bergen, J. R. (1995). Pyramid-based texture analysis/synthesis. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 229–238.

Hertzmann, A. (2003). A survey of stroke-based rendering. *IEEE Computer Graphics and Applications*, (4), 70–81.

Hertzmann, A., Jacobs, C. E., Oliver, N., Curless, B., and Salesin, D. H. (2001). Image analogies. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 327–340.

Higgins, I., Amos, D., Pfau, D., Racaniere, S., Matthey, L., Rezende, D., and Lerchner, A. (2018). Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*.

Huang, H., Wang, H., Luo, W., Ma, L., Jiang, W., Zhu, X., Li, Z., and Liu, W. (2017). Real-time neural style transfer for videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 783–791.

Jing, Y., Yang, Y., Feng, Z., Ye, J., Yu, Y., and Song, M. (2019). Neural style transfer: A review. *IEEE transactions on visualization and computer graphics*.

Johnson, J., Alahi, A., and Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer.

Julesz, B. (1962). Visual pattern discrimination. *IRE transactions on Information Theory*, **8**(2), 84–92.

Kato, H., Ushiku, Y., and Harada, T. (2018). Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916.

Kolliopoulos, A. (2005). *Image segmentation for stylized non-photorealistic rendering and animation*. University of Toronto.

Kyprianidis, J. E., Collomosse, J., Wang, T., and Isenberg, T. (2012). State of the" art?: A taxonomy of artistic stylization techniques for images and video. *IEEE transactions on visualization and computer graphics*, **19**(5), 866–885.

LeCun, Y., Haffner, P., Bottou, L., and Bengio, Y. (1999). Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer.

Locatello, F., Bauer, S., Lucic, M., Raetsch, G., Gelly, S., Schölkopf, B., and Bachem, O. (2018). Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*.

Mahendran, A. and Vedaldi, A. (2015). Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196.

Portilla, J. and Simoncelli, E. P. (2000). A parametric texture model based on joint statistics of complex wavelet coefficients. *International journal of computer vision*, **40**(1), 49–70.

Prisma-Labs (2016 (accessed June 22nd, 2020)). *Prisma: Turn Memories into art*.

Risser, E., Wilmot, P., and Barnes, C. (2017). Stable and controllable neural texture synthesis and style transfer using histogram losses. *arXiv preprint arXiv:1701.08893*.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, **115**(3), 211–252.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Simonyan, K., Vedaldi, A., and Zisserman, A. (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Sixth international conference on computer vision (IEEE Cat. No. 98CH36271)*, pages 839–846. IEEE.

Tucker, H. (2007). At the movies. *ITNow*, **49**(5), 8–9.

Winnemöller, H., Olsen, S. C., and Gooch, B. (2006). Real-time video abstraction. *ACM Transactions On Graphics (TOG)*, **25**(3), 1221–1226.