Can depthwise separable convolution make neural style transfer more lightweight? A Comparative Study

A Preprint

Shichao Guo
Department of Computer Science
Aarhus University
au779770@uni.au.dk

December 1, 2025

Abstract

This project investigates the efficacy of Depthwise Separable Convolutions (DSC) in reducing the computational cost of Neural Style Transfer. We implemented a baseline Fast Neural Style Transfer network and three progressive lightweight variants. Our experiments on the COCO 2017 dataset demonstrate that replacing standard convolutions with DSCs can reduce the model size by up to 87.62% and achieve a 1.44x speedup on CPU inference, while maintaining comparable visual quality. This study highlights the potential of architectural optimizations for deploying generative models on edge devices.

Keywords Neural Style Transfer · Depthwise Separable Convolutions

1 Introduction

Neural Style Transfer (NST), first introduced by Gatys et al. [Gatys et al., 2016], has become a popular application of deep learning, allowing users to blend the content of one image with the artistic style of another. While the original optimization-based approach was slow, Johnson et al. [Johnson et al., 2016] proposed a feed-forward network (Transformer Net) to generate stylized images in real-time. However, these networks often rely on heavy standard convolution layers, making them computationally expensive for real-time applications on mobile or edge devices.

The objective of this study is to determine if Depthwise Separable Convolutions—a technique popularized by MobileNet [Howard et al., 2017] for efficient computing—can be applied to the style transfer domain to create a "lightweight" generator without significantly compromising artistic quality. We conduct a comparative study (Ablation Study) across three different levels of architectural modification to answer the question: Can we make style transfer lighter without losing the "style"?

2 Related Work

Neural Style Transfer: Gatys et al. [Gatys et al., 2016] demonstrated that deep features from Convolutional Neural Networks (CNNs) can separate and recombine image content and style. Johnson et al. [Johnson et al., 2016] improved upon this by training a feed-forward network to approximate the optimization process, enabling real-time style transfer.

Efficient Deep Learning: To run deep models on mobile devices, reducing computational cost is crucial. Howard et al. [Howard et al., 2017] introduced MobileNets, which utilize Depthwise Separable Convolutions (DSC) to factorize a standard convolution into a depthwise spatial convolution and a pointwise (1×1) channel convolution. This factorization significantly reduces both parameters and computation (FLOPs).

Our work bridges these two fields by integrating DSCs into the architecture of Johnson et al., evaluating the trade-offs between efficiency and visual fidelity.

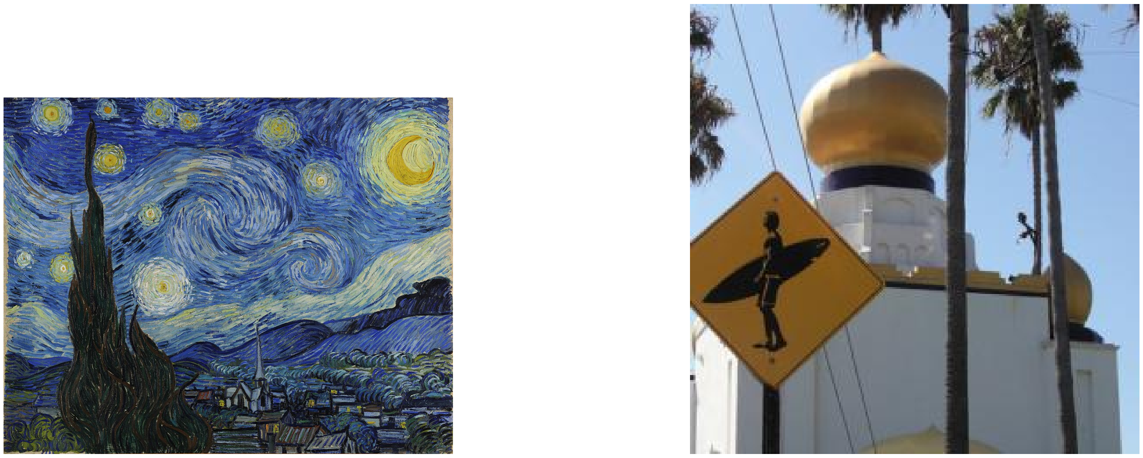
3 Methods

3.1 Dataset and Preprocessing

We utilized the **COCO 2017 Validation Set** as our content image source. Due to computational constraints, we used a subset of the data:

- **Training Set:** 4,455 images
- **Validation Set:** 495 images (used for monitoring loss)
- **Test Set:** 50 images (reserved for final qualitative evaluation)

All images were resized to 256×256 pixels, center-cropped, and normalized. The style reference image used was Vincent van Gogh's *The Starry Night* (See Figure 1).



(a) Style Image (Starry Night)

(b) Example Content Image

Figure 1: Inputs for the Style Transfer task.

3.2 Model Architectures

We designed four model variants to isolate the impact of the lightweight layers.

Baseline: A standard Transformer Net consisting of 3 convolution layers (Encoder), 5 Residual Blocks, and 3 Transposed Convolution layers (Decoder).

Lightweight Layer Design: We implemented a *LightweightConvLayer* that replaces the standard spatial convolution with a Depthwise Convolution (groups=C_{in}) followed by a Pointwise Convolution (1×1). Instance Normalization and ReLU activation were injected between the depthwise and pointwise stages to preserve non-linearity.

The Variants:

- **Lightweight-v1:** Replaces only the 5 **Residual Blocks** with lightweight blocks.
- **Lightweight-v2:** Replaces the **Encoder** (initial downsampling layers) and **Residual Blocks**.
- **Lightweight-v3 (Fully Lightweight):** Replaces the **Encoder**, **Residual Blocks**, and the **Decoder** (upsampling layers).

3.3 Training

The networks were trained to minimize a Perceptual Loss function ($L_{total} = \lambda_c L_{content} + \lambda_s L_{style} + \lambda_{tv} L_{tv}$), computed using a pre-trained VGG-16 network.

- **Optimization:** Adam optimizer ($lr = 1e - 3$) with `ReduceLROnPlateau` scheduler.
- **Training Duration:** 20 Epochs.
- **Hardware:** Training was performed on a GPU (A100), while inference benchmarking was conducted on a CPU to simulate edge constraints.

4 Results

4.1 Quantitative Analysis: Efficiency

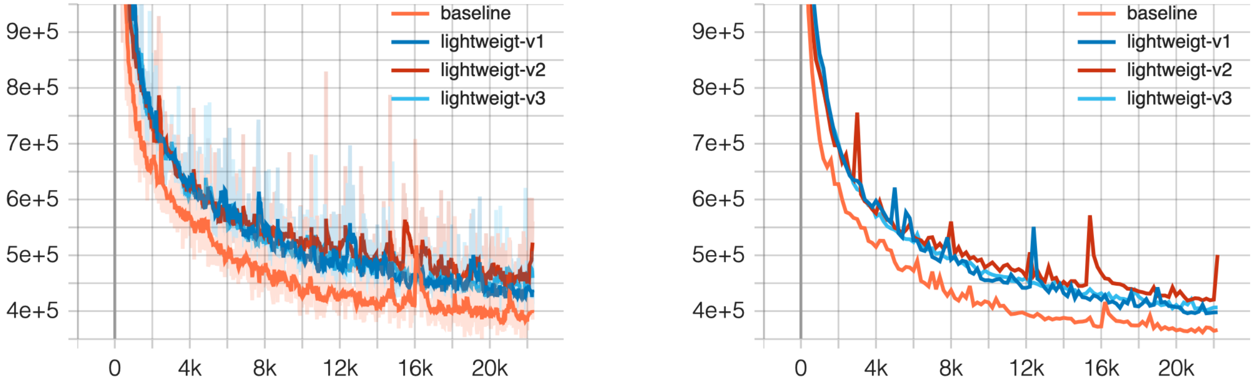
We evaluated the models based on Parameter Count (Space complexity) and Inference Latency (Time complexity). As shown in Table 1, the **Lightweight-v3** model achieved a massive **87.6% reduction in size**, shrinking the model from $\sim$1.68 million parameters to just $\sim$0.2 million. This translated to a consistent speedup, reducing inference time per image from $\sim$71ms to $\sim$49ms.

Table 1: Model Efficiency Comparison

Model Variant	Parameters	Reduction (%)	Avg Latency (CPU)	Speedup
Baseline	1,679,235	-	71.31 ms	-
Lightweight-v1	381,315	77.29%	60.46 ms	1.18x
Lightweight-v2	292,796	82.56%	57.65 ms	1.24x
Lightweight-v3	207,865	87.62%	49.49 ms	1.44x

4.2 Training Dynamics

We monitored the training process by tracking the total loss (content + style + variation) on both training and validation sets. As shown in Figure 2, all models demonstrated a consistent decrease in loss, indicating successful convergence without overfitting. The lightweight models (v1, v2, v3) showed a slightly slower convergence rate compared to the baseline, consistent with their reduced capacity.



(a) Training Loss

(b) Validation Loss

Figure 2: Loss curves showing consistent convergence for all models.

We also tracked the learning rate (Figure 3). The `ReduceLROnPlateau` scheduler reduced the learning rate when the validation loss plateaued, allowing for fine-grained optimization in later epochs.

4.3 Qualitative Analysis: Visual Quality

We compared the output images generated by the models on the test set.

Early Training (Step 800): As observed in Figure 4, the Baseline model learned the style features much faster than the lightweight variants. At step 800, the Baseline output is already recognizable, while v1, v2, and v3 are still struggling with basic color distribution. This confirms that the lightweight models have a "Capacity Gap" and converge slower.

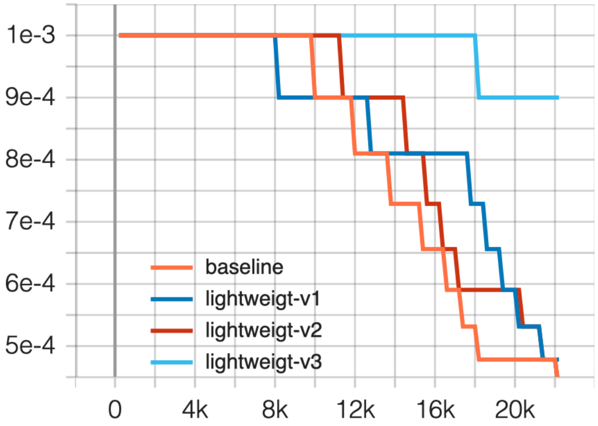
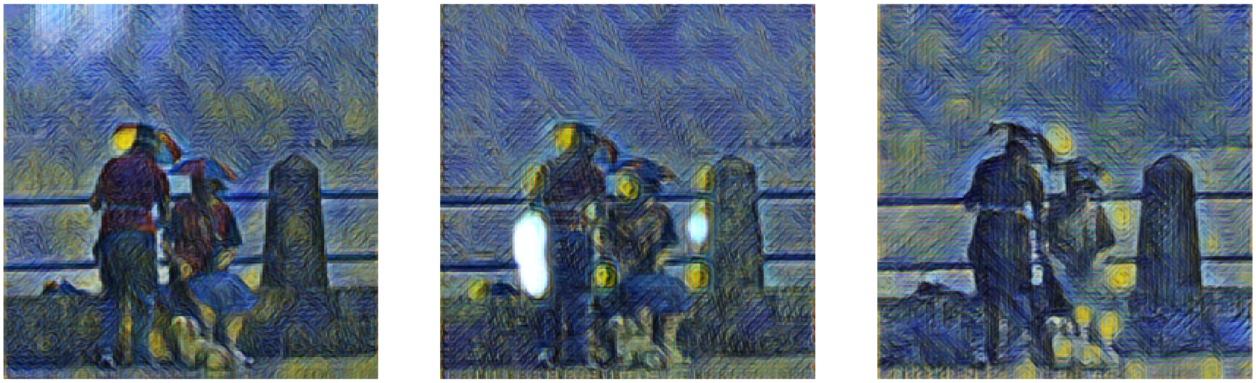


Figure 3: Learning Rate Schedule



(a) Baseline (Step 800) (b) Lightweight-v1 (Step 800) (c) Lightweight-v3 (Step 800)

Figure 4: Early training comparison. The Baseline converges significantly faster.

Final Result: Despite the slower start, all models converged successfully by the end of training. Figure 5 shows the final outputs. Visually, the lightweight models (v1, v2, v3) are almost indistinguishable from the baseline in terms of global structure and color. The residual bottleneck appears to be highly redundant, as removing it (v1) caused little to no visual degradation.

4.4 Specific Artifacts

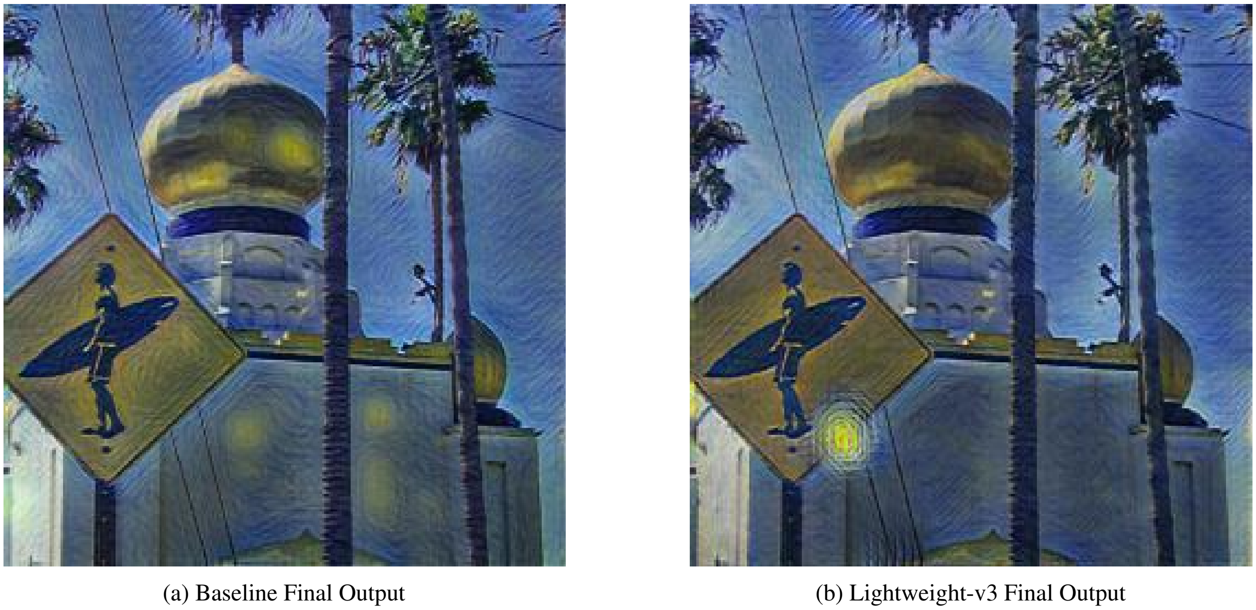
A specific artifact was observed in the lightweight models: array-like red spots appeared within the yellow celestial orbs (see Figure 6), which were absent in the baseline. This reduces the subjective quality slightly in high-contrast regions.

5 Discussion

Capacity Gap and Convergence: The lightweight models exhibited a slower convergence rate compared to the baseline. This is attributed to their reduced parameter capacity. However, the consistent decrease in both training and validation losses indicates that the lightweight architectures were successfully learning the style transfer task without suffering from overfitting.

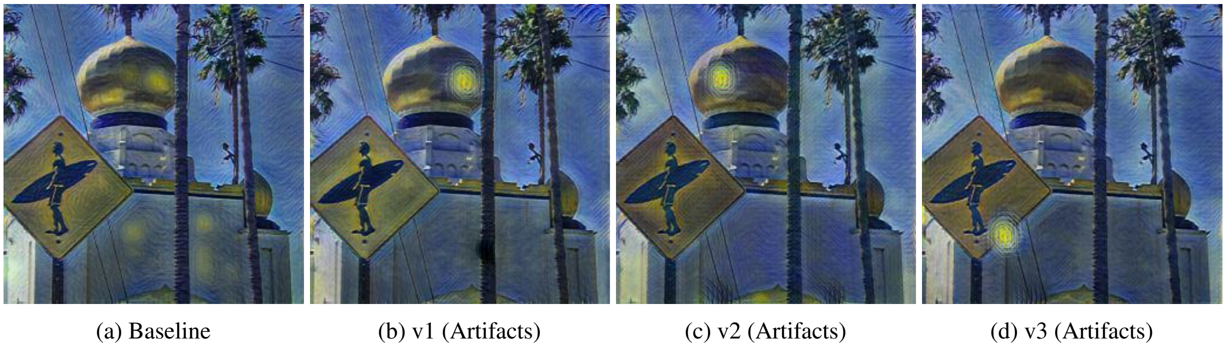
Artifacts from Decoupling: We hypothesize that the "red spot" artifacts are a side effect of Depthwise Separable Convolutions. By decoupling spatial and channel-wise correlations, the network's ability to smooth out high-frequency color transitions locally might be reduced, leading to checkerboard-like patterns in specific style features.

The Efficiency Trade-off: While the lightweight variants introduced minor visual artifacts, they offered a compelling trade-off for deployment. The visual quality saturation towards the end of training suggests that 20 epochs were sufficient, and further training yielded diminishing returns.


(a) Baseline Final Output

(b) Lightweight-v3 Final Output

Figure 5: Final comparison on a test image. The fully lightweight model (v3) produces a highly similar artistic effect despite being 87% smaller.


(a) Baseline

(b) v1 (Artifacts)

(c) v2 (Artifacts)

(d) v3 (Artifacts)

Figure 6: Detail view showing the "red spots" artifact in the yellow regions of the lightweight models.

6 Conclusion

We successfully demonstrated that replacing standard convolutions with depthwise separable convolutions is a highly effective strategy for Neural Style Transfer. Our **Lightweight-v3** model offers a practical solution for deployment, achieving an **87% size reduction** and **1.44x faster inference** with minimal impact on visual fidelity. Future work could explore combining this approach with model quantization to further eliminate the observed artifacts and enhance performance.

References

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.