applied sciences

MDPI

Article

Image Style Transfer via Multi-Style Geometry Warping

Ioana Alexandru, Constantin Nicula, Cristian Prodan, Răzvan-Paul Rotaru, Mihai-Lucian Voncilă, Nicolae Tarbă and Costin-Anton Boiangiu *

Computer Science and Engineering Department, Faculty of Automatic Control and Computers, Politehnica University of Bucharest, Splaiul Independenței 313, 060042 Bucharest, Romania; ioana.alexandru@stud.acs.upb.ro (I.A.); constantin.nicula@stud.acs.upb.ro (C.N.); cristian.prodan@stud.acs.upb.ro (C.P.); razvan_paul.rotaru@stud.acs.upb.ro (R.-P.R.); mihai_lucian.voncila@stud.acs.upb.ro (M.-L.V.); nicolae.tarba@upb.ro (N.T.)
* Correspondence: costin.boiangiu@cs.pub.ro

Abstract: Style transfer of an image has been receiving attention from the scientific community since its inception in 2015. This topic is characterized by an accelerated process of innovation; it has been defined by techniques that blend content and style, first covering only textural details, and subsequently incorporating compositional features. The results of such techniques has had a significant impact on our understanding of the inner workings of Convolutional Neural Networks. Recent research has shown an increasing interest in the geometric deformation of images, since it is a defining trait for different artists, and in various art styles, that previous methods failed to account for. However, current approaches are limited to matching class deformations in order to obtain adequate outputs. This paper solves these limitations by combining previous works in a framework that can perform geometric deformation on images using different styles from multiple artists by building an architecture that uses multiple style images and one content image as input. The proposed framework uses a combination of various other existing frameworks in order to obtain a more intriguing artistic result. The framework first detects objects of interest from various classes inside the image and assigns them a bounding box, before matching each detected object image found in a bounding box with a similar style image and performing warping on each of them on the basis of these similarities. Next, the algorithm blends back together all the warped images so they are placed in a similar position as the initial image, and style transfer is finally applied between the merged warped images and a different chosen image. We manage to obtain stylistically pleasing results that were possible to generate in a reasonable amount of time, compared to other existing methods.

Keywords: style transfer; geometric deformation; image warping; multi-style

Citation: Alexandru, I.; Nicula, C.; Prodan, C.; Rotaru, R.-P.; Voncilă, M.-L.; Tarbă, N.; Boiangiu, C.-A. Image Style Transfer via Multi-Style Geometry Warping. *Appl. Sci.* **2022**, *12*, 6055. https://doi.org/10.3390/app12126055

Academic Editors: Pedro Latorre-Carmona, Samuel Morillas and Nuria Ortigosa

Received: 13 May 2022
Accepted: 13 June 2022
Published: 14 June 2022

1. Introduction

Neural Style Transfer (NST) [1] is a technique that allows for the combination of an arbitrary image ("content image") with the texture and color from another image ("style image"), producing artistic results like drawings, paintings, and other digital art. However, the style of an artist is not defined solely by textures and colors. Borrowing these elements from an image does not guarantee the generation of quality artwork. This is a limiting factor for the standard style transfer since artists can use different techniques to make their artwork unique. Another important factor to look at, which is equally important, is the geometric deformations that happen to objects within the artist's work. Deformation of shapes is a common method to add something unique to an artwork. Such examples can be seen in caricatures, or in famous pieces such as Salvador Dalí's "The Persistence of Memory", which can be seen in Figure 1. Figures are distorted to enhance different features and to awaken certain feelings, while still being recognizable.

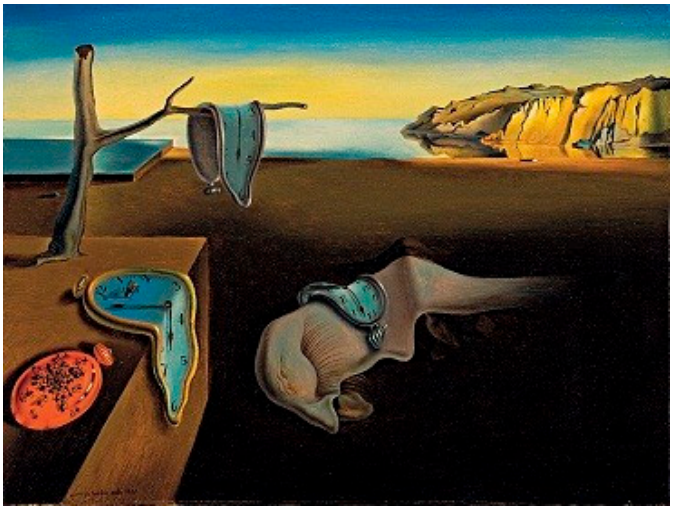


Figure 1. "The Persistence of Memory" by Salvador Dalí.

Recent research has picked up on the importance of geometry deformation and has started to become a subject of interest for obtaining realistic artwork. Most of the methods used can be classified into methods that are specialized for a particular use-case [2,3] and more general-use methods [4]. Examples of the former can be seen in the fields of character style transfer [3] and face deformation [2]. On the other hand, there have been approaches that try to broaden the domain of objects that can be deformed. Such an example is Deformable Style Transfer [4] (DST), which manages to create a technique that is not restricted to a particular use-case. Their framework can take any images that have similarities and attempt to deform them accordingly. Although great results have been achieved with the methods mentioned, they all have a common limitation: they are limited to deforming only one object at a time. Henceforth, we would like to explore some ways to apply a similar method to images with multiple objects or points of interest.

The method proposed in this paper takes one content image and multiple style images as input. Out of these style images, only one is an artistic-style image used for the "traditional" style application (which tries to imitate the colors, brush strokes, etc.), while the others are geometric-style images used for applying their shape/geometry to our content image. The first step is to detect objects in the original image, and then to attempt to find matching points for each detected object within the artistic images using a Neural Best Buddies (NBB) [5] network. The artwork that has the best match with the classified object is chosen, and the corresponding deformation is applied to the object using DST. The process is repeated until all detected objects have been deformed, and the results are merged using Poisson blending [6], thus obtaining an image deformed using the style of multiple artworks. Finally, the style of the artistic-style image is applied using NST to obtain the desired result. The proposed framework thus allows to obtain more varied artistic images in terms of style compared to previous proposed methods.

The rest of the paper is organized as follows. Section 2 is dedicated to related work. Section 3 contains the description of the framework. Section 4 presents some performance comparisons with other existing techniques, and the results of a human evaluation study performed on a set of generated images using our implementation. Lastly, Section 5 presents conclusions and possible future work in this field.

2. Related Work

The process of image stylization consists of mapping a content image and a specific style to an output image. This problem has been a significant field of research within Visual Computing for more than two decades, beginning with non-photorealistic rendering (NPR) [7] and continuing with NST and even more recently with DST.

Non-photorealistic rendering includes image synthesis from 3D objects, media, and substrate emulation, but we choose to focus on NPR [7] from images. Early algorithms aimed to map patches of pixels into brush strokes [7]. Later, the mapping targeted more notable regions. Although less common, higher forms of abstraction have been tackled using ad hoc approaches that emulate movements such as Cubism [8] and artists such as Arcimboldo [9].

Early style transfer techniques were based on algorithmic feature generation [7]. Gatys et al. [1] introduced the notion of NST, bringing a remarkable improvement to the state-of-the-art by weighing the features of a Convolutional Neural Network (CNN) trained for object recognition. In his work, Gatys represented the style using the Gram matrix of the extracted feature tensors from shallow layers of the network and the content by a feature vector extracted from a deeper layer. With respect to these variables, an output image would be created by simultaneously minimizing loss functions for both the style and the content representations. Subsequent works brought improvements by including spatial [10] or semantic [11] constraints, MRF priors [12], or by replacing the style or content representations [13].

Current state-of-the-art NST [1,12–14] techniques have become increasingly more adept at faithfully reproducing low-context features such as brush strokes and color palettes. However, their general inability to recreate high-context information such as object deformations and projective changes severely limits the quality of the reimagined artworks. This observation has prompted researchers to explore novel style transfer methods which incorporate geometric deformations.

While the body of literature regarding geometric NST is substantially more limited, recent publications have demonstrated the efficacy of this approach by consistently outperforming texture-based NST. Some of the first methods have been optimized for specific types of content such as faces [2] and text [3]. Recent works [4,15,16] have focused on developing general-purpose methods that are domain/class agnostic. Judging by the results presented in [16], this increased flexibility does not incur significant decreases in quality.

In DST, in Ref. [4], a novel optimization-based geometric style transfer technique was presented. This approach leverages NBB to find cross-domain correspondences between the provided content and the style images. Low-quality matches are further filtered using a heuristic based on NBB activation strength. This sparse set of displacements is extended to a full displacement field via Thin Plate Spline interpolation (TPS). The Style Transfer by Relaxed Optimal Transport and Self-Similarity (STROTSS) [13] texture style transfer loss is augmented with an additional warping term. This enables joint optimization of both style and geometric deformations. It should be noted that, while this approach can produce high-quality results, it is also computationally expensive.

The approaches described in [15,16] attempt to address the shortcomings of DST. In both cases, a general-purpose model is used to obtain a mapping from a 4D function of distance measurements to a 2D warp field. This approach is significantly faster in practice, because the mapping is learned offline. The main difference between the two lies in how the warp field is represented. The first uses a parametric warp function based on affine and bi-quadratic wraps, while the latter provides a more flexible implementation allowing arbitrary deformations as in DST.

Optimization-based methods have the goal of producing qualitative stylizations, but rather often this means the methods become computationally expensive in their process of backpropagation at each iteration, and gradually change the output image to match the desired error threshold. Model-based neural methods overcome these limitations and can be grouped into two different classes, each with specific tradeoffs related to their optimization policy. One of the classes trades flexibilities for speed and quality, meaning that the methods will output exceptional results quickly, but only for a limited range of styles. The other class trades quality and speed for flexibility, producing less qualitative images for a wider range of styles. With this approach, like [4], the proposed method falls into the second class, trading speed for quality and flexibility.

3. Materials and Methods

The proposed solution takes one content image (which may contain multiple objects of interest), one artistic-style image, and multiple geometric-style images as input. It produces an output image where the objects of interest are deformed based on the geometric-style images. These are then put back together into one image, and the result is styled using the artistic-style image from the input. Figure 2 shows the main steps of our method which are object detection, feature detection and warping, final composition, and style transfer.

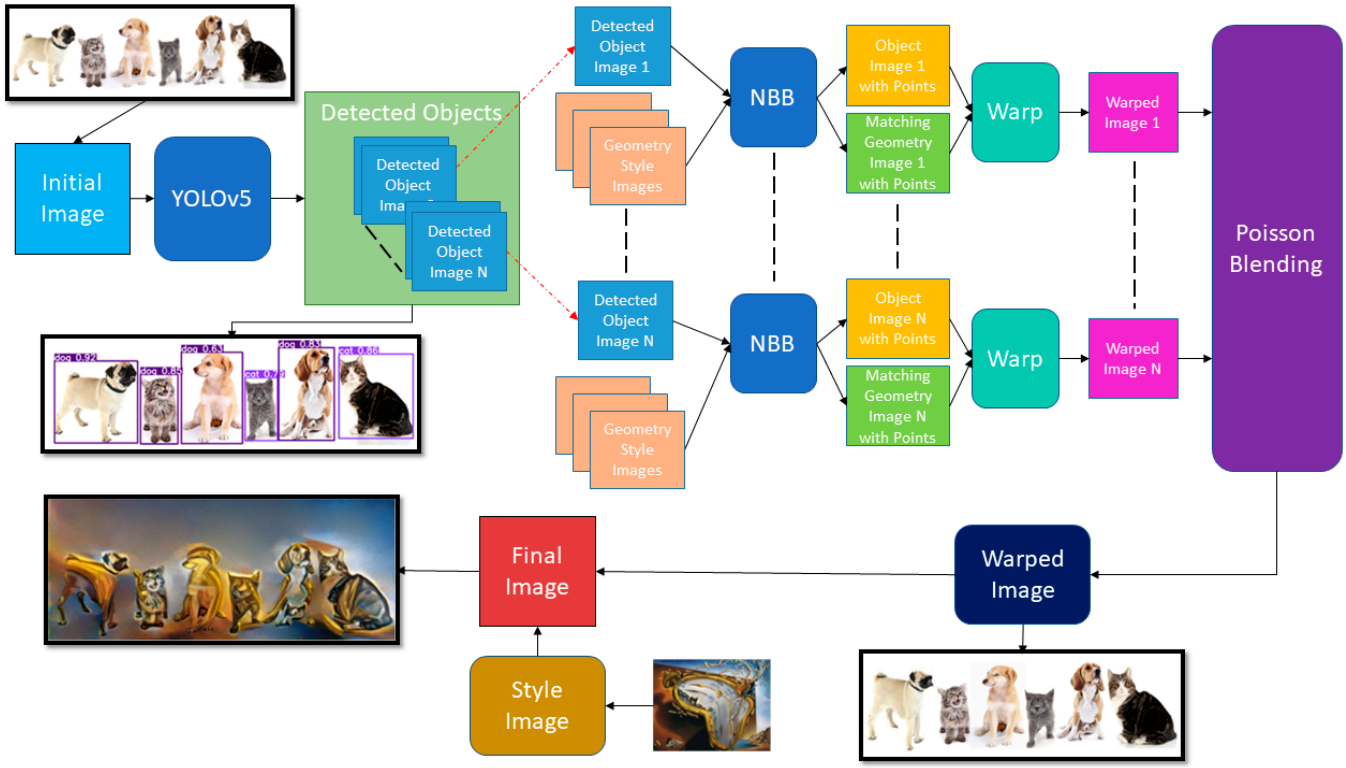


Figure 2. Overview of our proposed solution.

3.1. Object Detection

The first step requires extracting the objects that are present in the content image. A network based on the You Only Look Once (YOLO) architecture, more specifically the YOLOv5 [17] network, was used to achieve this. Classifying the objects is not necessary for the next step, only detecting them. This feature that YOLOv5 provides might prove useful for further research. Each detected object will be extracted into a separate image that will go through the rest of the pipeline.

YOLOv5 is the latest version of algorithms from the YOLO family. It manages to detect and annotate objects by dividing the input image into N grid cells of equal size, which will predict bounding boxes relative to their position and finally merge the best outputs and remove duplicates based on their probability score.

The focus of this step in the model is accuracy. The 5th version of YOLO was chosen because it outperforms its past versions in terms of accuracy. On the other hand, YOLOv5 seems to be less time efficient than YOLOv3 [18], but this is not a significant downside for the proposed solution.

3.2. Feature Detection and Warping

In the next step, for each image obtained previously, the most suited geometric-style image for the current object to be deformed with is determined using an NBB network to detect feature points on the content image and the geometric-style images.

After finding the best correspondence points for each style image, it is then decided based on some factors which pairs of style image and content image is the best candidate for the next step. The content object is then warped to resemble the artistic image chosen previously, thus obtaining the geometric deformation in the respective artistic image. This process is repeated for each object detected in the original content image.

3.2.1. Determining Correspondences with NBB

The NBB [5] network tackles the cross-domain correspondence issue by searching for pairs of neurons that represent similar features in two different images, called neural best buddies. The idea behind this technique is to pass pairs of images through a hierarchy of convolutional layers that narrow down the regions of interest by each level. The network will then keep track of neural best buddies with activations higher than a given threshold value. To choose the "best" matching pair between a specific object image and a geometry style image, we look for the pair of images that returns the most points matching certain criteria after running the NBB on them, which can be seen in Equations (1) and (2).

$$P_{NBB}(I_O(k)) = \left\{ NBB\left(I_O(k), I_{S_{NBB}}(k)\right) \middle| card\left(NBB\left(I_O(k), I_{S_{NBB}}(k)\right)\right) = max\left(card(NBB(I_O(k), I_S)), \forall I_S \in I_{S_{NBB}}\right) \right\} \quad (1)$$

$$\begin{aligned} NBB(A,B) = \{ & (a\_i, b\_i) |\ V(a\_i, b\_i) \geq t_{a_{NBB}}, \\ & \sqrt{(a_{i_x} - b_{i_x})^2 + \left(a_{i_y} - b_{i_y}\right)^2} \geq t_{d_{NBB}} \\ & card(NBB(A,B)) \leq t_{p_{NBB}} \} \end{aligned} \quad (2)$$

where $I_O(k)$ represents the k-th image detected that contains an object of interest, $P_{NBB}(I_O(k))$ represents the final set of matching points chosen, $I_{S_{NBB}}$ represents the set of geometric-style images, (a_i, b_i) a pair of matching points given to two images A, B, $V(a_i, b_i)$ the activation value for said points when considering the entire network, $t_{a_{NBB}}$ the activation threshold for considering the two points as a pair, $t_{d_{NBB}}$ the distance threshold for considering the two points as a pair, and $t_{p_{NBB}}$ represents the maximum number of pair points we want to choose. If more than the number of desired points gets chosen, then we only keep the points with the highest activation.

3.2.2. Applying Warping with DST

After obtaining all matching sets of points for all detected object images, we can supply all of them to DST [4] to obtain our set of warped images. DST works by defining different loss metrics like those found in STROTSS [13] and Gatys [1], more specifically, a $L_{content}$ and a L_{style}, the latter considering both the stylized un-warped image, as well as the stylized warped image. They also introduce an L_{warp} loss term to consider the potential loss after the warping is done. All of these loss terms are weighted with different parameters that we also consider. Equation (3) presents a slightly modified version of the original equation to fit more in line with the already proposed terms.

$$\begin{aligned} & L\left(I_{O_{DST}}(k), I_O(k), I_{S_{NBB}}(k), P_{NBB}(I_O(k)), \theta\right) \\ & = \alpha L_{content}\left(I_O(k), I_{O_{DST}}(k)\right) + L_{style}\left(I_{S_{NBB}}(k), I_{O_{DST}}(k)\right) + L_{style}\left(I_{S_{NBB}}(k), W\left(I_{O_{DST}}(k), \theta\right)\right) \\ & + \beta L_{warp}(P_{NBB}(I_O(k)), \theta) + \gamma R(f_\theta) \end{aligned} \quad (3)$$

where $I_{O_{DST}}(k)$ is the un-stylized image obtained by DST from the k-th object image and its NBB-identified geometric-style counterpart, α, β, γ represent the associated weights for each of the different loss functions, and θ is a parametrization factor for the spatial deformation. Finally, $R(f_\theta)$ represents the regularization function proposed in DST to allow smooth warping by making nearby pixels move in similar directions. Equation (4) presents the final output of the DST block.

$$\mathrm{I_{DST}} = \left\{ W\left(I_{O_{DST}}(k), \theta\right), k = \overline{1, n} \right\} \quad (4)$$

3.3. Final Composition

After all of the deformations have taken place, the modified objects are inserted back into the original image, yielding the deformed image. The reinsertion is done using Poisson blending [6] to minimize the artifacts that can occur at region boundaries. Naively pasting these deformed images into the original image would lead to visible seams, because the warping operation performed in the previous step does not preserve image boundaries. Various blending strategies can be employed to tackle this issue, such as alpha blending/feathering or Poisson blending. Alpha blending simply linearly interpolates the source and the target pixel colors. This approach is fast, but it is unsuitable for the compositing step, because it leads to excessive blurring. Unlike the previous technique, Poisson blending performs the blending operation in the gradient domain. This is done by minimizing the sum squared error of the gradients in the source and target images. Empirical tests have shown that Poisson blending provides more aesthetically pleasing images with indiscernible seams and reduced blurring. Equations (5) and (6) present the values computed for this step.

$$\nabla I_P(p_{x,y}) = \begin{cases} \nabla I_{DST}(p_{x,y}), \nabla I_{DST}(p_{x,y}) > \nabla I'_C(p_{x,y}) \\ \nabla I'_C(p_{x,y}), \nabla I_{DST}(p_{x,y}) \leq \nabla I'_C(p_{x,y}) \\ \nabla I_C(p_{x,y}), p_{x,y} \notin I'_C \end{cases} \tag{5}$$

$$I'_C = \{I_O(k), k = \overline{1,n}\} \tag{6}$$

where I_C is the original content image and I'_C represents the set of all the images that had objects detected within them.

3.4. Style Transfer

The last step consists of applying the style of an image using an NST method. Various variants of this method were tried, from the class implementation to more complex ones [12], but, in this case, the best results were obtained using STROTSS [13], mostly because this specific implementation of neural style transfer manages to correctly identify features for insertions of deformed objects (see Appendix A).

4. Results

4.1. Dataset and Test Parameters

We use two datasets in our paper: the first is the MS-COCO [19] dataset, which is used to train the YOLOv5 network on 80 different classes; while a second custom dataset is used to generate the style transferred images. The custom one is split into three parts, a content dataset, which uses the first 128 images from the COCO [20] dataset, denoted as the COCO128 dataset, a geometric style dataset, which contains 40 images used for the warping part, and a final style dataset, which contains 80 images and is used for the NST part of our approach. All input images are resized to have a long side of 256 pixels while keeping the aspect ratio; this is slightly different from the original STROTSS implementation, which starts with images at 64 pixels and goes up to 1024. This tends to speed up the results while not altering the results in a drastic manner.

Since we are combining different architectures together into a single pipeline, the model uses various hyperparameters, which are defined in Table 1, alongside the preferred values. Different values have been tested to generate both the intermediary and final images of our method, and the ones presented seem to give the best results.

Table 1. Different hyperparameters used in our framework.

Name	Tested Values	Preferred Value	Description
$t_{a_{NBB}}$	1	1	Threshold to remove pairs with lower activation values for NBB
$t_{d_{NBB}}$	3–10	5	Threshold to remove pairs that are too close to one another for NBB
$t_{p_{NBB}}$	10–100	80	Maximum number of pair points to consider for NBB
α	1–10	8	Content loss multiplier for DST
β	1–10	1	Warp loss multiplier for DST
γ	1–10	5	Regularizer multiplier for DST
lr_{DST}	0.001–1	0.1	DST learning rate
e_{DST}	10–500	250	Total number of epochs to run training for DST
β_2	0.8	0.8	The weight applied to style content for STROTSS
$tx_{STROTSS}$	1024	1024	The final texture size to consider for STROTSS

4.2. Performance Metrics

Because this topic is rather subjective in nature, there cannot be any exact metric for an objective comparison between techniques apart from the time needed to produce the outputs. Besides this measure, this paper proposes a human evaluation for a qualitative comparison between the state-of-the-art and our methods. Table 2 shows a comparison between the runtimes of our proposed method and other existing ones, for different steps, as well as images of different sizes. All tests were performed on a 3070Ti GPU, with runtimes being measured in seconds.

Table 2. Comparison between the runtimes (in seconds) of different methods.

Methods	Runtime(s)			
	Geometric Warping	Texture Rendering		
		256 × 256	512 × 512	1024 × 1024
Our method	76–119	65	111	183
Gatys et al.	N/A	13.7	31.1	117
AdaIN	N/A	0.04	0.14	0.52
DST	84–130	61	103	164
Learning to Warp	0.3–1.2	16	47	144

Apart from these evaluations, the specific measures of each component (YOLOv5, STROTSS, DST, and NBB) are relevant. The latter two fall in the same category of subjectivity as the proposed solution. A single numerical measure can be applied to both the runtime, which, for classical implementations, is about 2 min at decent resolution for NBB and 10 min at high resolution. However, the proposed solution uses a modified version of DST, which focuses solely on warping and has a runtime of approximately 2 min.

STROTSS is a technique from the NST family; therefore, it requires no training, and its metrics are visible only at runtime, and they are completely biased by the input images.

4.3. Human Evaluation

Because the target is to alter and create artistic images, there are no standard evaluation metrics. Besides the degenerative process of the proposed approach, art is tackled as the deformation reference. Therefore, traditional evaluation metrics do not apply to the presented results.

An online form was used to gather human input for the evaluation of the presented results. The results can be found in Figure 3.

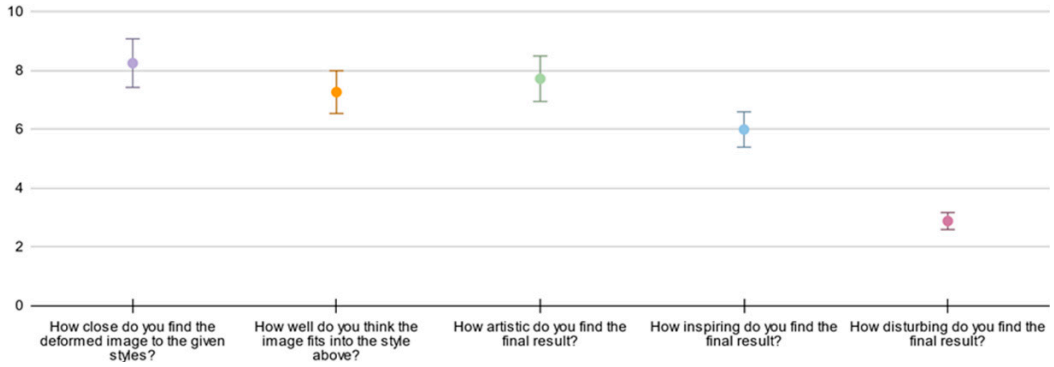


Figure 3. Survey results.

4.4. Heuristics

One of the components in the proposed solution that is open for further research and experimentation is the heuristic used to decide which artistic image is best suited to deform one object from the content image. Choosing another heuristic can lead to significantly better or worse results. Of course, manually choosing this image is possible, but automated methods were explored, so that the proposed pipeline would require minimal human input. In the following, the heuristics used during the development of the solution are described.

The most obvious and simple one, briefly touched upon during Section 3, is choosing the artistic image that has the most matching points resulting after NBB was applied. The advantage of this heuristic is straightforward. It is very simple to implement and requires minimal effort, since only one variable is needed to hold the maximum number of points found. The problem found with this heuristic is that NBB is not always reliable when it comes to finding matching points. For instance, it is possible for an image of a dog, depending on the artistic characteristics of the image, to find more matching points with other images that do not contain dogs, thus resulting in an unreliable deformation since the objects in the images might or might not be very different. One potential improvement for this could be considering the classification resulting from YOLOv5, but this is not trivial, since such object-detection methods do not give great results when trying to recognize objects in artistic images that have already been deformed. This is one area where further research would be needed.

Another approach was to compute a "distance" between the content image and the artistic style image based on the information found by the NBB. This distance was computed using the L-2 norm. This heuristic produced mixed results. Some classes of images were more accurately found, while others were still mismatched. While the advantage is debatable, the disadvantage is obvious. With this approach, the pipeline is significantly slower and more computationally expensive. However, those disadvantages are not notable for the purpose of this paper, thus this was chosen as the main heuristic in our experiments.

4.5. Generated Images

Figure 4 showcases some of the generated images obtained by applying the proposed method to the cats and dogs image. In the top-right corner of each generated image, the style image is also shown for reference.

Figure 4. Results of the proposed method, applied to the "cats and dogs" image, for various style images.

5. Discussion

5.1. Improvements to State-of-the-Art

The approach described in this paper offers more flexibility and artistic freedom compared to the existing State-of-the-Art (SotA) techniques. The proposed approach can deal with more complex images containing multiple objects of interest from various classes. The use of YOLOv5 allows automatic identification of relevant objects and their corresponding classes. The selection of adequate warp-style images is also managed internally using a heuristic approach.

The main advantage of the proposed technique is that each object can be deformed individually using the best-matching warping style. For a direct comparison with the most similar SotA technique, DST, the proposed pipeline can be run with a single style image, even though, normally, multiple style images would be used against the content image in Figure 5. The results of this run can be seen in Figure 6. The same inputs were also used for a DST run, for which the results can be seen in Figure 7.

It can be easily noticed that when using the DST method, the warping is not as pronounced, as the entire image is taken as one object and mapped to the features in the style image. A slight peak on the car on the left that may correspond to the house and a small curvature on the car on the right that may match the field under the tree can be observed. Comparatively, the warping of each individual object is more noticeable in the result using the proposed method, as it can be clearly seen that both cars have been warped to match the shape of the house.

Figure 5. The image used for comparison; left—content image; right—style image.

Figure 6. Results using the proposed method.

Figure 7. Results using the DST method.

5.2. Limitations of the Proposed Solution

Although the proposed solution works in terms of the functionality of the pipeline, the quality of the warping results is heavily reliant on the alignment between the content object and the style object. Consequently, the quality of the initial matching becomes a deciding factor in the final output. We observed that when major deviations exist or when insufficient matches are recognized, the results are less than ideal; these flaws can be observed in Figure 8. Based on preliminary testing, it is safe to say that this approach is not suitable for warping between objects from completely different classes, although it can, very occasionally, lead to interesting results. While NBB can often detect some matches, they are too sparse and do not provide enough context to perform proper deformations. Figure 9 shows how applying STROTSS-based style transfer as a final composition step proved to be a viable strategy. For the DST method, it is necessary to jointly optimize warp and style at the same time; however, it does not seem necessary for the proposed method.

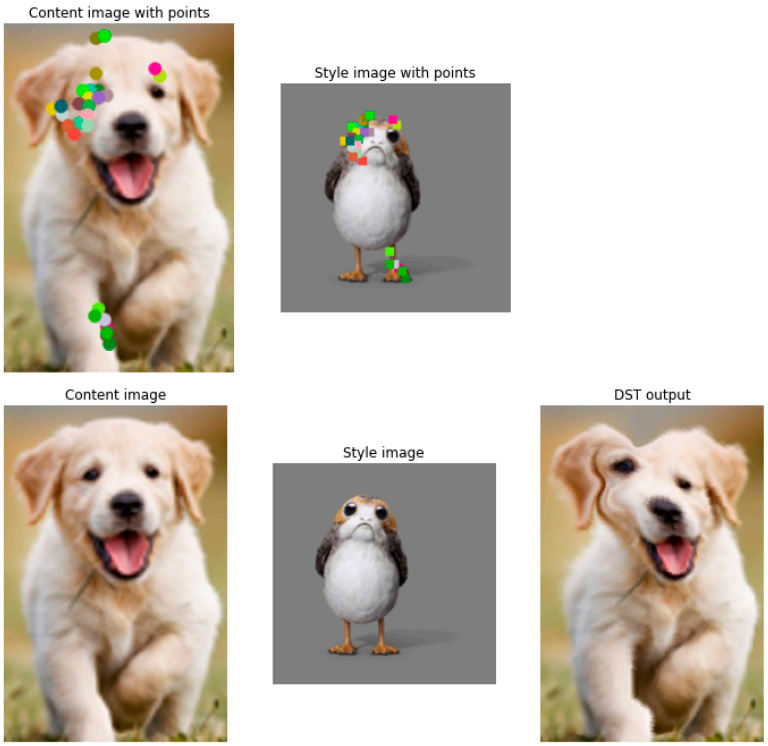


Figure 8. Results on two images whose features and alignment do not match.

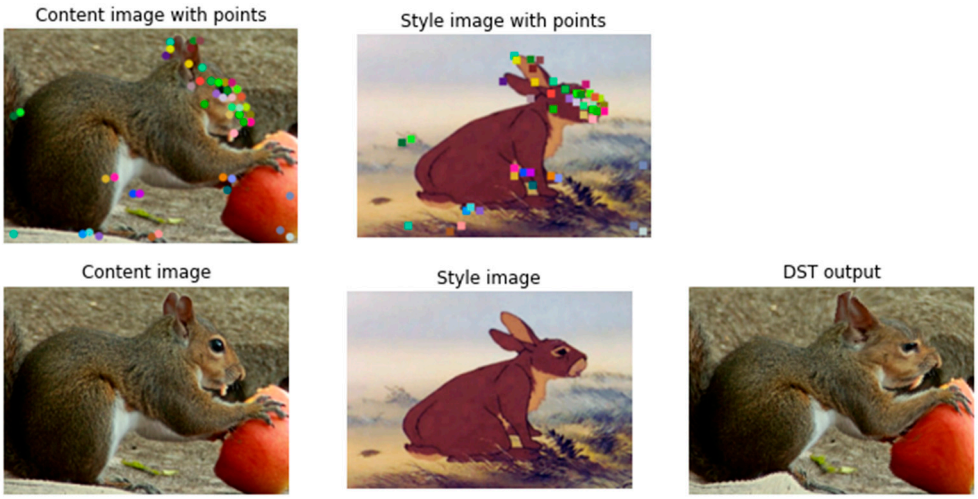


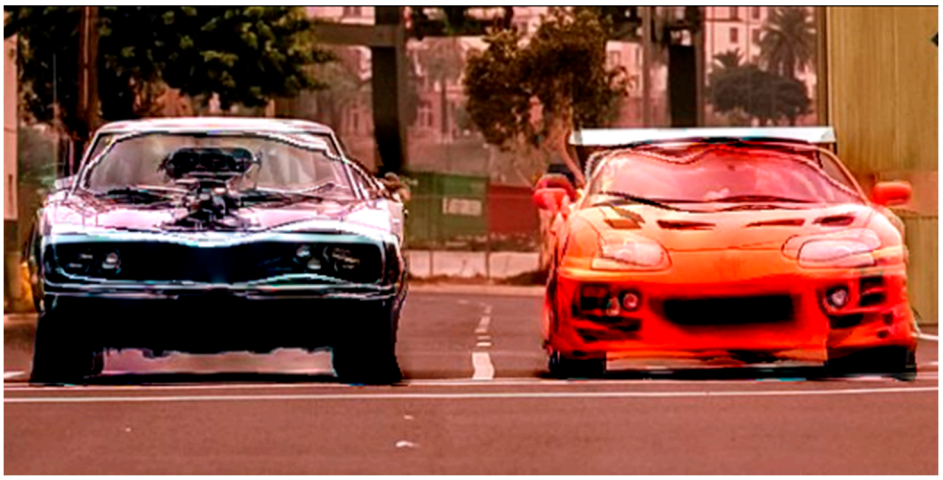
Figure 9. Results on two images whose features and alignment do not match.

Another limitation comes from the fact that handling of geometric deformations was realized using a framework like the one described in DST, which is heavily reliant on accurate point correspondences to produce acceptable results. Though initially touted as a robust cross-domain feature matching algorithm, NBB often fails to discover meaningful correspondence points. Since these matches are directly used to derive the warp field, any inaccuracy leads to a substantially degraded output. For this reason, a potential improvement could be to use a Warp Network akin to the one described in [16]. That approach should be more robust to outliers and may in fact require less parameter tweaking to provide acceptable results.

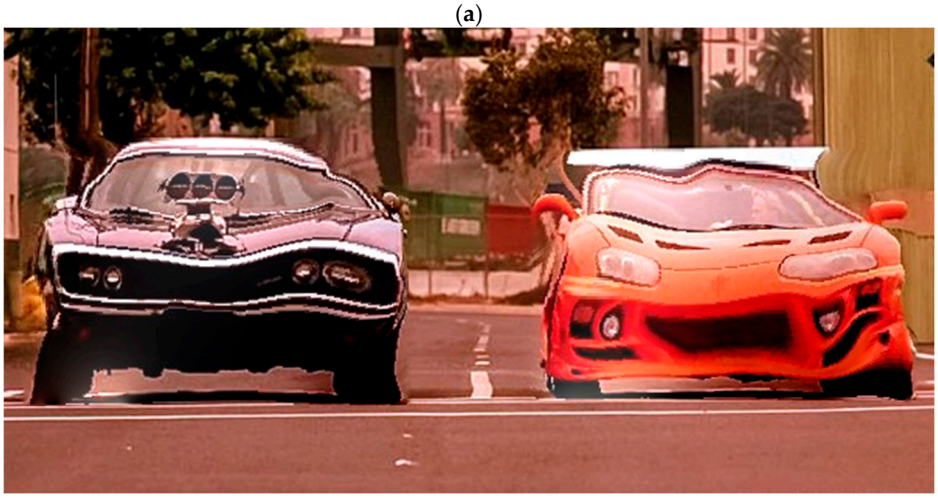
Another shortcoming of the proposed solution is the limited range for different classes of objects the YOLOv5 network was trained to detect. Increasing that range would increase the number of points of interest from an image to be deformed. For example, with the proposed implementation, a tree is not identified as an object and thus not deformed (so it

remains part of the background). If it were part of the classification set, more interesting results, where even trees can be warped in new ways, might be produced. Furthermore, as described in Section 4.4. Heuristics, this classification can be used in the heuristic for matching objects of interest to warping styles.

Another part of the pipeline that could be further improved is the blending technique. Poisson blending was used, which offers great results when compared to similar techniques like Alpha blending. A comparison between the two techniques can be seen in Figure 10. Nonetheless, unwanted behavior can appear for overlapping objects in the image, which would lead to spatial discontinuities of pixels in the final image. This is because the cropped objects in the image are first deformed, then blended back into the image. This could result in the blend of the last detected crop overwriting already deformed structures from another crop.



(**a**)



(**b**)

Figure 10. Outputs of different blending techniques: (**a**) Alpha blending; (**b**) Poisson blending.

Yet another potential improvement that could be made in the object detection step, which may even reduce the need for improved blending, would be better cropping. The objects of interest are extracted by making rectangular crops and then these crops are warped, so the warping may inadvertently affect part of the background that is included in these crops as well. If a more sophisticated approach for cropping out the objects as close to their outline as possible were to be used, then the warping could be applied only on the objects, with the background either being left un-warped or considered as a separate "object" and warped independently, depending on the desired result.

Something worth mentioning is the idea of limiting the number of artistic images available for deformation. In most of our experiments, we had enough artistic images

to cover every object in the content image, and if not, we used the same style image to perform multiple deformations. What happens if we decide to limit the number of artistic images, and no artistic image can be used twice? We experimented lightly with this idea, and preliminary results show that, while this significantly speeds up the pipeline, it is not ideal when a content image has multiple objects of interest from the same class—in that case, it may make sense to use the same style image for all of the deformations. Since speed was not our focus at the time, we did not pursue this further.

6. Conclusions

There has been an increase in researchers' interest in the field of AI-generated art. While there have been significant advances focusing on artwork with a single object of interest, there is still much more research to be done when looking at more complex art with multiple features. In this paper, we presented a pipeline that can be used for transferring the style of one image to another using various steps, such as object detection, warping based on NBB, Poisson blending, and finally the actual style transfer. Our method, while not necessarily the fastest when compared to other existing methods, does provide artistically pleasing results. Our method is similarly the first one to attempt integrating object detection into the style transfer process, which allows us to obtain more varied and unique results when compared to other methods. We believe that our approach for warping and style transfer of multiple geometric de-formations for images with several objects of interest is a significant step in the advancement of this field.

Author Contributions: Conceptualization, I.A., C.N., C.P., R.-P.R., M.-L.V., N.T. and C.-A.B.; Data curation, I.A., C.N., C.P. and R.-P.R.; Formal analysis, I.A., C.N., C.P. and R.-P.R.; Funding acquisition, C.-A.B.; Investigation, I.A., C.N., C.P. and R.-P.R.; Methodology, I.A., C.N., C.P., R.-P.R., M.-L.V., N.T. and C.-A.B.; Project administration, C.-A.B.; Resources, M.-L.V., N.T. and C.-A.B.; Software, I.A., C.N., C.P. and R.-P.R.; Supervision, M.-L.V., N.T. and C.-A.B.; Validation, I.A., C.N., C.P., R.-P.R., M.-L.V., N.T. and C.-A.B.; Visualization, I.A., C.N., C.P., R.-P.R., M.-L.V., N.T. and C.-A.B.; Writing—original draft, I.A., C.N., C.P. and R.-P.R.; Writing—review and editing, M.-L.V., N.T. and C.-A.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Obviously, not all images that we generate are going to be visually appealing. Depending on the parameters we choose for our networks, some images are bound to contain noise or various other unpleasing artifacts. Something that is very common is for noise to be present around the objects that are selected for warping, making them look weird. This is mostly, in fact, due to the use of Poisson blending when combining the multitude of warped images back into the original image. On a similar note, it is also possible that the style around objects of interest inside the image also gets applied in a stronger fashion, when compared to the rest of the image, which can also lead to equally intriguing, but not necessarily appealing, results. Figure A1 shows some of these more undesirable images obtained from various networks.

Figure A1. Results of unsuitable NST networks.

To correct some of these noisy features, we used STROTTS, which was able to properly match and identify the style to use around warped objects. Figure A2 shows two images generated for similar styles between one of our initial implementations and the implementation using STROTTS.

(**a**) (**b**)

Figure A2. Comparison of results between the default NST implementation (**a**) and STROTTS (**b**).

References

1. Gatys, L.A.; Ecker, A.S.; Bethge, M. Image Style Transfer Using Convolutional Neural Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2414–2423. [CrossRef]
2. Yaniv, J.; Newman, Y.; Shamir, A. The Face of Art: Landmark Detection and Geometric Style in Portraits. *ACM Trans. Graph.* **2019**, *38*, 1–15. [CrossRef]
3. Yang, S.; Liu, J.; Lian, Z.; Guo, Z. Awesome Typography: Statistics-Based Text Effects Transfer. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2886–2895. [CrossRef]
4. Kim, S.S.Y.; Kolkin, N.; Salavon, J.; Shakhnarovich, G. Deformable Style Transfer. In *Computer Vision—ECCV 2020. ECCV 2020. Lecture Notes in Computer Science*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M., Eds.; Springer: Cham, Switzerland, 2020; Volume 12371.
5. Aberman, K.; Liao, J.; Shi, M.; Lischinski, D.; Chen, B.; Cohen-Or, D. Neural Best-Buddies: Sparse Cross-Domain Correspondence. *ACM Trans. Graph.* **2018**, *37*, 1–14. [CrossRef]
6. Pérez, P.; Gangnet, M.; Blake, A. Poisson Image Editing. *ACM Trans. Graph.* **2003**, *22*, 313–318. [CrossRef]
7. Haeberli, P. Paint by Numbers: Abstract Image Representations. *SIGGRAPH Comput. Graph.* **1990**, *24*, 207–214. [CrossRef]
8. Collomosse, J.P.; Hall, P.M. Cubist Style Rendering from Photographs. *IEEE Trans. Vis. Comput. Graph.* **2003**, *9*, 443–453. [CrossRef]
9. Huang, H.; Zhang, L.; Zhang, H.-C. Arcimboldo-like Collage Using Internet Images. *ACM Trans. Graph.* **2011**, *30*, 155. [CrossRef]
10. Cole, F.; Belanger, D.; Krishnan, D.; Sarna, A.; Mosseri, I.; Freeman, W.T. Synthesizing Normalized Faces from Facial Identity Features. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 3386–3395. [CrossRef]
11. Champandard, A.J. Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artworks. *arXiv* **2016**, arXiv:1603.01768. [CrossRef]
12. Li, C.; Wand, M. Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis. *arXiv* **2016**, arXiv:1601.04589. [CrossRef]
13. Kolkin, N.; Salavon, J.; Shakhnarovich, G. Style Transfer by Relaxed Optimal Transport and Self-Similarity. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 10043–10052. [CrossRef]
14. Chen, X.; Yan, X.; Liu, N.; Qiu, T.; Ni, B. Anisotropic Stroke Control for Multiple Artists Style Transfer. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 3246–3255.
15. Liu, X.-C.; Li, X.-Y.; Cheng, M.-M.; Hall, P. Geometric Style Transfer. *arXiv* **2020**, arXiv:2007.05471.
16. Liu, X.-C.; Yang, Y.-L.; Hall, P. Learning to Warp for Style Transfer. In Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, 20–25 June 2021; pp. 3701–3710. [CrossRef]
17. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; NanoCode012; Kwon, Y.; TaoXie; Fang, J.; imyhxy; Michael, K.; et al. Ultralytics/Yolov5: V6.1—TensorRT, TensorFlow Edge TPU and OpenVINO Export and Inference. *Zenodo* **2022**. [CrossRef]
18. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
19. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Computer Vision—ECCV 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer International Publishing: Cham, Switzerland, 2014; pp. 740–755. [CrossRef]
20. Caesar, H.; Uijlings, J.; Ferrari, V. COCO-Stuff: Thing and Stuff Classes in Context. *arXiv* **2018**, arXiv:1612.03716. [CrossRef]