DOI: 10.1111/cgf.14744
EUROGRAPHICS 2023 / K. Myszkowski and M. Nießner
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 42 (2023), Number 2

Subpixel Deblurring of Anti-Aliased Raster Clip-Art

J. Yang[1], N. Vining[1,2], S. Kheradmand[1], N. Carr[3], L. Sigal[1], and A. Sheffer[1]

[1]University of British Columbia, Canada
[2]NVIDIA, Canada
[3]Adobe, United States of America

Figure 1: *Directly vectorizing anti-aliased low resolution (64x64px) clip-art images (a) using state of the art methods [Vec17] (c) produces inadequate results (note the missing arm). Our subpixel deblurring method produces blur-free double resolution outputs (e) which are well aligned with viewer expectations both as-is and after vectorization (f). We first predict a low-blur subpixel approximate image (d) and then use perception driven discrete optimization to obtain the final blur-free output (e). Our outputs are significantly more aligned with viewer expectation than those produced by state of the art superresolution [WXDS21] (b) and vectorization (c) methods. Sub-pixel deblurring benefits applications such as vectorization (f) and recoloring (g). Please zoom-in to see details.*

Abstract

Artist generated clip-art images typically consist of a small number of distinct, uniformly colored regions with clear boundaries. Legacy artist created images are often stored in low-resolution (100x100px or less) anti-aliased *raster form. Compared to anti-aliasing free rasterization, anti-aliasing blurs inter-region boundaries and obscures the artist's intended region topology and color palette; at the same time, it better preserves subpixel details. Recovering the underlying artist-intended images from their low-resolution anti-aliased rasterizations can facilitate resolution independent rendering, lossless vectorization, and other image processing applications. Unfortunately, while human observers can mentally deblur these low-resolution images and reconstruct region topology, color and subpixel details, existing algorithms applicable to this task fail to produce outputs consistent with human expectations when presented with such images. We recover these viewer perceived blur-free images at subpixel resolution, producing outputs where each input pixel is replaced by four corresponding (sub)pixels. Performing this task requires computing the size of the output image color palette, generating the palette itself, and associating each pixel in the output with one of the colors in the palette. We obtain these desired output components by leveraging a combination of perceptual and domain priors, and real world data. We use readily available data to train a network that predicts, for each anti-aliased image, a* low-blur *approximation of the blur-free double-resolution outputs we seek. The images obtained at this stage are perceptually closer to the desired outputs but typically still have hundreds of redundant differently colored regions with fuzzy boundaries. We convert these low-blur intermediate images into blur-free outputs consistent with viewer expectations using a discrete partitioning procedure guided by the characteristic properties of clip-art images, observations about the antialiasing process, and human perception of anti-aliased clip-art. This step dramatically reduces the size of the output color palettes, and the region counts bringing them in line with viewer expectations and enabling the image processing applications we target. We demonstrate the utility of our method by using our outputs for a number of image processing tasks, and validate it via extensive comparisons to prior art. In our comparative study, participants preferred our deblurred outputs over those produced by the best-performing alternative by a ratio of 75 to 8.5.*

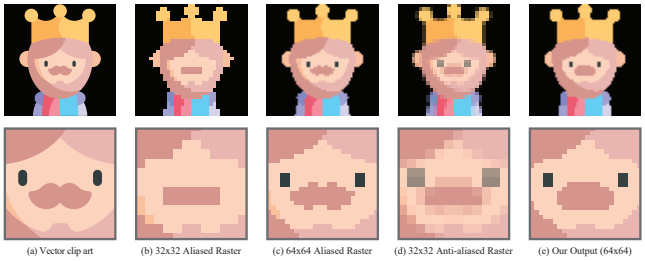


Figure 2: *Aliased rasterizations (b,c) of vector clip art (a) have a self-evident color palette and region topology. Anti-aliased renders of the same input (d) blur the color palette and make the single-color region topology harder to discern. Compared to the aliased renders at* the same *(here* 32×32*) resolution (b), however, they often better preserve sub-pixel features, such as the eyes in this example (d). Our subpixel deblurring method (e) uses such blurry images (d) as input, and produces double-resolution blur-free outputs consistent with viewer expectations.*

1. Introduction

Artist-generated clip-art images, consisting of small sets of distinct, uniformly-colored regions, are ubiquitous in digital media applications [DSG*20]. While designed to convey blur-free content, for a range of historic reasons legacy clip-art images, including sprites, are frequently stored in low-resolution (100x100px or less), *anti-aliased*, raster form (Fig. 1a). Antialiasing blurs inter-region boundaries, and at low resolutions obscures the artist-intended region topology and color palette (Fig. 2d). At the same time, antialiasing enables better preservation of subpixel details when compared to aliased, or blur-free, outputs at the same resolution (Fig. 2b). Recovering the underlying artist-intended images from these low-resolution anti-aliased rasterizations can facilitate a range of image processing applications including vectorization (Fig. 1f) and recoloring (Fig. 1g, Sec 6). Our work aims to recover these underlying images given the anti-aliased inputs.

When presented with low-resolution anti-aliased inputs, human observers can mentally approximate the underlying artist-intended *blur-free* visuals [TFCRS11].. However, prior research [HDS*18, DSG*20] suggests that observers do not hallucinate details that are not strongly hinted at by the inputs (e.g. the puffy sleeves or the elaborate mustache in Fig. 2a). Our observations (Sec. 3) suggest that the original details that humans discern in anti-aliased images are typically visible in aliased double-resolution renders of the original vector images (Fig. 2c). Furthermore, humans do not hallucinate details that aliased double-resolution images do not capture. To obtain results consistent with viewer expectations, while capturing subpixel details and avoiding hallucination of unexpected details, we therefore focus our efforts on recovering *double-resolution* blur-free raster clip-art images well aligned with viewer perception (Figs. 1, 2e). The outputs produced by our method can be used for a range of image processing applications which benefit from *subpixel deblurring* (Sec 6, Fig. 1fg), and are significantly better aligned with viewer expectations than those produced via existing methods that can be applied to this task (Secs 2, 6, Figs. 1bc).

One core difference between our input anti-aliased images and the desired deblurred outputs is the size of the *color palettes*, or the number of distinct colors present in the image. While typical anti-aliased images have palettes with 100 to 200 distinct colors (Tab. 1), the source vector content they are rasterized from, and blur-free rasterizations of that vector content, have on average about a dozen colors. To enable robust processing by downstream applications, and to produce outputs consistent with viewer expectations and artist intent, our output color palettes *must* be similarly *compact*. At the same time, the actual size of the intended or viewer perceived palettes of our input images varies and is unknown - the originating vector images of the inputs we tested had palettes with as few as 3 colors and as many as 172. Thus while, the consistency observation above suggests that subpixel deblurring can potentially be learned from pairs of anti-aliased and blur-free rasterizations of the same clip-art vector images, where the blur-free image has double the resolution of the anti-aliased one (Fig. 2dc), satisfying the compactness requirement without knowing the viewer expected palette, or even its size, in advance makes subpixel deblurring a much more challenging problem than those addressed by traditional upsampling or superresolution methods (Sec 2, Fig. 1c). While some state-of-the-art research seeks to learn palettes or alphabets to use during translation tasks, most machine learning methods operate continuously and produce outputs with thousands of colors (e.g. [IZZE17, WXDS21]). The alternative approach of training models using a small and fixed palette size [RGLM21] had been only demonstrated to work on very small palette sizes (3 for [RGLM21]) and is not applicable in our setting where size can vary arbitrarily and the average palette size is about 10.

We algorithmically obtain a compact color palette and a corresponding blur-free colorized output using a two stage process motivated by the observations above (Fig. 1de). We leverage pairs of anti-aliased and blur-free double-resolution renders of the same vector inputs to train a deep learning method for subpixel deblurring (Fig. 1d, Sec 4). The outputs of this method are closer to the ground truth in color space than those produced by state-of-the-art superresolution methods, retrained on our data (e.g. [WXDS21], Figs. 1c, 5). They clearly reveal fine details captured by the anti-aliasing such as the snowman's mouth and eyes, and contain observable subpixel refined inter-region boundaries. At the same time, as expected, these images have thousands of colors, and close inspection reveals them to contain residual blur, color variation and speckles.

We generate our final outputs from this data by leveraging the global characteristics of typical clip-art images and perceptual priors. We utilize the following observations: (1) typical clip art images consist of a small set of regions with a compact color palette: (2) human ability to distinguish between adjacent region colors is correlated with region size [Sto03, SASS14] , and (3) when presented with ambiguous inputs, humans opt for simpler explanations [WEK*12, Kof55]. We consequently cast the computation of the output regions and their colors as a discrete optimization problem of assigning color values to pixels. We first use observations about anti-aliasing mechanisms to obtain a candidate color palette for our output clip-art images, and then assign each pixel in the output image one of these candidate palette colors by solving a constrained graph labeling problem whose formulation is motivated by domain and perceptual priors (Fig. 7c, Sec 5).

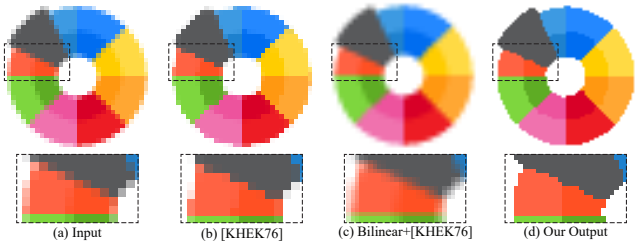


Figure 3: *Given the input image (a), traditional deblurring and sharpening methods such as [KHEK76] (b) reduce but do not eliminate anti-aliasing blur, and operate in-place producing outputs at the same resolution as the input. Generating sub-pixel outputs by first applying bilinear magnification followed by deblurring (c) enhances rather than reduces the blur. Our method computes blur-free outputs, recovering sub-pixel details (d).*

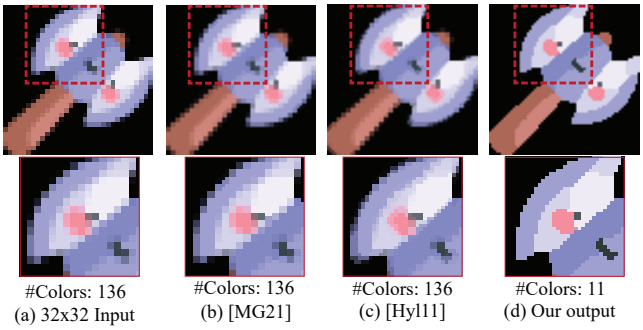


Figure 4: *Magnification methods such as (b) MMPX [MG21] and (c) XBR [Hyl11]) aim to preserve the input's look while doubling its resolution, and thus produce blurry outputs on anti-aliased inputs (a). When presented with the same output our method (d) produces blur-free outputs with compact color palettes.*

We evaluate our method on 112 diverse inputs across multiple resolutions (Sec 6), demonstrate the utility of our outputs for a number of applications (Fig. 1fg, Sec 6), and validate the superiority of our method via quantitative and qualitative comparisons to prior art and ground truth data. Our measurements confirm that the color space distance from our outputs to ground truth data is approximately half that of the closest competitor. Participants in a perceptual study comparing our results to those produced by prior methods preferred our outputs by a ratio of 75% to 8.5% over the best algorithmic alternative at resolutions below 100*px*.

2. Related Work

While few, if any, methods address the exact problem we tackle, our work has connections to methods for image segmentation, deblurring, magnification, and vectorization.

Semantics-free Image Segmentation. Our problem can be thought as one of obtaining a color palette and then assigning one of these colors to each quarter-pixel in our input image. As such it has connections to semantics-free image segmentation methods. These methods aim to approximate high-resolution detailed inputs with a compact set of color-coherent regions by grouping together similarly colored pixels into regions of roughly balanced size, and assigning a representative color to each region [FH04, OBW*08, LL06, SLWS07, XLY09, WZGW17, XSTN14, KKT20, XK17]. These methods can, in theory, be applied either directly to our inputs or to upsampled versions of these inputs. However, color similarity is not a reliable grouping cue in our context since pixels on the blurred boundary between highly distinct regions can have a shade that significantly differs from that of these regions.

Furthermore, size balancing is not a desirable property in our context - we often want output regions to dramatically differ in size (e.g. the eyes in Fig. 2). As a result, the outputs of these methods are very far from the viewer expected ones on the typical inputs we process (for example, see the inset on the left produced by applying the method of [FH04] to the input in Fig. 2d).

Traditional Deblurring and Sharpening. Traditional blind deblurring approaches (e.g. [FSH*06,LWDF09]) operate on input images that are assumed to be a convolution of a sharp image and an unknown blur kernel, plus noise, and seek to recover both the kernel and the unblurred image upon which the convolution was applied. These approaches target high-resolution natural imagery, and assume the existence of a blur kernel that models the effects of camera shake or lens defocus. Our input images are not consistent with this model: rather than being "blurred" by convolution, the anti-aliasing blur is caused by downsampling vector or high-resolution images to a lower-resolution target. Furthermore, our images are noise-free and contain subpixel information baked in by the anti-aliasing that we seek to exploit while deblurring and upscaling the input. Sharpening filters such as [TM98, KHEK76, KKD09] mimic deblurring by enhancing edge contrast and smoothing low contrast regions. When applied to our data, these filters visually sharpen the images, with [KHEK76] producing the sharpest results, but still retain much of the anti-aliasing blur (Fig. 3b). Sparsity-based image smoothing methods (e.g. [XLXJ11]) can produce piecewise constant color patches, but retain anti-aliasing blur while also removing important color differences (inset).

Moreover, these methods operate in place, and thus are not applicable as-is for subpixel deblurring. Upsampling the inputs first (via, for instance, bilinear magnification) and then applying the filter results in blurry outputs (Fig. 3c). Our method (Fig. 3d) produces outputs with much more compact palettes and that are significantly closer to our ground truth data.

Magnification of Clip-Art Imagery. Magnification methods for clip- and pixel-art data [Hyl11, Ste03, MG21] aim to double the resolution of the input while preserving its look. The method of [Ste03] intentionally anti-aliases the magnified images, the exact opposite of our goal. [Hyl11, MG21] target anti-aliasing free inputs and aim to preserve the input characteristics and thus utilize the original color palette. Consequently when presented with our inputs they preserve, rather than remove, the anti-aliasing blur (Fig. 4). Our method successfully addresses a different problem, that of subpixel deblurring. Our outputs can be subsequently fur-

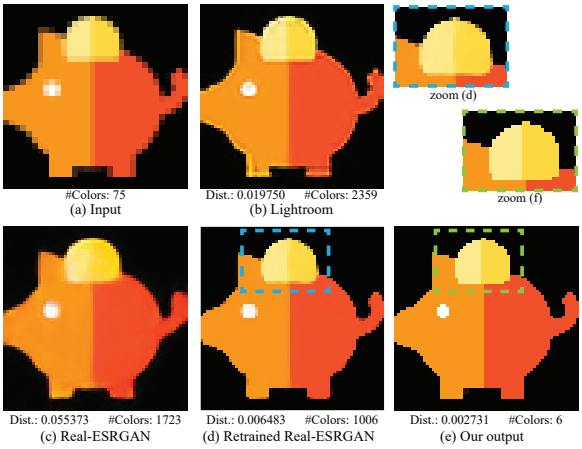


Figure 5: *Comparison to superresolution methods: (a) input; (b) Photoshop Lightroom [Ado21]; (c) Real-ESRGAN [WXDS21] (d Real-ESRGAN trained on our data; (e) our output.*

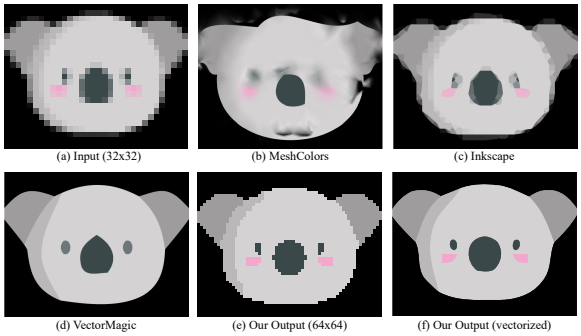


Figure 6: *Given an input 32px anti-aliased image (a), state of the art vectorization methods [HEK21] (b), Inkscape [Ink20] (c), and VectorMagic [Vec17] (d) produce inadequate results. Our output is well aligned with viewer expectations. Vectorizing our outputs using VectorMagic (f) produces vector outputs similarly well aligned with viewer expectations.*

ther magnified using the approaches of [Hyl11, MG21] facilitating resolution-independent rendering (Sec 6).

Superresolution. Superresolution methods target high-resolution natural images, and use convolutional networks trained on pairs of input and ground truth images to create high resolution versions of the input [DLHT15, WXDS21, ZLVGT21, LSZ*21, LCS*21]. While they can be applied to generate sub-pixel magnifications of the inputs we process, they are not trained to deblur the inputs in the process. Consequently, applying state-of-the-art commercial tools [Ado21] or state-of-the-art methods such as Real-ESRGAN [WXDS21] (using their double resolution model) as-is to our data retains, and sometimes even exaggerates, the anti-aliasing blur (Fig. 5bc). Retraining Real-ESRGAN [WXDS21] on our dataset (pairs of anti-aliased and double-resolution, non-anti-aliased, images) produces outputs (Fig. 5d) which are more consistent with viewer expectation and are up to an order of magnitude closer to the ground truth data in color space than those produced without retraining (Fig. 5c) . Unfortunately this approach produces output palettes with thousands of colors (Sec 6). Our outputs are on average 45% closer to ground truth than than those of retrained Real-ESRGAN and have only about a dozen colors on average (Fig. 5e).

Image Vectorization. The vast majority of image vectorization methods for both natural [HEK21, FLB16, XSTN14, LL06, OBW*08, SLWS07, WZGW17, XLY09, Ado17] and clip-art [YCZ*16, ZCZ*09, Ink20, HDS*18, DSG*20, KL11, SBv05] images first segment the inputs into color coherent regions using semantics-free methods similar to the ones described above, and then fit the boundary of each such region using piece-wise smooth vector curves. As discussed above, semantics-free segmentation methods fail to produce viewer expected regions on anti-aliased clip-art images; in turn, this inadequate segmentation leads to poor vectorization outcomes (Fig. 6bc). This limitation motivates most research on vectorization of clip-art imagery to consider only anti-aliasing free inputs [HDS*18, DSG*20, KL11], and to list processing of anti-aliased data as future work. These methods can therefore directly benefit from using our method as the first step in their vectorization pipeline when processing anti-aliased clip-art (Fig. 6f).

Dominici et al. [DSG*20] extend their vectorization method designed for anti-aliasing free inputs to binary anti-aliased images consisting of a background and a foreground region; we process inputs with arbitrary color count. Reddy et al. [RGLM21] vectorize clip art images using end-to-end neural networks trained with a fixed two or three color palette size, and operate on a narrow class of input images (e.g. face emojis or digits) with four regions or less. Our method addresses blur-free magnification of general anti-aliased clip-art inputs, and does not require the number of regions or colors to be capped or known *a priori*. On the inputs we handle, viewer-expected outputs have on average a dozen colors, with the number ranging from three to over a hundred.

The VectorMagic [Vec17, Die08] clip-art vectorization software vectorizes both aliased and anti-aliased inputs, and performs well at resolutions above $100 \times 100px$; however, when presented with anti-aliased inputs at lower resolutions their outputs tend to lack many viewer perceived features (e.g. arm and hatband in Fig. 1, cheeks in Fig. 6).

3. Problem Statement and Overview

Perception of Anti-Aliased Clip-Art. Visual perception research [TFCRS11] suggests that human observers are adept at recovering intended content from rasterized inputs, overcoming both downsampling and anti-aliasing artifacts. In their research on vectorizing aliased clip-art images, Hoshyari et al. [HDS*18] point to three key factors that likely impact the mental process humans employ when recovering content from raster inputs: *accuracy*, *simplicity*, and *continuity*. Accuracy predicts that the images humans envision are ones that, when rasterized using the same process as the one used to create the inputs, reproduce (or nearly reproduce) these inputs. The simplicity principle of Gestalt psychology indicates that human observers opt for the *simplest* interpretations possible that are *consistent* with the observed inputs [WEK*12]. In particular, recent computer graphics research [DSG*20, HDS*18] strongly suggests that, when presented with raster imagery, humans do not

hallucinate details not present in the inputs. Lastly, continuity suggests that the inter-region boundaries viewers envision are piecewise smooth, and are not restricted to follow the edges of the pixel grid.

While Hoshyari et al. [HDS*18] then proceed to analyze the impact of these factors on viewer perception of blur-free raster clip-art, we are interested in understanding how they apply to anti-aliased data. Since little is known about the exact cues viewers employ when presented with anti-aliased low-resolution clip-art imagery, we use observations about human perception, properties of typical clip-art imagery, and characteristics of the anti-aliasing process to identify these cues.

We observe that, in our context, accuracy and simplicity taken together strongly suggest that the mental images viewers assemble closely resemble the originating vector images, and are at most as complex as these originating images. In other words, these mental images are likely to have at most as many colors as the originating images, have region topology and details which are similar to, or simpler than, those of the originating vector art, and not contain content inconsistent with the raster input. Applying these observations in a practical setting suggests analyzing the color palettes and other properties of the originating images, and the impact of the antialiasing process on the degree to which these remain visible in the raster images that viewers are presented with.

Color Palette: Our analysis of representative vector clip-art images (App. A), consistent with studies of effective visualization techniques [Sto03, SASS14], suggests that artists typically employ *compact color palettes*, consisting of a small number of visually distinct colors, when creating vector clip art. Furthermore, artists typically colorize *immediately adjacent* regions using colors which are a *notable distance* apart in color space. We speculate that the mental images viewers assemble satisfy these properties. This assumption is validated by our small-scale study (App. C): when presented with anti-aliased clip art images and asked to mentally deblur them and count the number of colors in these mentally deblurred images, participants uniformly reported numbers which were either identical or slightly smaller than the color count in the originating vector images (see App. C for details).

Implications of the Anti-Aliasing Process: When rasterizing vector clip-art inputs, both aliased and anti-aliased rasterization methods keep the color of pixels that are entirely inside one of the input regions as-is. Given a pixel that intersects multiple regions, anti-aliased rasterization methods assign the pixel a color that is some weighted average of the colors of the intersected regions. Since the weights used when blurring adjacent pixels are typically different, the areas around region borders typically exhibit variation in color between adjacent pixels; hence patches of adjacent same-color pixels in anti-aliased clip art renders are likely to originate from region interiors, and thus belong to an original region of this same color. We speculate that such *uniformly-colored patches* are key to human ability to parse anti-aliased clip-art (see e.g. the uniformly colored patches in the zoomed in inputs of Figs. 3, 4). We note that only originating regions which are large enough to fully cover multiple adjacent pixels have corresponding *uniformly-colored patches* in antialiased raster images; the shade of pixels overlapping more narrow regions or region features is defined via a weighted average of their color with the colors of their surrounding regions. This suggests that originating regions become less discernible the smaller they are and the more similar their color is to that of their neighbouring regions. Conversely *outliers*, or pixels spanning small regions whose color is far away from their neighboring region colors in color space, continue to be visibly distinct even after color averaging (e.g. the eyes or cheeks of the koala in Fig. 6a). We speculate that observers mentally form regions around uniformly-colored patches and outliers, and their ability to discern outliers is dependent on the degree to which they stand out.

These observations are confirmed by our second study (App. C), in which participants were presented with anti-aliased clip-art images and were asked to trace the outlines of the regions they envision in these inputs. Participants were consistent when tracing the outlines of regions surrounding large uniformly colored patches, or when tracing the outlines of strong outliers; however, consistency diminished for renders of smaller and visually less distinct original regions. See App. C for details.

Problem Statement. Our work seeks to recover the mental images humans envision when presented with anti-aliased clip art images. We note that the above observations suggest that this task becomes increasingly harder as image resolution decreases, since the number and size of the uniformly colored patches that help anchor these images decrease. This observation motivates our focus on lower-resolution inputs, where algorithmically performing this task is likely to be most challenging. Rather than recover vector outputs from these inputs, we reconstruct raster imagery; once deblurred, our outputs can be converted into vector form using existing methods well suited for medium to high resolution aliased raster data (Figs. 1, 6). In particular, we formulate our problem as computing blur-free double resolution raster images best aligned with viewer perception; i.e for a given $n \times n$ input, we compute $2n \times 2n$ outputs. This choice allows us to capture the subpixel details viewers perceive, while explicitly avoiding generation of minuscule details that viewers are unlikely to hallucinate. Based on our observation about viewer expected color palettes, we require our outputs to have compact color palettes and aim for adjacent regions to be colorized with distinctly different colors. Following the accuracy cue, we want our outputs to be *cross resolution consistent*; that is, our outputs should be visually similar to $2n \times 2n$ aliased renders of the same original vector images, and should reliably include regions anchored by *uniformly-colored patches* and *outliers* present in the inputs. Finally, following the simplicity and smoothness cues, we look for regions with compact and visually smooth boundaries, avoiding unnecessary jaggies.

Overview. Our approach for sub-pixel deblurring is guided by the properties identified above (Fig. 7). Cross-resolution consistency suggests that we can leverage a data-driven approach in computing the outputs we seek. However, state of the art learning methods still struggle with tasks that require using a compact, but *a priori* unknown alphabet and resort to training different models for different alphabet sizes [VPB*22] or different palette sizes and image categories [RGLM21]. This approach is not suitable for our needs as we operate on inputs with very different viewer perceived palette sizes, where predicting the palette size is a major component of the problem we seek to solve. We sidestep this challenge by

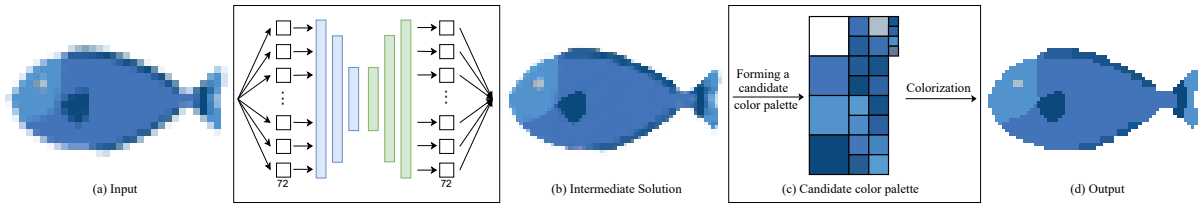


Figure 7: *Method Overview: (a) input; (b) low-blur intermediate solution; (c) candidate color palette; (d) output.*

using a two step process. We first relax the color palette constraint and focus entirely on cross-resolution consistency, obtaining outputs which are visually quite close to our desired ones but which have arbitrarily large color palettes with typically hundreds of colors (Fig. 7b). We compute our final outputs from these *low-blur* intermediate subpixel images by reintroducing the palette compactness constraints, and directly accounting for the other priors listed above. Specifically, our first step learns the intermediate low-blur images from pairs of anti-aliased and anti-aliasing free rasterizations of the same clip-art vector image, where the aliased input has double the resolution of the anti-aliased one (Sec. 4,Fig. 7b). Our second step first detects *uniformly-colored patches* and potential *outliers* in the input image by analyzing both the inputs and the intermediate solutions, and uses those to form a candidate color palette (Fig. 7c); it then uses discrete optimization to segment the intermediate images into regions corresponding to a subset of the candidate palette's colors. Our discrete optimization balances proximity to the intermediate solutions against compactness and other priors to generate the desired outputs (Sec. 5, Fig. 7d).

4. Learned Low-Blur Subpixel Approximation

Our first step learns to produce as-blur-free-as-possible subpixel approximations of the blur-free outputs we seek (Fig. 7b). We do so using a learned image-to-image translation network. We experimented with several different options when selecting our network architecture, including Real-ESRGAN [WXDS21], and SRGAN [LTH*17]. In our experiments we obtained the best results using the algorithm described below that leverages the *pix2pix* [IZZE17] architecture as its backbone; in particular, in the experiments reported in Sec 6, the average error (measured as color space distance) between our approximation step outputs and ground truth was 33% lower compared to the error of the retrained Real-ESRGAN, and the color palette size was 38% smaller. Our motivation for using *pix2pix* as our method's backbone is twofold. First, it uses the U-Net [RFB15] architecture as a generator, which leverages connections between down- and up-sampling in the network to achieve structural consistency in predictions. Second, it relies on a combination of per-pixel dissimilarity and patch-level discrimination to learn parameters, which are shown in [IZZE17] to simultaneously reduce both blurring and ringing artifacts, as compared to only reducing either blurring or ringing.

As *pix2pix* only accepts same-resolution inputs and outputs, we first upsample our $n \times n$ anti-aliased input images to a resolution of $2n \times 2n$ using nearest-neighbor upsampling (replacing each input pixel by four pixels of the same color). Our overall framework departs from the baseline formulation of [IZZE17] in a number of ways.

Color Space. To minimize perceivable color artifacts, we convert both input and output images to LAB space [FVVD*96], allowing our loss function to be computed in a space which is better aligned with human perception than standard RGB.

Gradient Prediction. In the superresolution approach of [MRC*20], it was observed that directly predicting high-resolution images using GAN-based formulations leads to structural artifacts and spatial inconsistencies. Motivated by their method, which proposes superresolving the gradient image instead and using it as guidance, we similarly predict the image gradients rather than the output pixels themselves. The differences of the three channels are computed and stored in positive and negative difference images. The training data then becomes the upsized $2n \times 2n$ anti-aliased LAB raster images and the corresponding $2n \times 2n$ positive/negative LAB difference images; we then train separate *pix2pix* network for each.

With this data and metric space, our *pix2pix* network is trained to optimize a joint objective consisting of an L1 term that forces low-frequency correctness by penalizing per-pixel discrepancies between the predicted difference image and the ground truth one, and a conditional discriminator (cGAN) that models high-frequency structure by ensuring that patches from the predicted image are statistically indistinguishable from those sampled from the ground truth image:

$$\mathcal{L} = \mathcal{L}_{cGAN} + \lambda \mathcal{L}_{L1}. \quad (1)$$

where the weight $\lambda = 100$ balances their relative contributions.

Denoising. Image generative models including ours, do not explicitly minimize the color palette size and thus tend to introduce high-frequency noise and local variation in output pixel color (Sec. 2). We reduce these artifacts by introducing a denoising procedure on top of our network outputs. In addition to directly applying the procedure above to the input low-resolution anti-aliased raster image at test time, we also apply it to the augmented versions of this image produced by rotations (at 90, 180, and 270 degrees) and flips (mimicking our training data augmentation procedure). The results are then aligned by applying corresponding inverse data transformations on each output image, and a per-pixel median value operator is applied to produce a final single output image. Given that noise artifacts do not tend to obey equivariant properties, this substantially reduces them in the final output.

Training We train separate models for each possible $n \times n$ input resolution using a dataset comprised of pairs of $n \times n$ anti-aliased

and $2n \times 2n$ aliased rasterizations of the same originating vector images. For details on rasterization, size, and composition of the dataset, see App. A

5. Sub-Pixel Segmentation

The outputs of our learning step have higher resolution details than the inputs and are dramatically less blurry (Fig. 7b); however, these images still contain blur and local color variations inconsistent with human expectations of magnified outputs. We deblur these *low-blur* intermediate images by leveraging clip-art domain priors, and formulate deblurring as a constrained labeling problem which associates each pixel in the output image I_o with one of the colors present in the input I, or low-blur I_a, images. We first narrow down the set of potential output colors by building a *candidate* color palette C^v containing a subset of the colors in these two images, Fig. 8e; we then assign one of these colors to each output pixel, Fig. 8f. When performing these computations we measure distance $D(.)$ between two colors using a modified Oklab [Ott21] color space (see App. A)

5.1. Candidate Color Palette

We build our candidate palette by using the observation that viewers use a combination of *patch* and *outlier seed* pixels to visually anchor the single-color regions in their perceived blur-free images (Sec. 3). Following this observation, we initialize our candidate color palette by detecting potential patch and outlier seeds in the input and low-blur magnified images and including their colors as candidates.

We define a patch of edge-adjacent same-color pixels in the input or low-blur images as a potential *patch seed*, S_i or S_a respectively, if this patch is at least two pixels wide in any direction. (Fig. 8c). We define a pixel in the low-blur magnified image as a potential outlier seed S_o if its color cannot be represented as a convex combination of its neighbor's colors (Fig. 8d). We define the sets of colors that correspond to these three seed types S_i, S_a, S_o as C_i^v, C_a^v, and C_o^v respectively.

Following observations about color distinctiveness, we require our color palettes to satisfy a baseline lower bound on color distance between pairs of colors:

$$D(c_i, c_j) > c_d,\ c_i, c_j \in C_o. \tag{2}$$

We set the threshold $c_d = 0.005$ such that 99.99% of differences between pairs of colors in image palettes across our training set satisfy it. For each pair of colors in $C_i^v \cup C_a^v \cup C_o^v$ which do not satisfy this requirement, we merge their respective seeds and remove the color with the smaller seed size from the palette. We set our candidate palette C^v to be this thinned out palette (Figs. 7c, 8e).

5.2. Colorization

We cast colorization as a discrete optimization problem of assigning each output image pixel p a color $c(p) \in C^v$.

Blur Characteristics. We express our expectation that output regions are anchored at one of our detected seeds by introducing the following energy term into our optimized function:

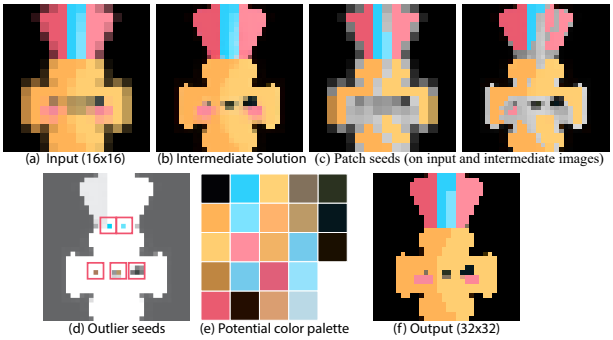


Figure 8: *Candidate Color Palette: (a) input image, (b) low-blur intermediate solution.;(c) candidate patch seeds identified in input and itermediate images, respectively; (d) candidate outlier seeds, framed in red (for clarity both sets of seeds are rendered over a greyscale version of the relevant images; (e) resulting palette. (f) Final output.*

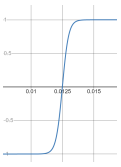
$$E_s = \sum_{p \in I_o} [(1 - n(c(p))) + L(c(p))] \tag{3}$$

$$L(c(p)) = \begin{cases} 0 & p \in S(c),\ c(p) \in C_i^v \\ 0.1 & p \in S(c),\ c(p) \in C_a^v \cup C_o^v \\ 1 & \text{otherwise} \end{cases} \tag{4}$$

where $S(c)$ are the seeds corresponding to the color c; for a color $c \in C_i^v$ originating from the input image, we define $n(c)$ as the number of pixels in its corresponding seeds divided by the input image size; for a color $c \in C_a^v \cup C_o^v$ originating from the low-blur intermediate image, we define $n(c)$ as the number of pixels in its corresponding seeds divided by the intermediate image size. The weights assigned to each color reflect the confidence in its corresponding seeds. Larger seeds, and those originating from the input rather than the intermediate image, are assigned a higher degree of trust and thus a lower cost.

Distinctiveness. We promote distinctiveness between the colors of adjacent pixels by penalizing assigning them similar, but not identical, colors:

$$E_d(I_o) = \sum_{\substack{(p_1,p_2) \in E, \\ c_o(p_1) \neq c_o(p_2)}} \left(e^{-\left(\frac{D_o(p_1,p_2)}{\sigma_d}\right)} + \left(0.5 - \frac{T(D_o(p_1,p_2))}{2}\right)\right) \tag{5}$$



where $T(u)$ is the shifted and scaled tanh function shown in the inset on the left, and $D_o(p_1,p_2)$ is the color space distance $D(c_o(p_1), c_o(p_2))$. The first term promotes distinctiveness between side-by-side pixels that do not have identical colors. We set $\sigma_d = 0.15$, chosen so that the first term drops to 0 when the the color difference nears the 90$^{\text{th}}$ percentile difference between all color pairs in the palettes of our training data images. The second term further penalizes assigning extremely similar colors to side-by-side pixels; it is defined so as to drop to near 0 when the difference in color is above that of 0.5% of adjacent colors in the training set images, and to increase to 1 when only 0.1% of such differences are below it. In setting these numbers we aim to be conservative and err on the side of preserving input details.

Cross-Resolution Consistency. We formulate the expectation that our output images should, in general, be close to the network predicted intermediate images as

$$E_a(I_o, I_a) = \sum_{p \in I_o} D(c_o(p), c_a(p)). \quad (6)$$

Simplicity. We promote simplicity, and penalize the formation of regions that viewers are unlikely to expect, via a combination of a corresponding penalty term and hard constraints:

$$E_c(I_o, I_a) = \sum_{(p_1,p_2)\in E, c_o(p_1)\neq c_o(p_2)} e^{\left(\frac{-D(c_a(p_1),c_a(p_2))^2}{\sigma_d^2}\right)}. \quad (7)$$

This term penalizes the assignment of different output colors to adjacent pixels, where the penalty reflects the difference in color between the pixels in the intermediate image. This formulation assigns lower penalties to assignments which are more consistent with the network predicted approximate solution. Note that this sum excludes same-color pairs.

Simplicity Constraints We explicitly suppress hallucination by imposing a combination of size and color difference constraints on the outputs. First, on the assumption that viewers do not hallucinate regions that are more narrow than a single pixel in the input image, we disallow *subpixel*, or 1-pixel wide, output regions. Second, our simplicity and color distinction properties suggest that viewers are unlikely to mentally form small regions whose color is very similar to one or more of their neighbors. We therefore require all small regions in our outputs to satisfy a lower bound on the color difference between them and each of their neighboring regions

$$D(c(p_1), c(p_2)) > c_a,\ p_1 \in r_s, (p_1, p_2) \in E. \quad (8)$$

Here $r_s \in R$ are all regions in the output whose size is less than $n/3$. Less than 10% of regions in our ground truth data fall below this threshold; we refer to such regions as *indistinct*.

Combined Energy Function. Our energy function combines all terms outlined above:

$$E(I_o) = E_s(I_o) + w_b(E_d(I_o) + E_c(I_o, I_a)) + w_a E_a(I_o, I_a). \quad (9)$$

We set $w_a = 10$ and $w_b = 0.5$, prioritizing pixel-level consistency with the approximate magnification output over all other considerations and weakly promoting seed anchoring.

5.2.1. Optimization

Minimizing the assignment energy $E(I_o)$ while enforcing the simplicity constraints requires solving an NP-hard problem with no standard solution mechanisms. We obtain a suitable approximate solution that satisfies all constraints above by first minimizing the assignment energy without enforcing the constraints, and then enforcing simplicity by modifying the output colorization.

Unconstrained Optimization. We obtain the color-to-pixel assignment that minimizes $E(I_o)$ using a classical graph-cut framework [BK04]. By promoting simplicity (Eq. 7) we preferentiate outputs with compact region boundaries, and implicitly minimize the number of colors used in the output colorization (Tab. 1). Our output palettes are typically about half the size of the candidate ones.

Satisfying Simplicity. We identify and remove all regions in our unconstrained optimization output that violate our simplicity constraints, using a bottom-up approach that merges such regions with their neighbors while keeping the overall energy $E(I_o)$ as small as possible. We remove sub-pixel regions first, and then indistinct ones. For each pair of regions we consider merging, we compute $E(I_o)$ before and after the merging step; we then perform the merging operation which increases $E(I_o)$ the least. We shorten unnecessarily long region boundaries sometimes introduced by this step as discussed in App. A.

6. Results and Validation

We tested our method on 112 diverse, previously unseen anti-aliased raster inputs generated from 77 different vector images: 12 inputs at 16×16, 43 inputs at 32×32, 37 inputs at 64×64, and 17 inputs at 128×128. These include depictions of organic (e.g. Fig. 6) and synthetic (e.g. Fig. 4) content, including both simple inputs (e.g. Fig. 5) and ones with intricate details (e.g. Fig. 1); see App. B.1 for details. Visual inspection, and qualitative and quantitative assessments described below, confirm that our results are consistently well aligned with viewer expectations. Our evaluation focuses on inputs with resolutions of up to $100 \times 100px$ since at higher resolutions anti-aliasing is highly localized which in turn makes color palette and region topology extraction much easier. We include $128 \times 128px$ inputs to verify that our method continues to be viable at larger resolutions; at this resolution our method continues to achieve state of the art performance (Fig. 15). To enable quantitative evaluation, all the inputs tested were generated by rasterizing vector images (using the default rasterization framework in Adobe Illustrator, see App. A for details)). The color palettes of the vector inputs we used range in size from 2 to 172 colors, with the median palette size being 8 colors. Throughout the paper we show our results on 30 inputs whose originating vector art color palettes range in size from 5 colors (Fig. 5) to 114 (Fig. 1), additional results are included in the supplementary. In addition to visual inspection, we evaluate our method by comparing our results to those generated using algorithmic alternatives, and demonstrate the usability of our outputs for a range of applications.

Visual Comparison to Prior Work. We compare our results to those produced by a range of prior methods in Figs. 1, 4-6. Figs. 9, 10 and 11 include additional comparisons between our method and the most competitive prior work, including Adobe Lightroom [Ado21], the recent neural superresolution method Real-ESRGAN [WXDS21] retrained and fine-tuned on our data, and the commercial VectorMagic software [Vec17]. For the latter, to provide an apples-to-apples comparison, we either vectorize our outputs using VectorMagic enabling vector-to-vector comparison (Fig. 1,11) or show both their vector outputs and the blur-free rasterizations of these outputs at our output resolution enabling raster-space comparison to our outputs (Figs. 9, 10). Results are illustrated on inputs of resolutions of $n = 16, 32$ and 64 (inputs whose sizes are not powers of two are padded using their background color to bring them to the nearest power of two). Additional comparisons are shown in the supplementary material.

As these comparisons show, the vast majority of prior methods potentially applicable to the problem we address fail to remove

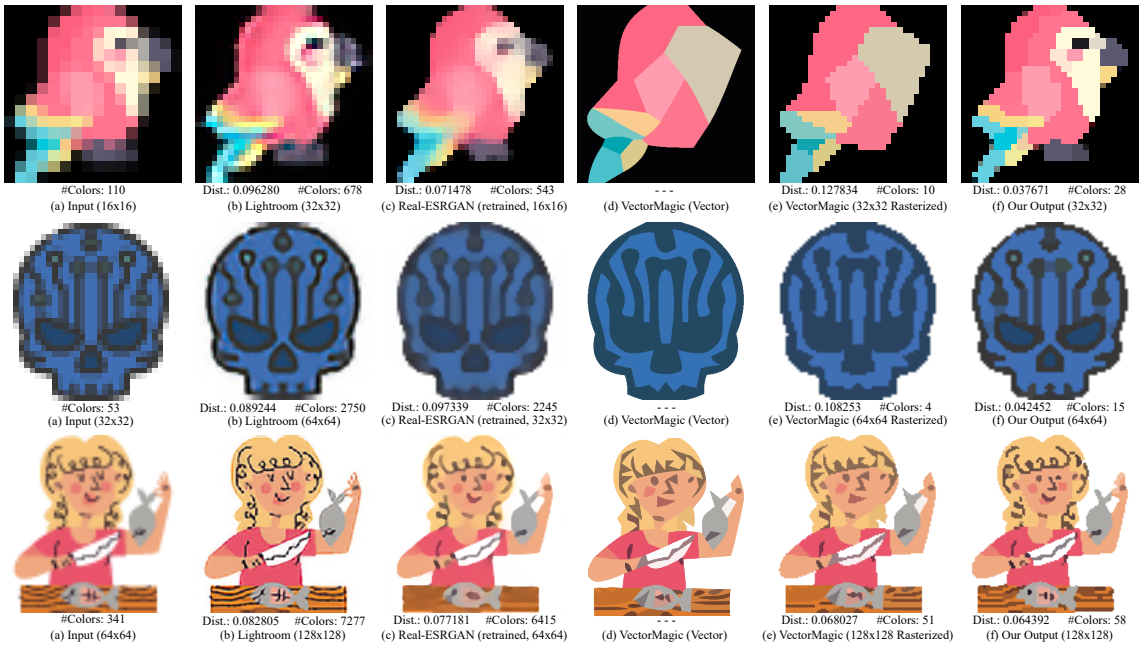


Figure 9: *Our results (f) and those created by Adobe Photoshop Lightroom [Ado21] (b), retrained Real-ESRGAN [WXDS21] (c), and VectorMagic [Vec17] (vector and rasterized) (d and e), on inputs of varying resolutions (a). Captions report the color counts in each image and the distance in RGB space between them and the corresponding GT images.*

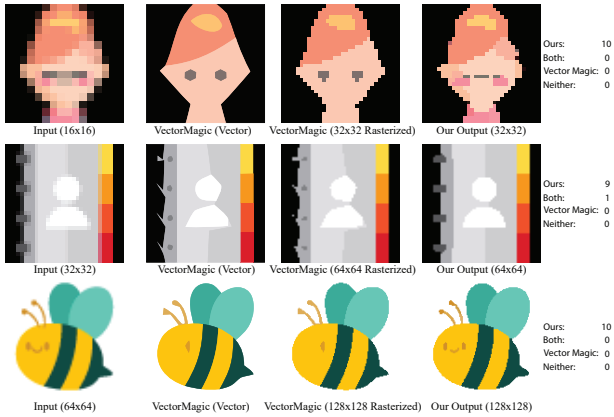


Figure 10: *Raster space comparisons with VectorMagic [Vec17] across varying resolutions. Left to right: input, VectorMagic output, VectorMagic output rasterized, our output. Column on left reports comparative study preference summary for each input ((c) vs (d)).*

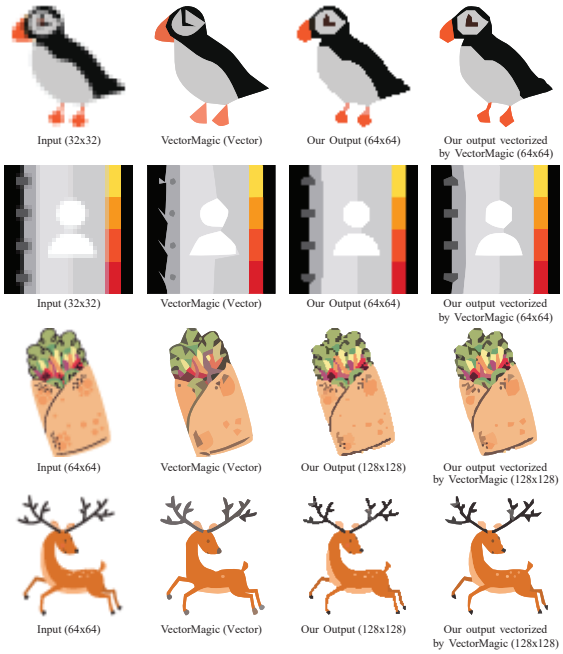


Figure 11: *Vector space comparisons with VectorMagic [Vec17]. Left to right: input, VectorMagic output, our output, our output vectorized with VectorMagic.*

anti-aliasing blur and often generate outputs which are even blurrier than the inputs. Our subpixel deblurring approach produces much crisper blur-free outputs, with drastically smaller color palettes. VectorMagic successfully reduces blur, but often loses fine grained details that our method successfully retains (e.g. the smile of the girl in Fig. 10, or the right arm of the snowman in Fig. 1)

Quantitative Comparison. Cross-resolution consistency suggests that our desired outputs should be similar, if not identical, to double resolution aliased rasterizations of our inputs' originating clip-art images (GT). We measure the degree of similarity between these GT images, and our and alternative results, using both RGB and Oklab color space distance (Figs. 5, 9). Tab. 1 reports these numbers for our method as well as the results of the three closest best methods: [Ado21], [Vec17] and [WXDS21] (retrained on our

	16px		32px		64px		all inputs	
	RGB & Oklab dist.	♯color	RGB & Oklab dist.	♯color	RGB & Oklab dist.	♯color	RGB & Oklab dist.	♯color
Antialiased input	0.1026 / 0.0449	67.5	0.0639 / 0.0248	102.4	0.0407 / 0.0159	167.2	0.0606 / 0.0244	122.8
Lightroom [Ado21]	0.0951 / 0.0520	640.5	0.0583 / 0.0274	1715	0.0419 / 0.0196	3888	0.0575 / 0.0281	2413
Real-ESRGAN [WXDS21]	0.0635 / 0.0283	525.3	0.0466 / 0.0191	1249	0.0272 / 0.0114	2552	0.0415 / 0.0174	1655
VectorMagic [Vec17]	0.0934 / 0.0378	8.5	0.0536 / 0.0203	9.3	0.0296 / 0.0111	13.7	0.0502 / 0.0193	11
Ours (intermediate solution)	0.0393 / 0.0208	329.3	0.0302 / 0.0128	751.1	0.0200 / 0.0088	1608	0.0275 / 0.0124	1027
Ours (final output)	**0.0334 / 0.0137**	**18.2**	**0.0250 / 0.0091**	**11.9**	**0.0158 / 0.0059**	**17.1**	**0.0226 / 0.0085**	**14.9**
Originating vector ♯colors		7.9		8.1		14.1		10.4

Table 1: *Quantitative comparisons of our outputs against the three closest competitors. Measurements include color space difference (L^1 norm) in RGB and Oklab between the results of each method and the corresponding GT image (aliased, double-resolution rasterization of the input's originating vector image) and color palette sizes. Our outputs are closer to GT in color space than those produced by the alternatives and have comparable palette sizes to the originating vector images. (Numbers computed across all 78 low-resolution inputs in comparison_ESRGAN_Lightroom_VectorMagic_ours.pdf in the supplementary material.)*

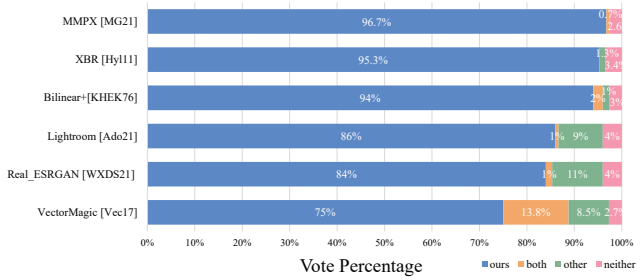


Figure 12: *Summary of user preferences in our comparative study on low resolution (100px or less) data. Our method is preferred by a significant margin over all alternatives.*

data). Numbers for other methods that we compare against are reported in the supplementary. As the numbers show, the color space distance we obtain is roughly half of that measured for other methods, with the difference most pronounced at lower resolutions. We also compare the sizes of the different color palettes. While our final palette size is very similar to that of GT data, the palettes produced by Real-ESRGAN are two orders of magnitude larger, making their results unsuitable for any of the downstream applications we target. Consistent with visual inspection, VectorMagic palettes are more similar size wise to the GT ones; however, VectorMagic outputs are a lot less faithful in terms of color space distance (0.0226 us vs 0.0502 VectorMegic).

For ablation purposes we report the same measurements for our intermediate low-blur solutions. We note that the distance from these intermediate solutions to GT is lower than that of [Ado21], [WXDS21] and [Vec17] by a significant margin (52% lower than AdobeLightroom, 34% lower than Real-ESRGAN and 45% lower than VectorMagic in RGB color space). At the same time, the number of colors in these intermediate solutions is two orders of magnitude higher than the corresponding GT color counts. Our final outputs have both 18% smaller color space distances to GT and two orders of magnitude smaller color counts.

Qualitative Comparison As stated above, our goal is to produce outputs consistent with viewer expectations. To evaluate how viewers perceive our outputs compared to those produced by alternative methods, we performed a comparative perceptual study (App. C). Study participants were shown input anti-aliased images (on top), together with our result and an alternative result (below) and were asked to "Mentally deblur and magnify the anti-aliased raster image on the top (A). Which of the images on the bottom (B or C) comes closest to the blur-free image you mentally assembled?" The answer options were "B", "C", "Both", and "Neither".

In our study, we compared our results to those produced by representative methods for vectorization [Vec17], superresolution [WXDS21], upsampling [Ado21], magnificationXBR, McGuire2021PixelArt, and deblurring [KHEK76]. We included 40 queries for VectorMagic, which showed the closest performance to ours: 10 queries at $16\times16px$, 15 at $32\times32px$, and 15 at $64\times64px$. For all other methods we included 15 queries each: 10 at $32\times32px$, and 5 at $64\times64px$. Our choice of split between resolutions was motivated by our focus on lower image resolutions, where we expect our method to be most impactful. We collected answers for each query from ten different participants; each participant answered 20-25 questions, organized so that no participant saw the same input twice.

Our study results are summarized in Fig. 12. Our method outperforms all baseline methods by a factor of 9 to 1 or more; compared to the closest best performing alternative [Vec17], participants preferred our results 75% of the time, judged them as on par 14% of the time, and preferred the alternative only 8.5% of the time. A plurality of participants preferred our outputs in *all* comparisons against the methods of [Ado21, Hyl11, MG21] and [KHEK76]. In comparisons against [WXDS21] a plurality preferred our outputs on 14 inputs, and judged them on par on one input. In a comparison against [Vec17] a plurality of participants preferred our results on 35 inputs, judged the results as on par on 4, and preferred the alternative on 1 input (Fig. 13). See supplementary for complete results.

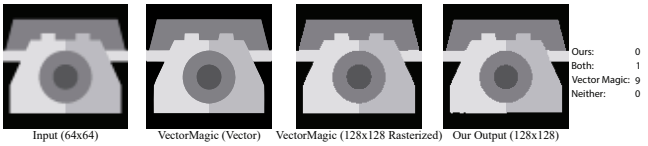


Figure 13: *The only under 100px study input for which a plurality of participants preferred the output of VectorMagic over our output.*

Applications. While our subpixel deblurred outputs are extremely useful on their own for applications where double-resolution deblurred versions of legacy images are needed (for instance, image retargeting for mobile devices [ABA*16], or remastering legacy content for video games), our outputs can also be used as input to a number of downstream applications. As Figs 1 and

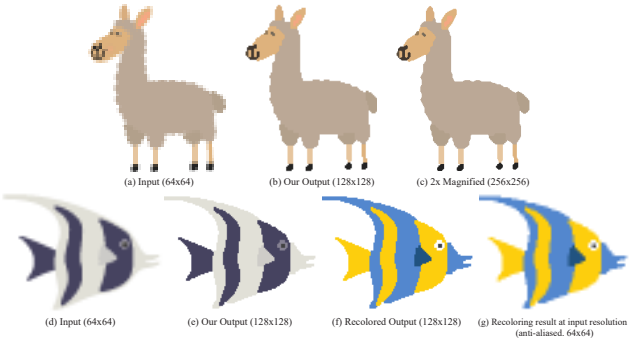


Figure 14: *Examples of subpixel deblurring applications. Top: our outputs (b) can be further magnified to user desired resolution (c, here done using [Hyl11]). Bottom: our deblurred outputs (e) can be recolored with just a few mouse clicks (f) and can then be downsized to input size using anti-aliased rasterization, obtaining recolored versions of the inputs (g).*

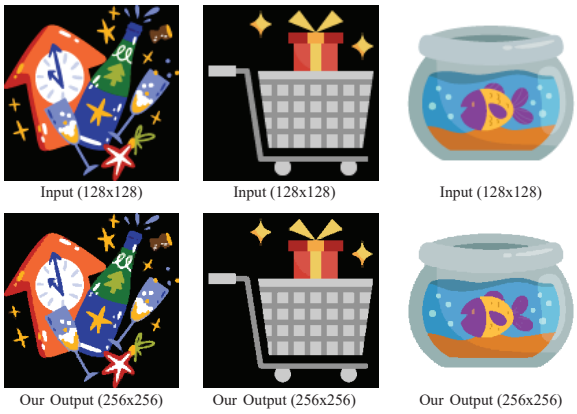


Figure 15: *Our results on 128x128px inputs.*

11 show, our method can be used as a stepping stone towards better quality vectorization of the inputs. Users can similarly use our outputs as an intermediate step for generating even higher resolution deblured outputs by magnifying them using one of the magnification methods discussed above, such as [Hyl11] (Fig. 14,top). Finally, as demonstrated in Figs. 1 and 14,bottom, our outputs facilitate easy in-place recoloring of the input images. In particular, while recoloring anti-aliased images requires manually adjusting the color of individual blurred pixels along region borders, a nearly impossible task, recoloring our outputs is a matter of a few flood-fill clicks (Fig. 14f). Once recolored, the image can be resized back to the input resolution using anti-aliased downsampling to obtain the desired results (Fig. 14g).

Higher-Resolution Image Deblurring. Our method targets lower resolution inputs (100px or less), since these are known to be the most challenging ones to process [HDS*18, DSG*20, KL11]. For completeness, we qualitatively and quantitatively evaluate it on larger (128x128px) inputs. To this end, in the perceptual study above we included 30 comparisons of our results generated from 14 input vector clip-art images at 128x128px: 5 against each of [Vec17, WXDS21, Ado21, Hyl11, MG21] and [KHEK76]. The results of the study are summarized in Fig. 16. At an input resolution of 128px our method was judged as significantly better than five out of the six alternatives; against VectorMagic participants judged our results as better or on par 52% of the time and preferred VectorMagic 44% of the time; participants selected "neither" 4% of the time. In terms of quantitative similarity to GT, our approach is closer in color space to GT than all alternatives, producing RGB distance of 0.0109 compared to 0.0159 for VectorMagic, and 0.0197 for Real-ESRGAN. Similar trends hold for measurements in Oklab space. In terms of the palette size, our results (42 colors on average) are also close to the GT (21.5 colors on average), but higher than the 22 obtained by VectorMagic. This difference likely explains the increase in viewer preference for VectorMagic at this resolution.

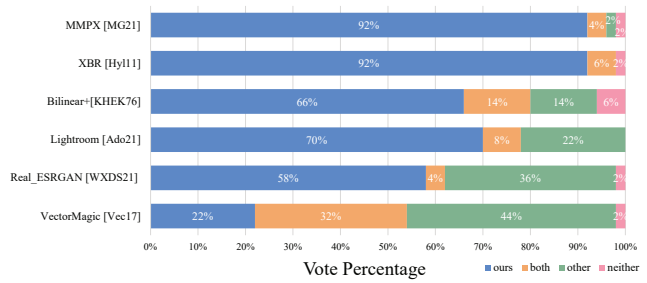


Figure 16: *Summary of user preferences on high resolution (128px) data. Our method significantly outperforms five of the alternatives, and performs on par with the best performing one.*

7. Conclusion

We presented a novel method for subpixel deblurring of low-resolution anti-aliased raster clip art images and demonstrated it to significantly outperform prior art. Key to our method is a two step approach that combines data driven learning of approximate subpixel deblurring with a discrete optimization process guided by perceptual and domain priors that further improves debluring quality and drastically compacts the output color palette.

Our work suggests a few avenues for future research. First, we employ only minimal regularization (App. A) to produce our outputs. As highlighted by prior work [DSG*20] human observers mentally regularize input images using cues such as symmetry or parallelism. One can easily apply these principles to further regularize our outputs, and our work might benefit from enforcing additional cues identified by clip- and pixel-art vectorization research [HDS*18, DSG*20, KL11] such as connectedness and symmetry. (We hypothesize that a lack of regularization is why viewers prefer the output of VectorMagic over our result in Fig. 13.) Our method is heavily reliant on color space metrics that aim to assess how visually distinct pairs of adjacent output regions are, and improving upon existing metrics is also an interesting and important topic for future research. Lastly performing end-to-end subpixel debluring within a learning framework is an intriguing future problem.

Acknowledgements. The authors wish to thank Chrystiano Araujo for his assistance in running experiments and comparisons.

J. Yang was supported by an NVIDIA research internship. S. Kheradmand was supported by an National Science and Engineering Research Council of Canada (NSERC) CGS-D scholarship. L. Sigal is supported by a Canadian Institute for Advanced Research AI Chair at the Vector AI Institute in Toronto, and an NSERC Canada Research Chair and Discovery Grant. A. Sheffer is supported by NSERC grant RGPIN-2018-03944, and a research grant from Adobe.

References

[ABA*16] ARTUSI A., BANTERLE F., AYDIN T., PANOZZO D., SORKINE-HORNUNG O.: *Image Content Retargeting: Maintaining Color, Tone, and Spatial Consistency*. CRC Press, 2016. 10

[Ado17] ADOBE: Adobe Illustrator 2017: Image Trace. http://www.adobe.com/, 2017. 4, 13

[Ado21] ADOBE: Adobe Photoshop Lightroom. Super Resolution. https://www.adobe.com/products/photoshop-lightroom/super-resolution.html, 2021. 4, 8, 9, 10, 11

[BK04] BOYKOV Y., KOLMOGOROV V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE PAMI 26*, 9 (2004), 1124–1137. 8

[Die08] DIEBEL J. R.: *Bayesian image vectorization: The probabilistic inversion of vector image rasterization*. Ph.D. dissertation, Stanford Univ., 2008. 4

[DLHT15] DONG C., LOY C. C., HE K., TANG X.: Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2 (2015), 295–307. 4

[DSG*20] DOMINICI E., SCHERTLER N., GRIFFIN J., HOSHYARI S., SIGAL L., SHEFFER A.: Polyfit: Perception-aligned vectorization of raster clip-art via intermediate polygonal fitting. *ACM Transaction on Graphics 39* (2020). 2, 4, 11

[FH04] FELZENSZWALB P. F., HUTTENLOCHER D. P.: Efficient graph-based image segmentation. *International Journal of Computer Vision 59*, 2 (2004), 167–181. 3

[FLB16] FAVREAU J.-D., LAFARGE F., BOUSSEAU A.: Fidelity vs. simplicity: a global approach to line drawing vectorization. *ACM SIGGRAPH* (2016). 4

[FSH*06] FERGUS R., SINGH B., HERTZMANN A., ROWEIS S. T., FREEMAN W. T.: Removing camera shake from a single photograph. *ACM TOG 25*, 3 (2006), 787–794. 3

[FVVD*96] FOLEY J. D., VAN F. D., VAN DAM A., FEINER S. K., HUGHES J. F., HUGHES J.: *Computer graphics: principles and practice*, vol. 12110. Addison-Wesley Professional, 1996. 6, 14, 15

[HDS*18] HOSHYARI S., DOMINICI E., SHEFFER A., CARR N., CEYLAN D., WANG Z., SHEN I.-C.: Perception-driven semi-structured boundary vectorization. *ACM Transaction on Graphics 37*, 4 (2018). doi:10.1145/3197517.3201312. 2, 4, 5, 11

[HEK21] HETTINGA G. J., ECHEVARRIA J., KOSINKA J.: Efficient Image Vectorisation Using Mesh Colours. In *Smart Tools and Apps for Graphics - Eurographics Italian Chapter Conference* (2021), Frosini P., Giorgi D., Melzi S., Rodolà E., (Eds.), The Eurographics Association. doi:10.2312/stag.20211484. 4

[Hyl11] HYLLIAN: Xbr. https://github.com/Hyllian/glsl-shaders/blob/master/xbr/shaders/xbr-lv2.glsl, 2011. 3, 4, 10, 11

[Ink20] INKSCAPE: Inkscape, 2020. URL: https://inkscape.org. 4

[IZZE17] ISOLA P., ZHU J.-Y., ZHOU T., EFROS A. A.: Image-to-image translation with conditional adversarial networks. *CVPR* (2017). 2, 6, 13

[KB14] KINGMA D. P., BA J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014). 14

[KHEK76] KUWAHARA M., HACHIMURA K., EIHO S., KINOSHITA M.: Processing of ri-angiocardiographic images. In *Digital processing of biomedical images*. Springer, 1976, pp. 187–202. 3, 10, 11

[KKD09] KYPRIANIDIS J. E., KANG H., DÖLLNER J.: Image and video abstraction by anisotropic kuwahara filtering. *Computer Graphics Forum 28*, 7 (2009), 1955–1963. Special issue on Pacific Graphics 2009. doi:10.1111/j.1467-8659.2009.01574.x. 3

[KKT20] KIM W., KANEZAKI A., TANAKA M.: Unsupervised learning of image segmentation based on differentiable feature clustering. *IEEE Transactions on Image Processing* (2020). 3

[KL11] KOPF J., LISCHINSKI D.: Depixelizing pixel art. *ACM TOG 30*, 4 (2011), 99:1–99:8. 4, 11

[Kof55] KOFFKA K.: *Principles of Gestalt Psychology*. International library of psychology, philosophy, and scientific method. Routledge & K. Paul, 1955. 2

[LCS*21] LIANG J., CAO J., SUN G., ZHANG K., GOOL L. V., TIMOFTE R.: SwinIR: Image restoration using swin transformer, 2021. arXiv:2108.10257. 4

[LL06] LECOT G., LEVY B.: Ardeco: Automatic Region Detection and Conversion. In *EGSR* (2006), pp. 349–360. 3, 4

[LSZ*21] LIANG J., SUN G., ZHANG K., VAN GOOL L., TIMOFTE R.: Mutual affine network for spatially variant kernel estimation in blind image super-resolution. In *IEEE International Conference on Computer Vision* (2021). 4

[LTH*17] LEDIG C., THEIS L., HUSZAR F., CABALLERO J., CUNNINGHAM A., ACOSTA A., AITKEN A., TEJANI A., TOTZ J., WANG Z., SHI W.: Photo-realistic single image super-resolution using a generative adversarial network. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)* (July 2017). 6

[LW16] LI C., WAND M.: Precomputed real-time texture synthesis with markovian generative adversarial networks. *ECCV* (2016). 13, 14

[LWDF09] LEVIN A., WEISS Y., DURAND F., FREEMAN W. T.: Understanding and evaluating blind deconvolution algorithms. *CVPR* (2009). 3

[MG21] MCGUIRE M., GAGIU M.: MMPX style-preserving pixel art magnification. *Journal of Graphics Techniques* (January 2021), 36. Journal of Graphics Techniques. 3, 4, 10, 11

[MRC*20] MA C., RAO Y., CHENG Y., CHEN C., LU J., ZHOU J.: Structure-preserving super resolution with gradient guidance, 2020. arXiv:2003.13081. 6

[OBW*08] ORZAN A., BOUSSEAU A., WINNEMÖLLER H., BARLA P., THOLLOT J., SALESIN D.: Diffusion curves: A vector representation for smooth-shaded images. *ACM TOG 27*, 3 (2008). 3, 4

[Ott21] OTTOSSON: Oklab. https://bottosson.github.io/posts/oklab/, 2021. 7, 14

[RFB15] RONNEBERGER O., FISCHER P., BROX T.: U-net: Convolutional networks for biomedical image segmentation. *MICCAI* (2015). 6

[RGLM21] REDDY P., GHARBI M., LUKAC M., MITRA N. J.: Im2Vec: Synthesizing vector graphics without vector supervision. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Los Alamitos, CA, USA, jun 2021), IEEE Computer Society, pp. 7338–7347. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR46437.2021.00726, doi:10.1109/CVPR46437.2021.00726. 2, 4, 5

[SASS14] STONE M., ALBERS SZAFIR D., SETLUR V.: An engineering model for color discriminability as a function of size. In *IS&T 22nd Color Imaging Conference* (2014). 2, 5, 14

[SBv05] SÝKORA D., BURIÁNEK J., ŽÁRA J.: Sketching cartoons by example. In *Proc. Sketch-Based Interfaces and Modeling* (2005), pp. 27–34. 4

[SLWS07] SUN J., LIANG L., WEN F., SHUM H.-Y.: Image vectorization using optimized gradient meshes. In *ACM SIGGRAPH* (2007). 3, 4

[Ste03] STEPIN M.: HQX. http://web.archive.org/web/20070717064839/www.hiend3d.com/hq4x.html, 2003. 3

[Sto03] STONE M. C.: *A Field Guide to Digital Color*. CRC Press, 2003. 2, 5

[TFCRS11] THOMPSON W., FLEMING R., CREEM-REGEHR S., STEFANUCCI J. K.: *Visual Perception from a Computer Graphics Perspective*, 1st ed. A. K. Peters, Ltd., USA, 2011. 2, 4

[TM98] TOMASI C., MANDUCHI R.: Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)* (1998), pp. 839–846. doi:10.1109/ICCV.1998.710815. 3

[Vec17] VECTOR MAGIC:. Cedar Lake Ventures http://vectormagic.com/, 2017. 1, 4, 8, 9, 10, 11

[VPB*22] VINKER Y., PAJOUHESHGAR E., BO J. Y., BACHMANN R. C., BERMANO A. H., COHEN-OR D., ZAMIR A., SHAMIR A.: Clipasso: Semantically-aware object sketching. *ACM Trans. Graph. 41*, 4 (2022). 5

[WEK*12] WAGEMANS J., ELDER J. H., KUBOVY M., PALMER S. E., PETERSON M. A., SINGH M., VON DER HEYDT R.: A century of gestalt psychology in visual perception i. perceptual grouping and figure-ground organization. *Psychological Bulletin 138*, 6 (2012), 1172–1217. 2, 4

[WXDS21] WANG X., XIE L., DONG C., SHAN Y.: Real-ESRGAN: Training real-world blind super-resolution with pure synthetic data. In *International Conference on Computer Vision Workshops (ICCVW)* (2021). 1, 2, 4, 6, 8, 9, 10, 11

[WZGW17] WANG C., ZHU J., GUO Y., WANG W.: Video vectorization via tetrahedral remeshing. *IEEE TIP 26*, 4 (April 2017), 1833–1844. 3, 4

[XK17] XIA X., KULIS B.: W-net: A deep model for fully unsupervised image segmentation. *arXiv: 1711.08506* (2017). 3

[XLXJ11] XU L., LU C., XU Y., JIA J.: Image smoothing via l0 gradient minimization. In *Proceedings of the 2011 SIGGRAPH Asia conference* (2011), pp. 1–12. 3

[XLY09] XIA T., LIAO B., YU Y.: Patch-based image vectorization with automatic curvilinear feature alignment. *ACM TOG 28*, 5 (2009). 3, 4

[XSTN14] XIE G., SUN X., TONG X., NOWROUZEZAHRAI D.: Hierarchical diffusion curves for accurate automatic image vectorization. *ACM Trans. Graph. 33*, 6 (2014), 230:1–230:11. 3, 4

[YCZ*16] YANG M., CHAO H., ZHANG C., GUO J., YUAN L., SUN J.: Effective clipart image vectorization through direct optimization of bezigons. *IEEE TVCG 22*, 2 (2016), 1063–1075. 4

[ZCZ*09] ZHANG S.-H., CHEN T., ZHANG Y.-F., HU S.-M., MARTIN R. R.: Vectorizing cartoon animations. *IEEE TVCG 15*, 4 (2009), 618–629. 4

[ZLVGT21] ZHANG K., LIANG J., VAN GOOL L., TIMOFTE R.: Designing a practical degradation model for deep blind image super-resolution. In *IEEE International Conference on Computer Vision* (2021). 4

Appendix A: Implementation Details

Dataset.

We assemble a dataset of 145 vector images by collecting vector clip-art from online repositories. These consist of a variety of complex shapes comprising single or multiple objects; different color palette sizes and different complexity (some with as few as 4 regions, others with over a hundred). We split this dataset into disjoint train/test subsets, with training composed of 68 and testing of 77 images.

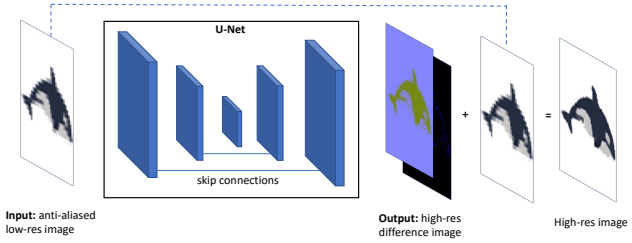


Figure 17: *Our U-Net architecture adopted from [IZZE17].*

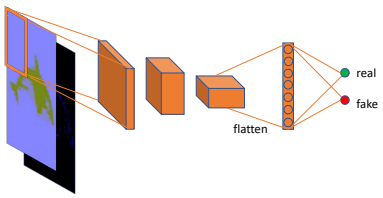


Figure 18: *Our patch discriminator adopted from [LW16].*

We rasterize vector images at different $n \times n$ resolutions in [Ado17] using the supersampling anti-aliasing setting designed for artwork (we explore the font hinted setting in Appendix B as an ablation). Additionally, we rasterize each input at double resolution $2n \times 2n$ with no anti-aliasing. Following the cross-resolution consistency principle, on training data we use these double-resolution inputs as ground truth, and use them for quantitative evaluation for test data.

Data Augmentation. In order to augment the training data, we transform each image pair by rotations, reflections and switching of RGB color channels. As a result, each distinct image in our training dataset has 72 variations in training data. Each distinct test image also has 72 variations, which we use for denoising pix2pix outputs (as described in more detail in Section 4 of our paper).

Preprocessing. We add two rows of background colored pixels around each input prior to processing; we define the background color as the most common color along the image perimeter. This process makes the background a single region. We remove this padding from the final inputs. In our experiments adding padding improved the performance of both steps of our method.

Architecture Details.

We inherit architectural design and corresponding inductive biases from pix2pix. This includes U-Net architecture for the generator (see Fig. 17), that preserves pixel correspondence and locality, and a default patch discriminator (see Fig. 18) with an additional L1 loss. Our own inductive biases for architecture design focused on perceptual color space (LAB) for the loss function computation and

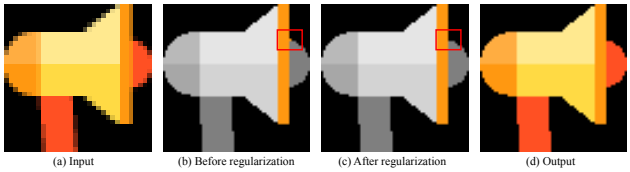


Figure 19: *Our regularization step removes redundant pixel-wide protrusions along region boundaries (b,c).*

gradient prediction (as opposed to direct prediction) of high resolution content (illustrated in both Figs. 17 and 18).

Training Details.

We train separate models for different resolutions setting n equal to $16px$, $32px$, $64px$ and $128px$. In order to upsample $n \times n$ anti-aliased images to the pix2pix input resolution of $2n \times 2n$ we use nearest-neighbour upsampling, copying one pixel in the $n \times n$ input image to four pixels in the $2n \times 2n$ counterpart. This simple upsampling is motivated by our desire to keep the original anti-aliasing and not to introduce additional interpolations into the input. If needed, such interpolations can be learned by the pix2pix network itself.

We use a ResNet backbone, with 6 residual blocks, for *pix2pix* itself (the predictor) and leverage PatchGAN [LW16] as a discriminator. We optimize networks for 300 epochs with batch size of 16 using the Adam optimizer [KB14]. We tune the learning rate for each resolution, as it is resolution dependent, and employ a learning schedule where this rate is fixed for the first 150 epochs and then linearly reduced to 0 over the remaining 150.

Color Space Distances

Measuring color-space distances in a manner consistent with human perception remains an open problem [SASS14]. In our colorization step we use a combination of established space metrics and heuristics based on observations of our training data. Specifically, unless stated otherwise, we use OkLab [Ott21] distance for all measurements. We overcome minute variations in pixel color by defining two colors as *the same* if the distance between them in RGB space is less than or equal to $\|(2,2,2)\|$. While in our experiments OkLab distances are generally well aligned with viewer perception for colors which are farther apart, we found them too sensitive for dark colors. Accordingly, if two colors both have RGB space values between $(0,0,0)$ and $(20,20,20)$ and the RGB space norm of their difference is below $\|(20,20,20\|$, we set the distance between them to zero.

We define pixels as outliers if their color is at least $\varepsilon = \|(5,5,5)\|$ apart from the closest affine combination of its neighbor colors in RGB space.

Boundary Regularization

As noted in Sec. 5.2, our simplicity enforcement step removes non-simple regions but can undesirably elongate region boundaries, and can in particular introduce single-pixel-wide protrusions. Since viewers are unlikely to hallucinate such protrusions, we seek to remove them by merging them with a neighboring region. We identify protrusions which are one pixel wide and two or more pixels long, ignoring ones which are part of constant slope lines. We merge the protrusion with the neighboring region of the most similar color if doing so does not introduce longer protrusions.

16×16 Inputs

Extremely low resolutions pose unique challenges, both for learning low-blur magnifications and for detecting patch seeds. The first challenge arises since the pix2pix network uses a kernel size of 3x3 and a fixed number of kernels per residual block. As an artifact the receptive field is fairly large and with extremely low resolution inputs the convolutional nature of the operations is effectively lost. We address this challenge when training our network on $16px$ data by first magnifying our inputs using nearest neighbor sampling to $32px$ and our outputs to $64px$ accordingly. At run-time, after running our approach we then sub-sample the $64px$ outputs back to $32px$ to produce the final result by using the median of each block of 4 neighboring pixels. Aside from this input/output magnification, we used the same data augmentation process, and train the network with the same hyperparameters as for other resolutions.

Using our default patch seed detection on $16px$ inputs is similarly problematic, as the number of pixels occupied by original regions drops dramatically (Fig. 10, top); keeping our default criterion leads to a loss of information encoded in long one-pixel wide regions (e.g. the princess's eyes in Fig. 10). Accordingly, for $16px$ inputs we redefine the patch seeds to include all pairs of adjacent same-color pixels. The rest of the processing remains the same.

Runtimes

Our training times are resolution dependent. Training the $16px$ and $32px$ networks took around 2.5 hours; training the $64px$ network took around 6 hours, and training the $128px$ network took around 24 hours. Our models were trained on a GeForce RTX 2080.

Our method's run-time is dominated by the coloring step (Sec. 5.2). Our median run-times are 0.6 seconds for $16px$ inputs, 3.5 seconds for $32px$ inputs, 33 seconds for $64px$ inputs, and 6.8 minutes for $128px$ inputs. Timings were measured on a Intel Core I7-8700k running at 3.70GHz with 32GB of system memory.

Appendix B: Ablations

Invariance to Rasterization For the experiments in the paper so far, we used inputs rasterized using standard supersampling based anti-aliasing; supersampling is the default method used for rasterizing clip-art images [FVVD*96]. Our blur-free magnification technique does not, however, assume any specific rasterization scheme and can adapt to differences in rasterization within pix2pix learning. To illustrate this, we also conduct experiments with font hinting as the rasterization mode. The results can be seen in Figure 20 and were produced with no *pix2pix* retraining. Despite the clear differences in the inputs induced by the two different rasterization techniques, our approach successfully produces anti-aliasing free outputs that are sharp and structurally consistent in both cases.

Figure 20: *Results on inputs rasterized using different schemes: (a) input produced using supersampling based anti-aliasing (b) input produced using font hinting based anti-aliasing (c) output for (a); (d) output for (b).*

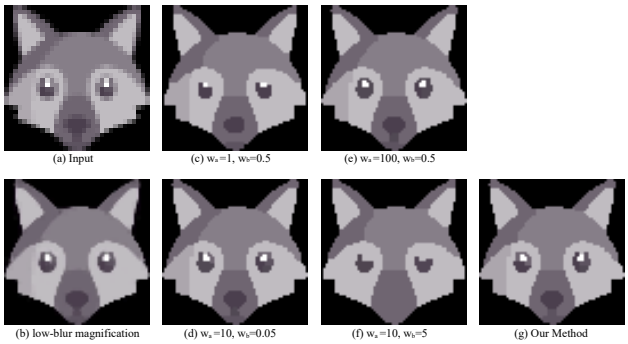


Figure 21: *Results of increasing and decreasing the weights w_a, w_b by a factor 10.*

Colorization Energy Our colorization energy $E(I_o)$ combines four terms measuring color distinction E_d, compactness E_C, cross-resolution consistency E_a, and seed anchoring E_s; we use the weights $w_a = 10$ and $w_b = 0.5$ to balance these terms. In our experiment (Fig. 21) we increased or decreased each of these weights by a factor of 10. Decreasing w_a or increasing w_c increases the importance of the compactness term, decreasing the number of output regions, while the inverse changes results in the preservation of redundant details. Our output balances the conflicting cues in a manner consistent with viewer expectations.

Impact of Color Space Our pix2pix network is trained using the LAB color space [FVVD*96]. Fig. 22 compares our results to those produced using a network trained in RGB space. The differences, while minor overall, can impact the recovery of fine details when the color difference between fine details and adjacent regions is not very large.

Comparison vs Real-ESRGAN. Fig. 23 shows the impact of replacing our first step, based on pix2pix, with nearest-neighbour upsampling and then deblurring based on the Real-ESRGAN model, retrained on our inputs. As the image shows, the blur and large color variation in their outputs means that we are no longer to reliably detect outliers and patches in the outputs of the learning step. Consequently, our palette computation is unable to extract a meaningful palette from this data.

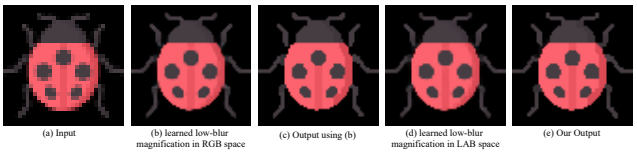


Figure 22: *Color space impact: (left) RGB space magnification; (right) OkLab pace magnification; while our method recovers the top part of the vertical line on ladybug's back, it gets removed by the RGB space method.*

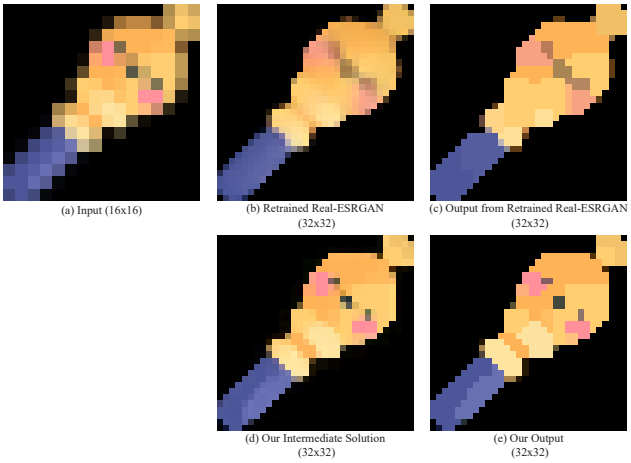


Figure 23: *Replacing our pix2pix network with Real-ESRGAN retrained on our inputs.*

Appendix C: Study Setup

We detail below the protocols used for the three studies reported on in the paper. All study data is provided in the supplementary.

Perception of Anti-Aliased Images

Color Palette Study

Our first informal study aimed to understand how human observers perceive the size of the artist-intended color palettes in anti-aliased clip-art. Participants in this study were presented with 10 anti-aliased, low resolution images and were asked the question "Mentally remove the anti-aliasing blur. How many distinct colors does the deblurred image have?" They were presented with two basic examples (two diagonally placed rectangles, with three colors total in the image; and one single color "O" shape with two colors total in the image); no other instructions were provided. The study included six participants, 3 male and 3 female.

In all cases, participants perceived input images as having small palette sizes, with answers that were largely consistent across all inputs and closely matching the number of colors in the originating vector images. When participants did not correctly identify the number of colors in the originating vector image, they tended to slightly underestimate, rather than overestimate, the number of colors used. This study confirms our focus on compact color palettes

as a key property of the mental images viewers conjure when presented with anti-aliased clip-art. The survey and participant answers are included in the supplementary material.

Segmentation Study

Our second informal study aimed to understand how human observers mentally segment anti-aliased clip-art images. (Fig. 2, in paper.) Participants in this study were presented with 6 images and were asked to "Mentally deblur and magnify this image. Trace the outlines of the single color regions in the blur-free output you envisioned. Please pay attention to details." They were presented with two basic tracing examples (single color "O" shapes and two diagonally placed rectangles); no other instructions were provided. The study included five participants, four male and one female.

Participants' traced outputs were largely consistent, with some variation in details, and were largely closely aligned with the region boundaries in the blur-free double resolution rasterizations of the underlying inputs. Participants did not hallucinate regions that were not evident in the input. The outputs were therefore consistent with our hypothesis of cross-resolution consistency and simplicity as major factors in perception of anti-aliased clip art imagery. Our outputs on the inputs traced by the participants are included in the supplementary, and are well aligned with the manual tracing outputs.

Comparative Study

In our comparative study, participants were shown input images, together with our result and an alternative result using the following layout. The input was shown at the top and marked as 'A', and the two magnified outputs were placed at the bottom and marked as 'B' and 'C'. The order of the magnified outputs on the bottom was randomized. Participants were then asked to "Mentally deblur and magnify the anti-aliased raster image on the top (A). Which of the images on the bottom (B or C) comes closest to the blur-free image you mentally assembled? Please zoom in to see the differences." The possible answer options were "B", "C", "Both", and "Neither". They were shown two ground truth examples: in one option, participants were shown the ground truth output and an anti-aliased double resolution rasterization of the originating image; in the other they were shown the ground truth image and a nearest neighbor magnification of the input. Participants were shown the answers to those. For VectorMagic we used the setting of "artwork with blended edges", "high quality" and "unlimited color' which is recommended for anti-aliased clip art and which produced the best results. We used default parameters for all other methods. The study included 70 participants, 53 male and 17 female. The complete list of questions and answer breakdowns are included in the supplementary.