Exploring CycleGAN and Its Application to Font Transfer

Asif Iqbal
Ryerson University
Toronto, ON, Canada
asif1.iqbal@ryerson.ca

Abstract

Unpaired image to image translation has gained quite a bit of attention with the advent of Cycle-Consistent Generative Adversarial Networks (CycleGANs). Translation domains like $horse \leftrightarrow zebra$, $apple \leftrightarrow orange$, $photo \leftrightarrow painting$, $summer \leftrightarrow winter$ and several others have been explored in the original work. In this paper, we plan to regenerate some of the works done in CycleGAN, try out a couple of the already tried domains on our own, and finally apply the CycleGAN concept to font style transfer. Specifically, we try it out on Arial to Times New Roman black fonts for single uppercase characters and also on lowercase multi-character words, and demonstrate that it might be a promising direction. Although it is not at all hard to get paired data for text fonts, we hope that our approach can in the future be extended to font image transfer tasks where paired data might indeed be hard to attain. The code has been made open source in the Github repository https://github.com/asif31iqbal/cycle-gan-pytorch.

1. Introduction

Image-to-image translation is a class of vision and graphics problems where the goal is to learn the mapping between an input image and an output image using a training set of aligned image pairs [28]. Image to image [28]. The field of image-to-image translation has been studied to quite an extent over the last couple of years. This problem can be more broadly described as converting an image from one representation of a given scene, x, to another, y, e.g., grayscale to color, image to semantic labels, edge-map to photograph [10, 28]. Years of research in computer vision, image processing, computational photography, and graphics have produced powerful translation systems in the supervised setting, where example image pairs $\{x_i, y_i\}_{i=1}^{N}$ are available [3, 4, 9, 11, 13, 16, 19, 23, 24, 27]. However, obtaining paired data for many tasks can be difficult and expensive. Obtaining input-output pairs for graphics tasks like artistic stylization can be even more difficult since the desired output is highly complex, typically requiring artistic authoring [28]. Let's say we want to transfer a particular summer scene into a winter one and vice versa. We can easily imagine how the corresponding winter version of a summer scene or a summer version of a winter scene might look like even though we might have never seen a summer and winter version of the same scene side by side. Based on this insight, the authors of CycleGAN [28] came up with the algorithm that can learn to translate between domains without paired input-output examples, assuming that there is some underlying relationship between the domains for example, that they are two different renderings of the same underlying scene and seek to learn that relationship. Although the algorithm lacks supervision in the form of paired examples, it can exploit supervision at the level of sets: we are given one set of images in domain X and a different set in domain Y. We may train a mapping $G : X \leftarrow Y$ such that the output $\hat{y} = G(x), x \in X$, is indistinguishable from images $y \in Y$ by an adversary trained to classify $\hat{y}$ apart from y [28]. However, as discussed in [28] that there could be infinitely many mappings G that will induce the same distribution over $\hat{y}$. Also, there is the problem of mode collapse [7], where all input images map to the same output image and the optimization fails to make progress.

To tackle these problems, the CycleGAN [28] authors leverage the notion of *cycle consistency*, in the sense that if we transfer the font style of a character from Arial to Times New Roman, and then translate it back from Times New Roman to Arial, we should get back the original character. Mathematically, if we have a translator $G : X \leftarrow Y$ and another translator $F : Y \leftarrow X$, then G and F should be inverses of each other, and both mappings should be bijections. We apply this structural assumption by training both the mapping G and F simultaneously, and adding a cycle consistency loss [**?**] that encourages $F(G(x)) \approx x$ and $G(F(y)) \approx y$.

The authors [28] have applied this idea to a wide range of applications, like collection style transfer, object transfiguration, season transfer and photo enhancement. In this paper, we first attempt to regenerate their work and

network architecture from scratch, apply it to couple of domains that they have already tried out, namely season ($summer \leftrightarrow winter$) transfer and object style transfer ($apple \leftrightarrow orange$). Next, we apply it to the domain of font style transfer. We limit ourselves to just two fonts - Arial and Times New Roman. We try it on single uppercase black English characters and on lowercase black English words. Although it is not difficult to get paired data for this sort of font style transfers for well known fonts which are widely available, there are unknown fonts, text and calligraphy styles that are available in the wild for which it is not easy to get paired data and applying the CycleGAN concept might be a good idea. The attempt with known fonts in this paper is a baby step towards the possible applicability of unknown font style transfers using cycle consistency.

2. Related Work

Over the last couple of years, Generative Adversarial Networks (GANs) [7, 8] have achieved quite a bit of success in image generation. The key idea behind GAN's success is the *adversarial loss* that forces the generated image to be indistinguishable from the input image. In the CycleGAN case, the adversarial loss has been adopted in such a way that the generated images are indistinguishable from the images in the target domain.

Much of the related work regarding unpaired image-to-image translation have been mentioned in the original paper [28]. Approaches like [9, 16, 18, 12] and **Pix2Pix** work on paired training examples, as opposed to the unpaired training concept that CycleGAN relies on.

There has been a few works on unpaired image-to-image translation as well. Works like [1, 14, 15] uses a weight sharing strategy between domains. Another group of work like [2, 20, 25] encourages the input and output to share specific content features even though they may differ in style. Unlike these approaches, the CycleGAN concept does not rely on any task specific, predefined similarity measurements. It's more of a general purpose framework.

As mentioned in the original paper, the idea of cycle consistency also has quite a bit of a history. Of these works, [6], [26] and [**?**] are the ones that are conceptually most similar to CycleGANs.

Neural Style Transfer [11, 5, 21] is another family of work for image to image translation, which synthesizes an image by combining the content of one image with the style of another image. Again, this is a paired training concept while CycleGAN is unpaired.

The primary focus of CycleGAN and hence also of this paper, is learning the mapping between two two image collection, rather than between two specific images, by trying to capture correspondence between higher-level appearance structures. We try to apply the same idea in case of font style transfer.

3. Problem Formulation

We formulate our problem in the same way the original authors [28] did, where the goal is to learn mapping between two domains A and B given training samples $\{a_i\}_{i=1}^{N}$ where $a_i \in A$ and $\{b_j\}_{j=1}^{M}$ where $b_j \in B$. The problem formulation has been depicted broadly in Figure 1.

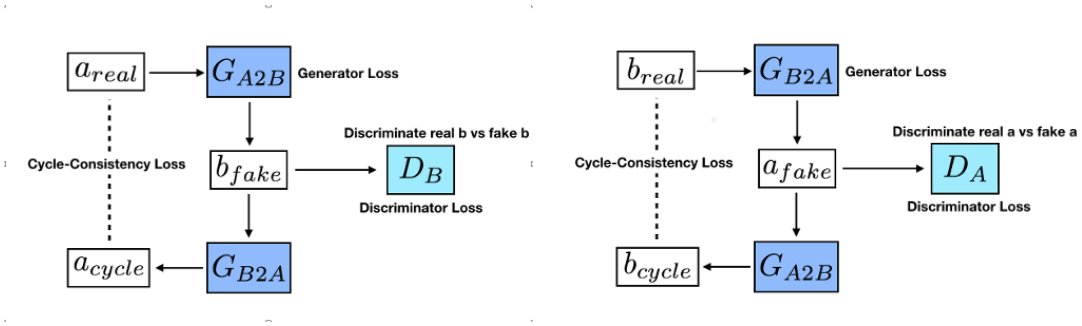
We have two generators G_A and G_B, and two discriminators D_A and D_B. G_{A2B} takes a real image a_{real} from domain A and generates a fake image b_{fake} in domain B, while G_{B2A} takes a real image b_{real} from domain B and generates a fake image a_{fake} in domain A. Discriminator D_A tries to discriminate between the generated image a_{fake} and real images in domain A, while discriminator D_B tries to discriminate between the generated image b_fake and real images in domain B. The generated fake image b_{fake} is then fed back to G_{B2A} to generate an image a_{cycle} in domain A, while the generated fake image a_{fake} is then fed back to G_{A2B} to generate an image b_{cycle} in domain B.

3.1. Loss Functions

As can be seen from Figure 1, there are broadly 2 sorts of losses - *adversarial loss* (generator loss and discriminator loss) and *cycle-consistency loss*. The adversarial loss comprises of the losses incurred from the generators trying to fool the discriminators to take fake images as real in their corresponding domain, and from the discriminators trying to distinguish fake images from real ones. As discussed in [28], with large enough capacity, a network can map the same set of input images to any random permutation of images in the target domain, where any of the learned mappings can induce an output distribution that matches the target distribution. Thus, adversarial losses alone cannot guarantee that the learned function can map an individual input a_{real} to a desired output y_{fake}. This is where the *cycle-consistency* loss kicks in - the image a_{cycle} should be the same as image a_{real}, and the image b_{cycle} should be the same as image b_{real}.

To be complete, the loss functions can be broken down into the following components:

1. D_A must approve all the original images a_{real} of the domain A
2. D_A must reject all the images b_{fake} which are generated by G_{B2A} to fool it
3. G_{B2A} must make D_A approve all the generated images b_{fake}, so as to fool it
4. Image b_{cycle} must retain the property of original image b_{real}
5. D_B must approve all the original images b_{real} of the domain B



(a) Generator G_{A2B} takes an image a_{real} from domain A and outputs a_{fake} which Discriminator D_B tries to distinguish from actual images in domain B. Image a_{fake} is then passed onto generator G_{B2A} which generates a_{cycle}. This is used for calculating the *cycle-consistency* loss.

(b) Generator G_{B2A} takes an image b_{real} from domain B and outputs a_{fake} which Discriminator D_A tries to distinguish from actual images in domain A. Image b_{fake} is then passed onto generator G_{A2B} which generates b_{cycle}. This is used for calculating the *cycle-consistency* loss.

Figure 1: Problem Formulation

6. D_B must reject all the images a_{fake} which are generated by G_{A2B} to fool it
7. G_{A2B} must make D_B approve all the generated images a_{fake}, so as to fool it
8. Image a_{cycle} must retain the property of original image a_{real}

In the above list, items $1, 2, 3, 5, 6$ and 7 are adversarial components of the loss, while items 4 and 8 a re the *cycle-consistency* components.

The original authors used L_2 (MSE) loss for the adversarial components, and L_1 loss for the *cycle-consistency* component since it earned them better results. We adhere to the same principle for our work. The loss equations can be mathematically written as follows:

$$\mathcal{L}_{disc} = \|D_A(a_{real}) - 1\|_2 + \|D_B(b_{real}) - 1\|_2 + \|D_A(a_{fake}) - 0\|_2 + \|D_B(b_{fake}) - 0\|_2$$

This captures 1, 2, 5 and 6 above.

$$\mathcal{L}_{gen} = \|D_A(a_{fake}) - 0\|_2 + \|D_B(b_{fake}) - 0\|_2$$

This captures 3 and 7 above.

$$\mathcal{L}_{cycle} = \|a_{real} - a_{cycle}\|_1 + \|b_{real} - b_{cycle}\|_1$$

This captures 4 and 8 above.

So the total loss comes down to:
$\mathcal{L}_{total} = \mathcal{L}_{disc} + \mathcal{L}_{gen} + \lambda\mathcal{L}_{cycle}$

where λ is a parameter to control how much weight we want to put on the *cycle-consistency* as opposed to the adversarial behaviour.

4. Data Collection

For the $summer \leftrightarrow winter$ and $apple \leftrightarrow orange$ transformations, we collect the datasets from the original authors' source data [17]. The *summer2winter* dataset contains 1231 train and 309 test images for summer and 962 train and 238 test images for winter. The *apple2orange* dataset contains 995 train and 266 test images for apple and 1019 train and 248 test images for orange.

For the font style transfer, we have written python programs (included in the github repository) to generate images of single uppercase English characters and lowercase words. We obtain the words vocabulary from [22]. For the single uppercase character scenario, we generated 11 train images and 10 test images for each character with random scaling and translation, so in total we got 286 train images and 260 test images. For the word scenario, we randomly sampled 1981 words from [22] to create four disjoint datasets (sizes 500, 493, 496, 492) in a way that each of them has an approximately equal number of words starting with a certain character, to avoid bias as much as possible. We used the first 2 of these sets to generate images with Arial font and use those for training and testing respectively. Similarly, we used the latter 2 sets to generate training and testing images for Times New Roman font. We also applied random scaling and translation to the words before generating the words.

5. Network Architecture

We essentially rebuilt the same network architecture as the original authors [28] have used. The architecture is shown in Figure 2. The numbers of channels at all intermediate stage have been shown in red, as well as the kernel, stride and padding size in blue. If the type of padding is Reflection padding, it has been used indicated in the figure in the corresponding blocks, otherwise padding is done with 0 values everywhere else.

5.1. Generator

The generator has 3 main segments - Encoding, Transformation an Decoding. Figure 2a The encoding part has 3 general convolutional layers each of which is a convolution followed by instance normalization [28] and Relu.b The transformation part is a series of 9 residual blocks (shown in more detail in Figure 2b). Finally, the decoding segment is a couple of general deconvolutional layer followed by a convolution+tanh(kernel size 7). Each general deconvolutional layer is a deconvolution followed by instance normalization and Relu. The details have benn shown in Figure 2a.

5.2. Discriminator

For the discriminator, we use the same 70×70 PatchGAN concept used in [28]. It basically has 4 general convolutional layers. The first one is a convolution plus Leaky Relu, while the other three are convolution plus instance normalization plus Leaky Relu. These 4 layers are then followed by one single convolution only layer. The details have been shown in red in Figure 2c.

Notice that both the generator and the discriminator networks are end-to-end fully convolutional. Before feeding our images to these networks, we resize them to 256×256. The output size of the generator network would be of the same size as it's input ($256 \times 256 \times 3$). The output of the discriminator is just a one single channel 30×30 feature map, which is compared against a 30×30 all 1's or all 0's tensor while calculating discriminator loss.

6. Experimentation

We train our networks on a GeForce 1080 GPU using the training data collected for $summer2winter$ and $apple2orange$, and also using the generated training data for our font style transfer.

6.1. Language

All manuscripts must be in English.

6.2. Dual submission

Please refer to the author guidelines on the CVPR 2015 web page for a discussion of the policy on dual submissions.

6.3. Paper length

For CVPR 2015, the rules about paper length have changed, so please read this section carefully. Papers, excluding the references section, must be no longer than eight pages in length. The references section will not be included in the page count, and there is no limit on the length of the references section. For example, a paper of eight pages with two pages of references would have a total length of 10 pages. **Unlike previous years, there will be no extra page charges for CVPR 2015.**

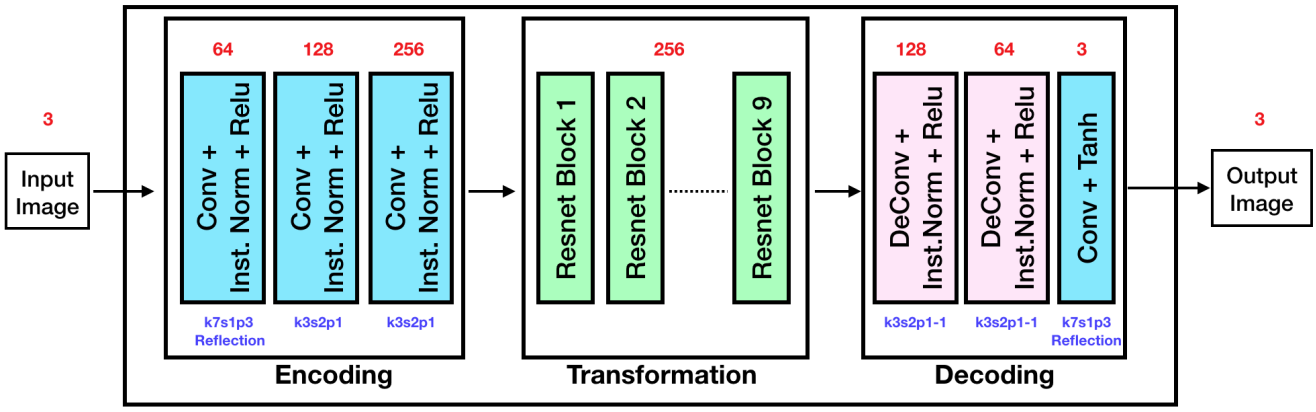
Overlength papers will simply not be reviewed. This includes papers where the margins and formatting are deemed to have been significantly altered from those laid down by this style guide. Note that this LaTeX guide already sets figure captions and references in a smaller font. The reason such papers will not be reviewed is that there is no provision for supervised revisions of manuscripts. The reviewing process cannot determine the suitability of the paper for presentation in eight pages if it is reviewed in eleven.

6.4. The ruler

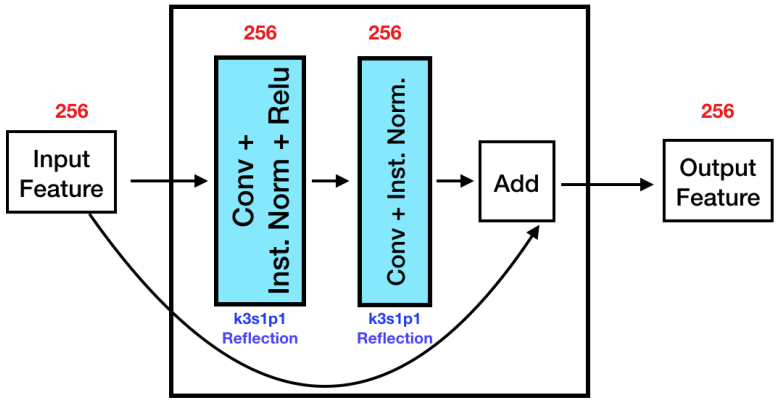
The LaTeX style defines a printed ruler which should be present in the version submitted for review. The ruler is provided in order that reviewers may comment on particular lines in the paper without circumlocution. If you are preparing a document using a non-LaTeX document preparation system, please arrange for an equivalent ruler to appear on the final output pages. The presence or absence of the ruler should not change the appearance of any other content on the page. The camera ready copy should not contain a ruler. (LaTeX users may uncomment the `\cvprfinalcopy` command in the document preamble.) Reviewers: note that the ruler measurements do not align well with lines in the paper — this turns out to be very difficult to do well when the paper contains many figures and equations, and, when done, looks ugly. Just use fractional references (e.g. this line is 095.5), although in most cases one would expect that the approximate location will be adequate.

6.5. Mathematics

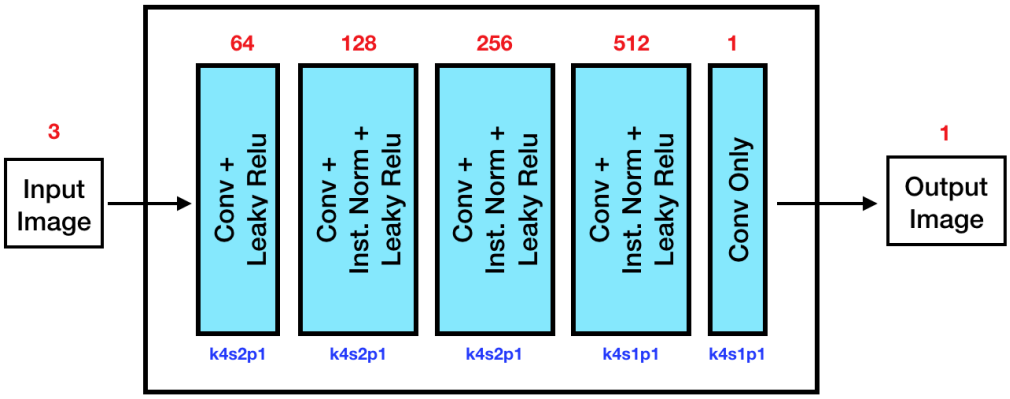
Please number all of your sections and displayed equations. It is important for readers to be able to refer to any particular equation. Just because you didn't refer to it in the text doesn't mean some future reader might not need to refer to it. It is cumbersome to have to use circumlocutions like "the equation second from the top of page 3 column 1". (Note that the ruler will not be present in the final copy, so is not an alternative to equation numbers). All authors will benefit from reading Mermin's description of how to write mathematics: `http://www.pamitc.org/documents/mermin.pdf`.



(a) Generator



(b) Resnet Block in Detail



(c) Discriminator

Figure 2: Network Architecture. The numbers in red refers to the number of output channels of each block. The letter-number combination in blue in form kxsypz beneath the blocks corresponds to the kernel size (x), stride (y) and padding (z). For example, k4s2p1 would mean a kernel size of 4×4, stride of 1 and padding of 1. If the padding is form $z - z$, that means that a padding of z is applied to the output as well (in addition to the input padding of z). The use of reflection padding has been indicated by the word Reflection in blue

6.6. Blind review

Many authors misunderstand the concept of anonymizing for blind review. Blind review does not mean that one

must remove citations to one's own work—in fact it is often impossible to review a paper unless the previous citations are known and available.

Blind review means that you do not use the words "my" or "our" when citing previous work. That is all. (But see below for techreports.)

Saying "this builds on the work of Lucy Smith [1]" does not say that you are Lucy Smith; it says that you are building on her work. If you are Smith and Jones, do not say "as we show in [7]", say "as Smith and Jones show in [7]" and at the end of the paper, include reference 7 as you would any other cited work.

An example of a bad paper just asking to be rejected:

> An analysis of the frobnicatable foo filter.
>
> In this paper we present a performance analysis of our previous paper [1], and show it to be inferior to all previously known methods. Why the previous paper was accepted without this analysis is beyond me.
>
> [1] Removed for blind review

An example of an acceptable paper:

> An analysis of the frobnicatable foo filter.
>
> In this paper we present a performance analysis of the paper of Smith *et al*. [1], and show it to be inferior to all previously known methods. Why the previous paper was accepted without this analysis is beyond me.
>
> [1] Smith, L and Jones, C. "The frobnicatable foo filter, a fundamental contribution to human knowledge". Nature 381(12), 1-213.

If you are making a submission to another conference at the same time, which covers similar or overlapping material, you may need to refer to that submission in order to explain the differences, just as you would if you had previously published related work. In such cases, include the anonymized parallel submission [**?**] as additional material and cite it as

> [1] Authors. "The frobnicatable foo filter", F&G 2014 Submission ID 324, Supplied as additional material `fg324.pdf`.

Finally, you may feel you need to tell the reader that more details can be found elsewhere, and refer them to a technical report. For conference submissions, the paper must stand on its own, and not *require* the reviewer to go to a techreport for further details. Thus, you may say in the body of the paper "further details may be found in [**?**]". Then submit the techreport as additional material. Again, you may not assume the reviewers will read this material.

Sometimes your paper is about a problem which you tested using a tool which is widely known to be restricted to a single institution. For example, let's say it's 1969, you have solved a key problem on the Apollo lander, and you believe that the CVPR70 audience would like to hear about your solution. The work is a development of your celebrated 1968 paper entitled "Zero-g frobnication: How being the only people in the world with access to the Apollo lander source code makes us a wow at parties", by Zeus *et al*.

You can handle this paper like any other. Don't write "We show how to improve our previous work [Anonymous, 1968]. This time we tested the algorithm on a lunar lander [name of lander removed for blind review]". That would be silly, and would immediately identify the authors. Instead write the following:

> We describe a system for zero-g frobnication. This system is new because it handles the following cases: A, B. Previous systems [Zeus et al. 1968] didn't handle case B properly. Ours handles it by including a foo term in the bar integral.
>
> ...
>
> The proposed system was integrated with the Apollo lunar lander, and went all the way to the moon, don't you know. It displayed the following behaviours which show how well we solved cases A and B: ...

As you can see, the above text follows standard scientific convention, reads better than the first version, and does not explicitly name you as the authors. A reviewer might think it likely that the new paper was written by Zeus *et al*., but cannot make any decision based on that guess. He or she would have to be sure that no other authors could have been contracted to solve problem B.

FAQ: Are acknowledgements OK? No. Leave them for the final copy.

6.7. Miscellaneous

Compare the following:

`$conf_a$`	$conf_a$
`$\mathit{conf}_a$`	conf_a

See The TEXbook, p165.

The space after *e.g*., meaning "for example", should not be a sentence-ending space. So *e.g*. is correct, *e.g.* is not. The provided `\eg` macro takes care of this.

When citing a multi-author paper, you may save space by using "et alia", shortened to "*et al*." (not "*et. al.*" as "*et*" is a complete word.) However, use it only when there are three or more authors. Thus, the following is correct: " Frobnication has been trendy lately. It was introduced by Alpher [**?**], and subsequently developed by Alpher and Fotheringham-Smythe [**?**], and Alpher *et al*. [**?**]."

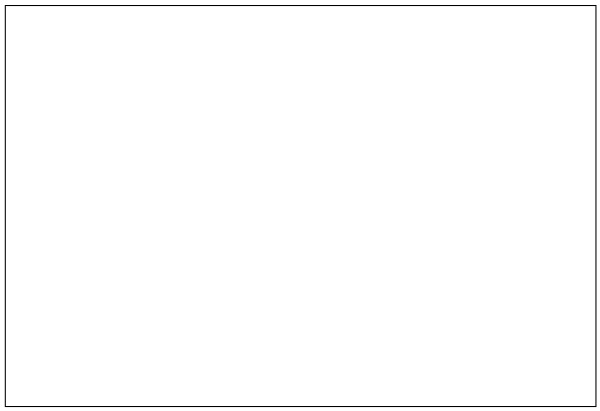


Figure 3: Example of caption. It is set in Roman so that mathematics (always set in Roman: $B \sin A = A \sin B$) may be included without an ugly clash.

This is incorrect: "... subsequently developed by Alpher *et al*. [**?**] ..." because reference [**?**] has just two authors. If you use the `\etal` macro provided, then you need not worry about double periods when used at the end of a sentence as in Alpher *et al*.

For this citation style, keep multiple citations in numerical (not chronological) order, so prefer [**?**, **?**, **?**] to [**?**, **?**, **?**].

7. Formatting your paper

All text must be in a two-column format. The total allowable width of the text area is $6\frac{7}{8}$ inches (17.5 cm) wide by $8\frac{7}{8}$ inches (22.54 cm) high. Columns are to be $3\frac{1}{4}$ inches (8.25 cm) wide, with a $\frac{5}{16}$ inch (0.8 cm) space between them. The main title (on the first page) should begin 1.0 inch (2.54 cm) from the top edge of the page. The second and following pages should begin 1.0 inch (2.54 cm) from the top edge. On all pages, the bottom margin should be 1-1/8 inches (2.86 cm) from the bottom edge of the page for 8.5×11-inch paper; for A4 paper, approximately 1-5/8 inches (4.13 cm) from the bottom edge of the page.

7.1. Margins and page numbering

All printed material, including text, illustrations, and charts, must be kept within a print area 6-7/8 inches (17.5 cm) wide by 8-7/8 inches (22.54 cm) high. Page numbers should be in footer with page numbers, centered and .75 inches from the bottom of the page and make it start at the correct page number rather than the 4321 in the example. To do this fine the line (around line 23)

```
%\ifcvprfinal\pagestyle{empty}\fi
\setcounter{page}{4321}
```

where the number 4321 is your assigned starting page.

Make sure the first page is numbered by commenting out the first page being empty on line 46

```
%\thispagestyle{empty}
```

7.2. Type-style and fonts

Wherever Times is specified, Times Roman may also be used. If neither is available on your word processor, please use the font closest in appearance to Times to which you have access.

MAIN TITLE. Center the title 1-3/8 inches (3.49 cm) from the top edge of the first page. The title should be in Times 14-point, boldface type. Capitalize the first letter of nouns, pronouns, verbs, adjectives, and adverbs; do not capitalize articles, coordinate conjunctions, or prepositions (unless the title begins with such a word). Leave two blank lines after the title.

AUTHOR NAME(s) and AFFILIATION(s) are to be centered beneath the title and printed in Times 12-point, non-boldface type. This information is to be followed by two blank lines.

The ABSTRACT and MAIN TEXT are to be in a two-column format.

MAIN TEXT. Type main text in 10-point Times, single-spaced. Do NOT use double-spacing. All paragraphs should be indented 1 pica (approx. 1/6 inch or 0.422 cm). Make sure your text is fully justified—that is, flush left and flush right. Please do not place any additional blank lines between paragraphs.

Figure and table captions should be 9-point Roman type as in Figures 3 and 4. Short captions should be centred.
Callouts should be 9-point Helvetica, non-boldface type. Initially capitalize only the first word of section titles and first-, second-, and third-order headings.

FIRST-ORDER HEADINGS. (For example, **1. Introduction**) should be Times 12-point boldface, initially capitalized, flush left, with one blank line before, and one blank line after.

SECOND-ORDER HEADINGS. (For example, **1.1. Database elements**) should be Times 11-point boldface, initially capitalized, flush left, with one blank line before, and one after. If you require a third-order heading (we discourage it), use 10-point Times, boldface, initially capitalized, flush left, preceded by one blank line, followed by a period and your text on the same line.

7.3. Footnotes

Please use footnotes[1] sparingly. Indeed, try to avoid footnotes altogether and include necessary peripheral observations in the text (within parentheses, if you prefer, as in this sentence). If you wish to use a footnote, place it at the

[1]This is what a footnote looks like. It often distracts the reader from the main flow of the argument.

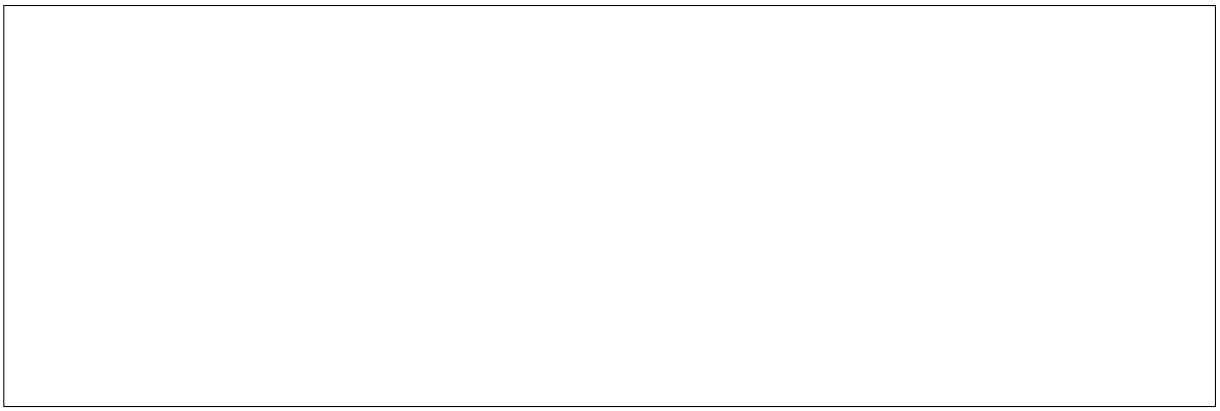


Figure 4: Example of a short caption, which should be centered.

Method	Frobnability
Theirs	Frumpy
Yours	Frobbly
Ours	Makes one's heart Frob

Table 1: Results. Ours is better.

bottom of the column on the page on which it is referenced. Use Times 8-point type, single-spaced.

7.4. References

List and number all bibliographical references in 9-point Times, single-spaced, at the end of your paper. When referenced in the text, enclose the citation number in square brackets, for example [**?**]. Where appropriate, include the name(s) of editors of referenced books.

7.5. Illustrations, graphs, and photographs

All graphics should be centered. Please ensure that any point you wish to make is resolvable in a printed copy of the paper. Resize fonts in figures to match the font in the body text, and choose line widths which render effectively in print. Many readers (and reviewers), even of an electronic copy, will choose to print your paper in order to read it. You cannot insist that they do otherwise, and therefore must not assume that they can zoom in to see tiny details on a graphic.

When placing figures in LaTeX, it's almost always best to use `\includegraphics`, and to specify the figure width as a multiple of the line width as in the example below

```
\usepackage[dvips]{graphicx} ...
\includegraphics[width=0.8\linewidth]
                {myfile.eps}
```

7.6. Color

Please refer to the author guidelines on the CVPR 2015 web page for a discussion of the use of color in your document.

8. Final copy

You must include your signed IEEE copyright release form when you submit your finished paper. We MUST have this form before your paper can be published in the proceedings.

References

[1] Y. Aytar, L. Castrejon, C. Vondrick, H. Pirsiavash, and A. Torralba. Cross-modal scene networks. *PAMI*, 2016.

[2] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *CVPR*, 2017.

[3] D. Eigen and R. Fergus. Predicting depth, surface normal and semantic labels with a common multi-scale. *ICCV*, 2015.

[4] D. Eigen and R. Fergus. Predicting depth, surface normal and semantic labels with a common multi-scale. *ICCV*, 2015.

[5] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. *CVPR*, 2016.

[6] C. Godard, O. M. Aodha, and G. J. Brostow. Unsupervised monocular depth estimation with left-right consistency. *CVPR*, 2017.

[7] I. Goodfellow. Nips 2016 tutorial: Generative adversarial. *arXiv preprint arXiv:1701.00160*, 2016.

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *NIPS*, 2014.

[9] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. *SIGGRAPH*, 2001.

[10] P. Isola, J. Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[11] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. *ECCV*, 2016.

[12] L. Karacan, Z. Akata, A. Erdem, and E. Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. *arXiv preprint arXiv:1612.00215*, 2016.

[13] P.-Y. Laffont, Z. Ren, X. Tao, C. Qian, and J. Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM TOG, 33(4):149*, 2014.

[14] M.-Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. *NIPS*, 2017.

[15] M.-Y. Liu and O. Tuzel. Coupled generative adversarial. *NIPS*, 2016.

[16] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *CVPR*, 2015.

[17] T. Park. Cyclegan dataset. `https://people.eecs.berkeley.edu/~taesung_park/CycleGAN/datasets`.

[18] P. Sangkloy, J. Lu, C. Fang, F. Yu, and J. Hays. Scribbler:controlling deep image synthesis with sketch and color. *CVPR*, 2017.

[19] Y. Shih, S. Paris, F. Durand, and W. T. Freeman. Datadriven hallucination of different times of day from a single outdoor photo. *ACM TOG, 32(6):200*, 2013.

[20] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. *CVPR*, 2017.

[21] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. *ICML*, 2016.

[22] I. University of California. Uci bag-of-words vocabulary. `https://archive.ics.uci.edu/ml/machine-learning-databases/bag-of-words/vocab.kos.txt`.

[23] X. Wang and A. Gupta. Generative image modeling using style and structure adversarial networks. *ECCV*, 2016.

[24] S. Xie and Z. Tu. Holistically-nested edge detection. *ICCV*, 2015.

[25] A. P. Y. Taigman and L. Wolf. Unsupervised cross-domain image generation. *ICLR*, 2017.

[26] Z. Yi, H. Zhang, T. Gong, Tan, and M. Gong. Unsupervised dual learning for image-to-image translation. *ICCV*, 2017.

[27] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. *ECCV*, 2016.

[28] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CVPR*, 2017.