Photo-Sketching: Evaluation Project

Michael Tran, Victor Chen

**Abstract**—Edges, boundaries and contours are important subjects of study in both computer graphics and computer vision. On one hand, they are the 2D elements that convey 3D shapes, on the other hand, they are indicative of occlusion events and thus separation of objects or semantic concepts. In this paper, we aim to leverage an existing framework titled PhotoSketch which generates contour drawings (boundary-like drawings), that capture the outline of a visual scene. We use these contours to propose a solution to problems that plague neural networks, namely adversarial images that aim to trick a neural network into misclassification. <TODO: Add some more stuff here>

**Index Terms**—PhotoSketch, Contours, Deep Learning, Deep Neural Networks, CNN <TODO: Add some more stuff here>

COSC 6373 – Computer Vision

—————————— ◆ ——————————

# 1 Introduction

E

dge-like visual representation appearing in form of image edges, object boundaries, line drawings and pictorial scripts, is of great research interest in both computer vision and computer graphics. Contour drawings of images contain object boundaries and salient inner edges such as occluding contours and salient background edges. These sets of visual cues convey 3D perspective, length, width, thickness, and depth. This paper explores, studies, and evaluates the possibilities of image contours.

Deep learning algorithms such as Convolutional Neural Networks (CNNs) are powerful models that are trained on terabytes of data to be able to classify an image. CNNs consider the entire image when classifying and can recognize complex shapes and patterns no matter where they appear in the image. In many image recognition tasks, they can equal or even beat human performance. However, it is very easy to fool a neural network by simply changing a few pixels in an image to be darker or lighter. A hacked image can cause a neural network to misclassify an image that it would, in normal circumstances, classify correctly.

In this paper, we will evaluate the potentials of using image contours to prevent hacked images from tricking convolutional neural networks. This evaluation depends on the effectiveness of a neural network trained on contoured images.

<TODO>

# 2 PhotoSketch

Mentian et Al. [1] proposes a learning-based method named PhotoSketch that resolves diversity in the annotation, and unlike boundary detectors, can work with imperfect alignment of the annotation and the actual ground truth.

Specifically, they propose a contour generation algorithm to output contour drawings given input images. The generation process involves identifiying salient boundaries and is connected with the salient boundary detection in computer vision.

## 2.1 Dataset

The dataset consists of 5000 high-quality drawings made by humans of 1000 outdoor images crawled from Adobe Stock. To collect this dataset, they used a popular crowd-sourcing platform called Amazon Mechanical Turk. [2] Turkers were allowed to trace over a fainted background image to ensure that drawings are roughly boundary aligned.

## 2.2 Method

The paper leverages a conditional generative adversarial network (cGAN), and a novel MM-loss (Min-Mean-loss) to generate the contours.

For cGANs, the generator aims to generate “real” images conditioned on the input images. Adversarially, the discriminator network is trained to tell the generated images from the actual ground truth images. Typically, this method expects a 1-to-1 mapping between the two domains. However, to accommodate the extra images in each training example, they used a MM-loss to account for this.

In the novel MM-loss, two different aggregation functions are used for the generator and discriminator respectively. The “mean” aggregate function asks the discriminator to learn from all modalities in the target domain, and treat those modalities with equal importance. The “min” aggregate function allows the generator to adaptively pick the most suitable modality to generate on-the-fly.

## Model

In this paper, we will be using the original authors pre-trained model. This model is trained on the dataset mentioned in the previous section, which mainly contains pictures of humans and dogs. Thus, the contours generated will be most accurate when generating contours of humans and dogs, however, we have tested it on other types of images with success.

# 3 Hacked Images

Generating hacked images is essentially the same as “generating an adversarial example”. It is intentionally crafting a piece of data such that a machine learning model will misclassify it as something completely different. This can be used for something harmless such as a prank, but it can also be used for something malicious such as uploading an image (e.g. a pornographic image) that violates a websites terms of services. A deep neural network would normally be the first line of defence and able to catch something like that; however, a hacked image is a way to bypass that defence. Hacked images can even fool neural networks even when they are printed out on a piece paper! [3] This means that hacked images can not only fool systems that upload an image file directly, but also fool physical cameras or scanners.

## 3.1 Hacked Image Generation

The generation of hacked images is suprisingly simple, we just need to change a few pixels in an image to be darker or lighter. One would expect that changing a couple of pixels on an image would not matter to a deep neural network. However, in a famous paper in 2014 by C. Szegedy et al. [4] discovered that it isn’t always true. If one knew exactly which pixels to change and exactly how much to change them, you can intentionally force the neural network to predict the wrong output without making any obvious changes to the human eye.

## 3.2 Pipeline

For this task, we use and aim to trick the Inception v3 image recognition model. This model was created by Google and has been shown to attain greater than 78.1% accuracy on the ImageNet dataset. [5] The model itself is made up of symmetric and asymmetric building blocks, including convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers.

The following is a pipeline of how we will generate the hacked images:

1. Feed in a training photo.
2. Check Inception v3’s prediction and see how far off the image is from the fake prediction.
3. Tweak the photo using back-propagation to make the final prediction slightly closer to the fake prediction.
4. Repeat steps 1-3 with the same photo until the Inception v3 network gives us the fake prediction we want.

One thing to keep in mind is that we cannot allow any single pixel to be adjusted without any limitations, or else the changes to the image can be drastic enough that it can be seen by the human eye. These changes will show up as discolored spots or wavy areas. To prevent these distortions, we add a constraint to our algorithm: no one single pixel in the hacked image can ever be changed by more than a tiny amount from the original image (we use 0.01%). This forces the algorithm to tweak the image in a way that still fools the neural network without having any obvious changes from the original image.

## 3.3 Results

For this paper, we force every input image we want to hack to be classified as a toaster. Figure 1 and Figure 2 below shows two seemingly identical images that Inception v3 believes to be totally different. We note that Inception v3 does not classify humans, but for all intents and purposes, we show that we can still force the network into thinking the image is a toaster. Figure 3 shows that we can even improve the networks confidence of a toaster image being a toaster.



**Figure 1** – Left: Welsh\_springer\_spaniel with 26.75% confidence. Right: Toaster with 92.83% confidence.



**Figure 2** – Left: Wig with 60.11% confidence. Right: Toaster with 99.56% confidence.



**Figure 3** – Left: Toaster with 98.26% confidence. Right: Toaster with 100% confidence.

## 3.4 Image Contours and Hacked Images

Given the original image and hacked image, we feed both images into the PhotoSketch generator. Figure 4 and Figure 5 shows the resultant image contours.



**Figure 4** – Left: Image contours of original image. Right: ****Image contours of hacked image.



**Figure 5** – Left: Image contours of original image. Right: Image contours of hacked image.

At first glance, the image contours of Figure 5 may look identical, but there are slight differences in the contours (some contour lines are longer in one image than the other). Overall, we can say that (at least from the human eye’s perspective) the contours for hacked and non-hacked images are almost identical.

# 4 Neural Networks and Contour Images

One test we conducted was the effectiveness of a Neural Network trained with contour images. For this task, we used PhotoSketch to generate contour images and then trained two CNNs using the two datasets, contours and real images. We propose that, the CNN trained on PhotoSketch’s generated contour images could still perform classification.

## 4.1 Method

We used Google’s TensorFlow for our CNN. Our CNN was composed of 4 convolutional and max pooling layers followed by fully connected layer.

## 4.2 Datasets

For our datasets, we chose to use two different datasets. The first dataset we chose was used to differentiate completely different objects and was provided by M. Li [1]. This dataset is the dog and human dataset.

The second dataset is the CelebA [7] dataset, a compilation of celebrity face images. This dataset was chosen to test if the salient features are retained in contour images and can be used to make accurate gender predictions.

## 4.3 Results

The results from testing on the second dataset are straightforward and follow our initial prediction. Contour images can be used to train CNNs, but suffer an effectivity penalty. We used a small sample size of 114 female and 106 male celebrity faces. The train-test-validation split was 80-10-10. The CNN trained with real celebrity face images performed, on average, 91% accuracy over both classes. The CNN trained with PhotoSketch generated contour images of the same real celebrity face images performed, on average 85% accuracy over both classes. Figure 6 demonstrates two sets of images. A real celebrity face image and its PhotoSketch generated counterpart.

**Figure 6** – Left: Original real image. Right: Contour image

# 5 Topic2

# 4 Citations

IEEE Computer Society style is to note citations in individual brackets, followed by a comma, e.g. “[1], [5]” (as opposed to the more common “[1, 5]” form.) Citation ranges should be formatted as follows: [1], [2], [3], [4] (as opposed to [1]-[4], which is not IEEE Computer Society style). When citing a section in a book, please give the relevant page numbers [2]. In sentences, refer simply to the reference number, as in [3]. Do not use “Ref. [3]” or “reference [3]” At the beginning of a sentence use the author names instead of “Reference [3],” e.g., “Smith and Smith [3] show ... .” Please note that references will be formatted by IEEE Computer Society production staff in the same order provided by the author.

# 5 Equations

If you are using Word, use the MathType add-on (<http://www.mathtype.com>) for equations in your paper (Insert | Object | Create New | Microsoft Equation or MathType Equation). “Float over text” should not be selected.

For display equations as seen below, number equations consecutively with equation numbers in parentheses flush with the right margin, as in (1). First, use the equation editor to create the equation. Then, select the “Equation” markup style. Press the tab key and write the equation number in parentheses. To make your equations more compact, you may use the solidus ( / ), the exp function, or appropriate exponents. Use parentheses to avoid ambiguities in denominators. Punctuate equations when they are part of a sentence, as in

 (1)

Be sure that the symbols in your equation have been defined before the equation appears or immediately following. Italicize symbols (*T* might refer to temperature, but T is the unit tesla). Per IEEE Computer Society, please refer to “(1),” not “Eq. (1)” or “equation (1),” except at the beginning of a sentence: “Equation (1) shows ... .” Also see *The Handbook of Writing for the Mathematical Sciences*, 1993. Published by the Society for Industrial and Applied Mathematics, this handbook provides some helpful information about math typography and other stylistic matters. For further information about typesetting mathematical equations, please visit the IEEE Computer Society style guide: <http://www.computer.org/portal/web/publications/style_math>.

Please note that math equations might need to be reformatted from the original submission for page layout reasons. This includes the possibility that some in-line equations will be made display equations to create better flow in a paragraph. If display equations do not fit in the two-column format, they will also be reformatted. Authors are strongly encouraged to ensure that equations fit in the given column width.

# 6 Helpful Hints

## 6.1 Figures and Tables

Because IEEE Computer Society staff will do the final formatting of your paper, some figures may have to be moved from where they appeared in the original submission. Figures and tables should be sized as they are to appear in print. Figures or tables not correctly sized will be returned to the author for reformatting.

Detailed information about the creation and submission of images for articles can be found at <http://www.computer.org/portal/web/peerreviewjournals/author#figures> where you can [View transactions art\_guide.pdf (PDF, 4.69MB)](http://www.computer.org/cms/Computer.org/Journal%20templates/transactions_art_guide.pdf) . We strongly encourage authors to carefully review the material posted here to avoid problems with incorrect files or poorly formatted graphics.

Place figure captions below the figures; place table titles above the tables. Figure captions appear as left justified. Table captions are restricted to one sentence and are formatted as title case. Any additional sentence in a table caption will be formatted as a footnote below the table (see Table 1 in this document). If your figure has two parts, include the labels “(a)” and “(b)” as part of the artwork. Please verify that the figures and tables you mention in the text actually exist. Figures and tables should be called out in sequential order, as this is how they will be placed in your paper. For example, avoid referring to figure “8” in the first paragraph of the article unless figure 8 will again be referred to after the reference to figure 7. **Please do not include figure captions as part of the figure. Do not put captions in “text boxes” linked to the figures. Do not put borders around the outside of your figures.** Per IEEE Computer Society, please use the abbreviation “Fig.” even at the beginning of a sentence. Do not abbreviate “Table.” Tables are numbered numerically.

For journals that use print for publication, please verify with IEEE Computer Society that the journal you are submitting to does indeed accept color before submitting final materials. **Do not use color unless it is necessary for the proper interpretation of your figures.**

Figures (graphs, charts, drawing or tables) should be named fig1.eps, fig2.ps, etc. If your figure has multiple parts, please submit as a single figure. Please do not give them descriptive names. Author photograph files should be named after the author’s LAST name. Please avoid naming files with the author’s first name or an abbreviated version of either name to avoid confusion. If a graphic is to appear in print as black and white, it should be saved and submitted as a black and white file (grayscale or bitmap.) If a graphic is to appear in color, it should be submitted as an RGB color file.



Fig. 1. Magnetization as a function of applied field. Note that “Fig.” is abbreviated. There is a period after the figure number, followed by one space. It is good practice to briefly explain the significance of the figure in the caption.

Figure axis labels are often a source of confusion. Use words rather than symbols. As an example, write the quantity “Magnetization,” or “Magnetization *M*,” not just “*M*.” Put units in parentheses. Do not label axes only with units. As in Fig. 1, for example, write “Magnetization (A/m)” or “Magnetization (Am−1),” not just “A/m.” Do not label axes with a ratio of quantities and units. For example, write “Temperature (K),” not “Temperature/K.” Table 1 shows some examples of units of measure.

Multipliers can be especially confusing. Write “Magnetization (kA/m)” or “Magnetization (103 A/m).” Do not write “Magnetization (A/m) × 1,000” because the reader would not know whether the top axis label in Fig. 1 meant 16,000 A/m or 0.016 A/m. Figure labels should be legible, approximately 8 to 12 point type. When creating your graphics, especially in complex graphs and charts, please ensure that line weights are thick enough that when reproduced at print size, they will still be legible. We suggest at least 1 point.

## 6.3 Footnotes

Number footnotes separately in superscripts (Insert | Footnote)[[1]](#footnote-1). Place the actual footnote at the bottom of the column in which it is cited; do not put footnotes in the reference list (endnotes). Use letters for table footnotes (see Table 1). Please do not include footnotes in the abstract and avoid using a footnote in the first column of the article. This will cause it to appear above the affiliation box, making the layout look confusing.

TABLE 1  
Units for Magnetic Properties



Statements that serve as captions for the entire table do not need footnote letters.

aGaussian units are the same as cgs emu for magnetostatics; Mx = maxwell, G = gauss, Oe = oersted; Wb = weber, V = volt, s = second, T = tesla, m = meter, A = ampere, J = joule, kg = kilogram, H = henry.

## 6.4 Lists

The IEEE Computer Society style is to create displayed lists if the number of items in the list is longer than three. For example, within the text lists would appear 1) using a number, 2) followed by a close parenthesis. However, longer lists will be formatted so that:

1. Items will be set outside of the paragraphs.
2. Items will be punctuated as sentences where it is appropriate.
3. Items will be numbered, followed by a period.

## 6.5 Theorems and Proofs

Theorems and related structures, such as axioms corollaries, and lemmas, are formatted using a hanging indent paragraph. They begin with a title and are followed by the text, in italics.

**Theorem 1.** *Theorems, corollaries, lemmas, and related structures follow this format. They do not need to be numbered, but are generally numbered sequentially.*

Proofs are formatted using the same hanging indent format. However, they are not italicized.

**Proof.** The same format should be used for structures such as remarks, examples, and solutions (though these would not have a Q.E.D. box at the end as a proof does). 

# 7 End Sections

## Appendices

Appendices, if present, appear online as supplemental material. In the event multiple appendices are required, they will be labeled “Appendix A,” “Appendix B, “ etc.

IEEE Computer Society Transactions accepts supplemental materials for review with regular paper submissions. These materials may be published on our Digital Library with the electronic version of the paper and are available for free to Digital Library visitors. Please see our guidelines below for file specifications and information. Any submitted materials that do not follow these specifications will not be accepted. All materials must follow US copyright guidelines and may not include material previously copyrighted by another author, organization or company. More information can be found at <http://www.computer.org/portal/web/peerreviewjournals/author#supplemental>.

## 7.2 Acknowledgments

The preferred spelling of the word “acknowledgment” in American English is without an “e” after the “g.” Use the singular heading even if you have many acknowledgments. Avoid expressions such as “One of us (S.B.A.) would like to thank ... .” Instead, write “F. A. Author thanks ... .” Sponsor and financial support acknowledgments are included in the acknowledgment section. For example: This work was supported in part by the US Department of Commerce under Grant BS123456 (sponsor and financial support acknowledgment goes here). Researchers that contributed information or assistance to the article should also be acknowledged in this section. Also, if corresponding authorship is noted in your paper it will be placed in the acknowledgment section. Note that the acknowledgment section is placed at the end of the paper before the reference section.

## 7.3 References

Unfortunately, the Computer Society document translator cannot handle automatic endnotes in Word; therefore, type the reference list at the end of the paper using the “References” style. See the IEEE Computer Society’s style for reference formatting at: <http://www.computer.org/portal/web/publications/style_refs>. The order in which the references are submitted in the manuscript is the order they will appear in the final paper, i.e., references submitted nonalphabetized will remain that way.

Please note that the references at the end of this document are in the preferred referencing style. Within the text, use “et al.” when referencing a source with more than three authors. In the reference section, give all authors’ names; do not use “et al.” Do not place a space between an authors' initials. Papers that have not been published should be cited as “unpublished” [4]. Papers that have been submitted or accepted for publication should be cited as “submitted for publication” [5]. Please give affiliations and addresses for personal communications [6].

Capitalize all the words in a paper title. For papers published in journals not published in English, please give the English citation first, followed by the original foreign-language citation [7].

## 7.4 Additional Formatting and Style Resources

Additional information on formatting and style issues can be obtained in the IEEE Computer Society Style Guide, which is posted online at: <http://www.computer.org/portal/web/publications/styleguide>. Click on the appropriate topic under the Special Sections link.

# Conclusion

Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. A conclusion might elaborate on the importance of the work or suggest applications and extensions. Authors are strongly encouraged not to reference multiple figures or tables in the conclusion—these should be referenced in the body of the paper.

**Acknowledgment**

The authors wish to thank A, B, C. This work was supported in part by a grant from XYZ.

**References**

1. M. Li, “Photo-Sketching: Inferring Contour Drawing from Images” 2019 IEEE Winter Conference on Applications of Computer Vision. (2019) Available: https://arxiv.org/abs/1901.00542
2. Amazon Mechanical Turk. [Online] Available: https://www.mturk.com/
3. Adversarial Examples in the Physical World. [Online] Available: https://www.youtube.com/watch?v=zQ\_uMenoBCk
4. C. Szegedy, “Intriguing properties of neural networks” (2014) Available: https://arxiv.org/abs/1312.6199
5. Advanced Guide to Inception v3 on Cloud TPI. [Online] Available: <https://cloud.google.com/tpu/docs/inception-v3-advanced>
6. Z. Liu, “Deep Learning Face Attributes in the Wild” 2015 Proceedings of International Conference on Computer Vision (ICCV) Available: http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

1. It is recommended that footnotes be avoided (except for the unnumbered footnote with the receipt date on the first page). Instead, try to integrate the footnote information into the text. [↑](#footnote-ref-1)