DSC412 Project: Image to Knot Pattern Conversion

Cassidy Petrykowski
Computer Engineering Dept.
North Carolina State University
Raleigh, NC
cmpetryk@ncsu.edu

***Abstract*— The arts of weaving and pattern-making are intertwined, and yet the process to convert a design into a pattern is arduous, and if a pattern is lost, often so is the design. The objective of this project is to translate an object or design from an image into a knotted pattern, that can be used to hand weave textiles. The model used was PIX2PIX, a model trained with a conditional GAN.**

I. Background

Woven textiles are a staple across different cultures and time. And from Iranian rugs to Brazilian lace-making, patterns are a defining piece of the weaving process. For the scope of this project, the focus will be on macrame patterns, specifically hand-woven knotted textiles generally made in the form of bracelets, lanyards, belts and other long thin form factors. Colloquially (in the US) this kind of textile is referred to a friendship bracelet, but the knotting techniques and general use cases extend far beyond bracelets.

Typically knotwork patterns are made by the craftsperson, taking inspiration from a design, natural entity or object they see around them. However, when the craftsperson begins drafting the pattern, typically represented by a series of pixels/boxes (knots), as shown in figure 1 [1], they are making a series of educated guesses based off their experience, and it is very difficult bringing together their intended design with their intended size and thread count.

The primary problem being approached is the lack of baseline pattern for a particular intended design, especially if that design falls outside of very standard designs dating back thousands of years. Craftspeople can see a design or object, but the transfer to a usable pattern is an arduous process. This presents a barrier of entry to newcomers. Furthermore, even if an experienced craftsperson loses their pattern for a particular design, they begin all over again. Pattern loss is actually a relevant problem for lace weaving as well, as the difficulty to enter the industry and sustain a career means that less and less traditional lace weavers are working every year. Combined with the practice of destroying lace patterns, often patterns are lost to time without the ability to recreate them [2].

The objective is not to take the art from the artist, but rather to aid them in developing the baseline pattern which they can then tweak, adjust the size of or expand upon. Pattern loss and barrier of entry to pattern creation are two sides of the same issue, or at the very least two issues that can be approached with a similar method of solving.

A. Novelty

Multiple papers have been published attempting to use machine learning or AI techniques in regards to textiles. The majority of the research has been in analyzing and differentiating textiles, especially in areas like knitting or machine-woven fabric. More intricate, often hand-woven techniques, like knotwork, have yet to be studied in such a manner. The primary novelty of the project lies in its ability to generate rudimentary patterns from images for knotwork, which has not been successfully done yet or has yet to be published.

Machine learning modeling has not been applied to knotwork but has been applied to machine knitting [3] and pattern-making for lace [2]. Machine learning has also been applied in the textiles industry to help with woven pattern recognition [4].

II. Data Set

BraceletBook [5] is a very large vault of friendship bracelet-style macrame patterns. It has a filtered system that shows both the image of the bracelet, the gridded pattern, *and* the final pattern. The image of the bracelet is one part of the data, while the gridded pattern is the second part. The bracelet images are referred to as the "photo data" while the gridded patterns are referred to as "grids" hereafter.

My roommate generously agreed to loan her creations and their patterns. Images were collected of both and added to the dataset [6].

Collecting the data from Braceletbook was a huge learning process. The data was webscraped, first obtaining the grid, then procuring all of its respective photos. However, some pages have additional grids, so those had to be avoided and the photos had to be organized into specific folders associated with their respective grid. The images would later have to be merged to adhere to the training format, and their merged locations were added to patterns.csv. The data augmentation was predominantly applied *during* the webscraping process, as described in V. The improvements that could be made in the future to this process and would likely have very direct impact on the output image quality are described in VIII.

The data was organized into two categories, photos and grids, then subdivided into training, test, and validation data following the classically recommended ratio of 80/10/10 percentages respectively. Overall 13,000 image pairs were scraped for training.

III. Model Selection

Two model types were seriously considered for the project: an image segmentation model or a GAN-based approach. Color-based image segmentation was considered because the patterns have a different color for each knot/pixel, but ultimately the idea was discarded as the primary model because the approach is not looking for each knot individually but trying to find a large region filled with the same color and isolating it. A different method of image-segmentation called thresholding came into consideration later for data augmentation. and a GAN (or multiple) will be used in conjunction with one another for this project [7].

GANs were generally more applicable to the goals of this project–image generation. However, the difficulty lay in finding a GAN that accepted image inputs *as well as* image outputs. Enter PIX2PIX[8]. After researching, PIX2PIX appeared to not only accept image inputs, but also the example training data bore similarity to the data used in this project which, although not indicative of the outcome, is reassuring.

IV. Data Analysis

The very first step in analyzing image data is simply visualizing it. Looking at the data, it became apparent that both the photos and the grids needed to be augmented to have a stronger relationship with one another.

The shape of the image was also inspected and the sizes were averaged, to see if the images could be run through the PIX2PIX model. The structure only accepts 256 x 256 images without modification. The sizes of the grids were also printed out to get an idea of how the grids would need to be modified.

There were three big takeaways from the data analysis. Firstly, both the photos and grids needed to be resized. The photos were 218 x 218 and just needed to be scaled up, but the grids came in all shapes and sizes, the only similarity being that the heights were under or at 256. The second takeaway was that images needed to be RGB not RGBA to be used by the model. The third and final takeaway was that both the photos and the grids need to be modified to more closely resemble one another.

Fig. 1. Photo of Tiger's Eyes

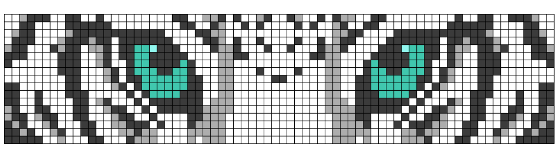

Fig. 2. Grid of Tiger's Eyes

Figures 1 is a photograph taken by the author of a real handmade bracelet [6]. Figure 2 is of the grid pattern that the bracelet in figure 1 is based upon. The two images look strikingly alike. However, having inspected approximately 100 photos and their associated grids, there are some clear differences.

Firstly, the photos have backgrounds that at times share colors with the bracelet itself. Conversely, the grids do not have backgrounds but do not come in a square shape, meaning that the grid must either be cropped or have a standard color filled in to fit it to the same square size as its photo pair.

It is worth noting that the patterns are not all of bracelets, and there is not a clear way to discern between when an image is of a bracelet or a keychain or a bookmark. Fundamentally all are fairly similar shapes, but the differing element is whether or not the pattern runs vertically or horizontally.

All in all, the photos and grids had to be resized, in RGB format, *and* required data augmentation to get the images pairs as close to resembling one another as possible.

Although it could be argued, like the old adage of 'Which came first? the chicken or the egg?', that the data influenced the model, truthfully the model heavily influenced how the data had to be modified to fit the input requirements. That being said, the model was picked due to the type of data, so both really influenced each other.

V. Data Augmentation

Data augmentation most likely took up the most programming time, after data collection.

The data was surprisingly clean for scraped data. If this project were being used in a non-educational sense, perhaps there is room for refinement, but every grid was entirely pixels and every image had a photo of a pattern in it. There were a few images (<5%) where an image was not very close to the bracelet or part of the bracelet turned sideways, but even in these images they still showed >75% of the pattern.

The easiest parts of data augmentation was removing the 'A' from RGBA images. In the loading step, the fourth 'A' value was cut off of the tensor, ensuring that no RGBA tensors were loaded into the model.

The second easiest part was scaling up the images from 218 x 218 to 256 x 256 sidelengths. Python's image library PILLOW has functions that retain scale and can resize an image [9].

The not-easy-in-any-way-shape-or-form part was modifying the images to resemble one another more. Two approaches were taken to image modification, the first being to shrink and fill in the grids, the second to crop the grids to fit within the square shape. The first modification methodology posed issues because more often than not the output was a black square with a small line of color on top. The second methodology could create a square with a grid, but got consistently confused by the background of the photos.

Background removal for the photo images is a possible solution for this. Image segmentation was already considered during the model selection point in the process, and is tested during the data analysis. However, multiple attempts to implement thresholding image segmentation as part of the data augmentation process failed and it began to hold up the project. How the data collection and data augmentation process could have been implemented better, thereby allowing the backgrounds to be removed from the photos is addressed later in VIII.

VI. Training Methodology

The model was trained using the training script which was taken primarily from PIX2PIX[8]. The data used is described at length from section II to section V. The dataset was split into training, testing, and validation data as described in II. At first, the model was trained with a 70/21/9 perecent split for training, testing and validation respectively which was too small and adjusted to the standard percentage split for GANs of 80/10/10.

The hyperparameter training was definitely an area that could have been better addressed. No modifications were made outside of those set by the original PIX2PIX code.

The validation loss is tough to discern with PIX2PIX. The realization that has occurred *after* training the model is that to view the validation loss to determine overfitting/underfitting, the validation and training loss had to be defined before training, which is a problem with this particular model, addressed but not resolved entirely on its git repo [10].

Taking a look at the training loss in figures 3 and 4, however, there is a positive trend of the discriminator loss waning while the generator loss waxes.

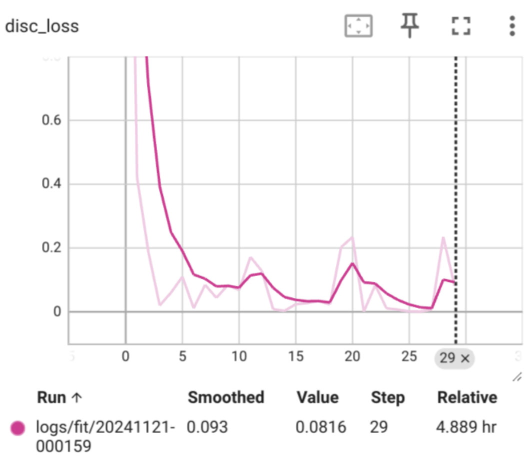

Fig. 3. Discriminator Loss

VII. Results

Quantifying the results of a GAN is difficult because image data is inherently qualitative. Originally, the proposed method was to appeal to online forums that weavers use such as BraceletBook [5], but the model does not output a result that felt consistent enough or accurate enough to publish to the internet. For the purposes of the project, the results are assessed by means of inspection.

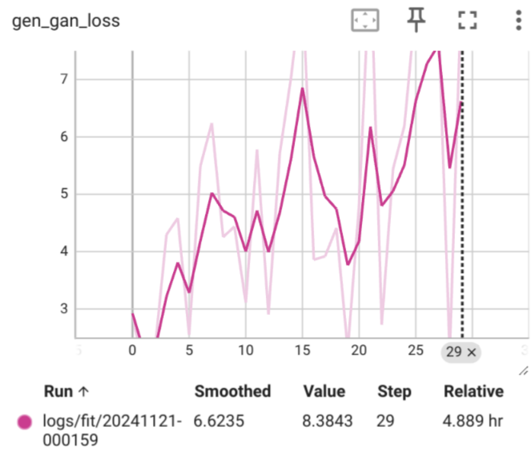

Fig. 4. Generator Loss

One of the frustrating pieces of working with PyTorch is the lack of inbuilt metrics for measuring the results of an image processing or GAN-based project. Conversely PyTorch offers an inbuild metric for Frechet Inception Distance (FID), Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM)[11]. At length, a version of FID, whether using PyTorch alongside Tensorflow in a Frankenstein-ed fashion or from open source code was attempted, but was not successful. This is further addressed in VIII.

Shown below in figure 5 are the input, expected output, and actual output of the model built using the photos and cropped grids. The output resembles a grid at at around step 10,000 more strongly resembled the expected output but does not fully translate the desired image. The shape of the pattern, the background, and/or any unrelated objects plan a large part in the model struggling to correlate the photos to the appropriate grids.

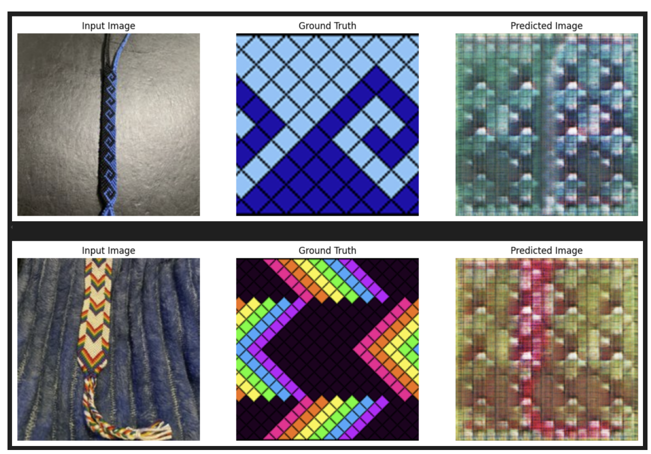

Fig. 5. Comparison of input, expected output 'Ground Truth' and actual output from model

VIII. Future Work

Two approaches to data augmentation were attempted, but one of the fundamental issues behind both approaches was that the data was augmented as it was scraped. So, if this project were continued, the first amendment would be to separate the data augmentation and webscraping, although not an easy task due to the volume of data. With this amendment, it might be easier to be to the 500,000 to 100,000 image range recommended for training GANs. The 13,000 and subsequent re-downloading of data was time-consuming, and if the processes were split, data would never have to be redownloaded and would likely take less time.

The second change would be in the data augmentation approaches. The LaMa model [12] was originally considered for the actual model, but the openCV thresholding [7] method was better suited for data augmentation. If the data augmentation was separated from the image procurement, this process would be easier to add before the images are merged and fed into the model. Removing the background would help alleviate distractions like other objects in the images. The only drawback is if the thresholding method either chooses the wrong object over the bracelet or if the bracelet shares colors with the background. There are more accurate background removal methods that often involved machine learning models, such as the online background removal software used to prepare figure 1 [13]. However, they do not take a high volume of images (for free) or expeditiously, nor is their code available (logically), so the free methods available (without an accessible code base) are as they are. That said, with more research, a better solution could almost certainly be found or developed. The question then that remains is after removing the background and replacing it with all black coloring, would the square images of the patterns with an all black backdrop then be useful? Or would the grids without a black background still yield better results?

IX. Stakeholder Acknowledgments

My roommate and other weavers are the potential stakeholders for this project, making them also an excellent resource for inquiries into what would be a useful outcome and what would not be. With further development of the project, more stakeholders across textiles might become apparent, but for the time being, the broader weaving, knotting and macrame community are the primary stakeholders.

My roommate would test out one of the patterns, but they are not quite in a usable state yet.

If the results were further along, the potential negative implications would be removing the artist from the art, but as addressed previously, this is a tool *for* artists like digital art pads or art software, not a replacement for the design creation or knotwork itself.

X. Conclusion

My biggest takeaways from this process were threefold:

1) Data analysis and resultant augmentation bears the largest impact on the output of the model.
2) There were so many more different adjustments that could have been made. With more time and separation between data collection and data augmentation, those changes could be made and tested more efficiently.
3) Defining metrics for image data is important *because* it is difficult. In the future, more clearly defining those metrics would help make the training smoother and more logically approached.

Overall, this project was chosen for my personal interest in it, rather than an end goal in sight, meaning that I wanted to explore the idea but should have had a better idea of how to measure my success. I will, however, continue forward with the project, because I would like it to be in a usable enough state to give to my roommate for the holidays.

References

[1] J. Coslina, "Pattern 176719." [Online]. Available: https://www.braceletbook.com/patterns/

[2] S. A. Ahteck, H. Gold, C. Cheng, R. Gu, A. Huynh, E. Scott, G. Goldstein, and S. V. Boriskina, "Generative AI for Textile Engineering: Blending Tradition and Functionality Through Lace," *An MIT Exploration of Generative AI*, sep 10 2024, https://mit-genai.pubpub.org/pub/nz0prx2u.

[3] F. Scheidt, J. Ou, H. Ishii, and T. Meisen, "deepknit: Learning-based generation of machine knitting code," *Procedia Manufacturing*, vol. 51, pp. 485–492, 2020, 30th International Conference on Flexible Automation and Intelligent Manufacturing (FAIM2021). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2351978920319259

[4] M. A. Iqbal Hussain, B. Khan, Z. Wang, and S. Ding, "Woven fabric pattern recognition and classification based on deep convolutional neural networks," *Electronics*, vol. 9, no. 6, 2020. [Online]. Available: https://www.mdpi.com/2079-9292/9/6/1048

[5] J. Kleň, "Home." [Online]. Available: https://www.braceletbook.com/

[6] M. Hurst, "Artwork," 2024.

[7] "Image thresholding." [Online]. Available: https://docs.opencv.org/4.x/d7/d4d/tutorial$_{py_t}hresholding.html$

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 2018. [Online]. Available: https://arxiv.org/abs/1611.07004

[9] A. Clark, "Pillow (pil fork) documentation," 2015. [Online]. Available: https://buildmedia.readthedocs.org/media/pdf/pillow/latest/pillow.pdf

[10] "L1 loss on validation data," 2020. [Online]. Available: https://github.com/affinelayer/pix2pix-tensorflow/issues/200

[11] "Metrics - torcheval main documentation." [Online]. Available: https://pytorch.org/torcheval/main/torcheval.metrics.html

[12] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky, "Resolution-robust large mask inpainting with fourier convolutions," 2021. [Online]. Available: https://arxiv.org/abs/2109.07161

[13] "Remove bg - image background remover." [Online]. Available: https://bgremove.com/