논문 리뷰 내용

제목: ScrabbleGAN: Semi-Supervised Varying Length Handwritten Text Generation

저자: Sharon Fogel†, Hadar Averbuch-Elor§, Sarel Cohen, Shai Mazor† and Roee Litman†

† Amazon Rekognition, Israel § Cornell Tech, Cornell University

**들어가는 말:**

이 논문은 CVPR 2020에 개제된 논문으로 아마존과, 코넬대학교에서 같이 썼다. 본 논문은 제목에 나와있 듯, Handwritten Text Generation을 하는 논문이다. 최근 handwriting 생성 모델에 관심을 가지게 되면서, 해당 논문을 리뷰하게 되었다. Handwriting은 단지 자연스러운 모양만 아니라 의미를 알 수 있는 글자의 형태를 생성해야 된다는 점에서 어떤 기법이 쓰였는지 궁금했다. 읽은 논문이 별로 없어, 이 논문을 기점으로 handwriting에 관한 논문을 계속해서 찾아서 읽어보려고 한다. 자세하게 내용을 다루기 위해서 논문에서 나온 순서대로 하나하나 리뷰를 진행해보기로 했다.

**Abstract:**

초록의 처음에는 handwritten text recognition(HTR)의 한계점을 지적한다. 데이터 수집 과정에서의 비용과 어려움을 말하면서 semi-supervised learning을 통해 supervised learning보다 성능을 향상시킬 수 있다고 말한다. 여기서 Handwritten Text Generation에서 recognition을 언급하는 것이 의아할 수 있지만 결국 기존 데이터만 아니라 Handwritten Text Generation으로 recognition task를 수행할 때, 학습 데이터로 활용할 수 있다는 contribution을 말하려고 하는 것으로 봤다.

다음으로는 ScrabbleGAN에 대해 소개한다. ScrabbleGAN은 스타일과 내용이 변하기 쉬운 손글씨를 합성하는 semi-supervised approach라고 한다. 또, ScrabbleGAN은 임의의 길이, 즉 길이에 국한받지 않고 손글씨 이미지를 생성할 수 있는 모델임을 강조한다. 또 이 생성 결과를 data로 기존 HTR모델의 성능을 더욱 향상시켰고, 다양한 형태의 손 글씨를 만들 수 있다고 한다.

**Introduction:**

인트로의 처음 부분에는 손글씨 자체를 설명을 하는데 이 부분은 넘어가도록 하자.

본 논문의 contribution으로 세가지를 이야기하는데,

* 임의의 길이를 생성할 수 있는 fully convolutional neural network을 만들었다는 점. (기존 다른 모델들은 RNN based 모델이였음) 그리고 character 단위의 annotation없이 character character embedding을 학습할 수 있다는 점.
* Semi-supervised learning을 통해 unlabeled data를 적용했다는 점(handwritten text synthesis framework에서의 첫 시도라고 함.)
* 본 논문에서 결과물을 통해 기존 데이터셋과 결합하여 기존 HTR 모델들의 성능을 끌어올렸다는 점을 꼽는다.

**Previous work:**

논문에서 소개하는 관련 연구들을 살펴보자.

**Handwriting text recognition**

먼저 **Handwriting text recognition의** 온라인 방식과 오프라인 방식을 말한다. 온라인 방식은 그 과정을 샘플링한 stroke라는 개념을 통해 손 글씨에 대한 정보를 더 줄 수 있다. 하지만 오프라인 방식은 그 과정에 대한 정보가 아니라, 최종 결과물만 볼 수 있다고 말한다. 이런 점에서 stroke를 기록해야하는 온라인 데이터가 더 모으기 힘들다고 지목하고, 오프라인에는 아예 적용할 수 없는 방법론이라고 말한다. 따라서 논문에서는 오프라인 방법론에 대해 초점을 맞췄다고 한다.

나도 온라인 손글씨 논문에 관한 논문을 찾아보면서 그 stroke를 통해 결과를 내는 논문 (DeepWriting(CHI 2018), Handwriting Beautification Using Token Means(SIGGRAPH 2013))들을 봤는데 굳이 그 stroke를 활용하는가에 대해서는 의문을 갖는다. 물론 성능이야 좋겠지만 상용적인 면에서 따로 stroke를 측정할 수 있는 소프트웨어가 없으면 무용지물이라는 생각도 들었다.

**Handwriting text recognition**에 대한 말은 여기서 줄이고,

**Handwriting text generation**에 관련한 논문을 중점적으로 살펴보고자 한다. 논문의 핵심이기도 하고 HTR까지 설명하면 배보다 배꼽이 더 큰 느낌이기도 하다.

1. Alex Graves. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850, 2013.

먼저는 토론토 대학의 Alex Graves가 발표한 Generating sequences with recurrent neural networks이란 논문인데 citation 수가 무려 3500여 회로 굉장히 영향력 있는 논문이다. 이 논문에서는 제목 그대로 RNN을 이용한 sequential한 생성에 대해 이야기한다. 여기서 사용한 데이터는 위에서 언급한 stroke가 포함된 IAM online 손글씨 데이터셋을 사용하여, 글씨를 쓰는 과정에 있어서 다음 지점이 어딘지 LSTM을 통해 예측하고 계속해서 글씨를 만들어 낸다.

1. Bo Ji and Tianyi Chen. Generative adversarial network for handwritten text. arXiv preprint arXiv:1907.11845, 2019

이 논문은 GAN 구조를 이용한 손글씨 생성에 대한 논문이다. GAN 이 realistic한 이미지를 만드는 것에 어느정도 정평이 나있다고 생각했는데, 손 글씨 생성하는 논문이 2019년에야 제안되었다니 생각보다 늦다고 할 수 있다. 이는 손 글씨가 가지고 있는 특징 때문인데 이 논문에서는 sequential한 데이터를 CNN-LSTM방식의 discriminator를 제안하여 LSTM모델을 generator로 CNN-LSTM 구조를 discriminator로 하여 GAN 구조로 손글씨 학습을 시도했다. 이 논문 또한 IAM online 손 글씨 데이터셋을 사용한 것으로 보인다.

1. Eloi Alonso, Bastien Moysset, and Ronaldo Messina. Adversarial generation of handwritten text images conditioned on sequences. arXiv preprint arXiv:1903.00277, 2019.

본 논문보다 G + D + R 인 먼저 전체적인 구조를 제안하였다. 본 논문의 전신이라고 봐도 무방한듯, 이 논문이랑 비교를 많이한다 주요 차이는 RNN -> CNN 구조일듯 싶다.

**Data augmentation using generative models.**

Yu-Xiong Wang, Ross Girshick, Martial Hebert, and Bharath Hariharan. Low-shot learning from imaginary data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7278–7286, 2018

* **GAN을 이용해서 만든 이미지를 트레이닝이미지로 쓰는 논문인듯,**

Ayan Kumar Bhunia, Abhirup Das, Perla Sai Raj Kishore, Shuvozit Ghose, and Partha Pratim Roy. Handwriting recognition in low-resource scripts using adversarial learning. arXiv preprint arXiv:1811.01396, 2018

* Adversarial Feature Deformation Module (AFDM)을 제안, 글씨를 warp시켜서 data augmneation하는 내용

Praveen Krishnan, Kartik Dutta, and CV Jawahar. Word spotting and recognition using deep embedding. In 2018 13th IAPR International Workshop on Document Analysis Systems (DAS), pages 1–6. IEEE, 2018

* **임베딩 하는데 임의로 만들어진 데이터를 사용하였음. ->** Praveen Krishnan and C. V. Jawahar. Generating synthetic data for text recognition, 2016 여기서 나옴.. 이게 더 중요한거 같은데. -> 밑에 나옴

R. Reeve Ingle, Yasuhisa Fujii, Thomas Deselaers, Jonathan Baccash, and Ashok C. Popat. A scalable handwritten text recognition system. ArXiv, abs/1904.09150, 2019

-> 이 논문에서는 만들어진 글자 데이터로 오프라인 recongzier의 성능 향상을 제안함.

Praveen Krishnan and C. V. Jawahar. Generating synthetic data for text recognition, 2016. 3

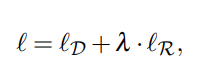
**데이터를 생성한다기보다는 데이터를 augmentation시키는거같음. ->** [**https://github.com/Belval/TextRecognitionDataGenerator**](https://github.com/Belval/TextRecognitionDataGenerator)

Joan Puigcerver. Are multidimensional recurrent layers really necessary for handwritten text recognition? In 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), volume 1, pages 67–72. IEEE, 2017. 3

* **CNN + LSTM 을 제안한 논문인데, augmentation으로 recongization 성능이 크게 오른것을 언급하는 듯 하다.**

1. **Method**

**먼저 전체적인 로스를 보여주는데 Recog loss랑 GAN loss를 비율적으로 전체 로스로 준다. 그 이유는 굉장히 직관적인데 실제같아야하고 (GAN), 알아볼 수 있어야한다는 것이다 (Recong)**

****

**Fully convolutional generator**

먼저 generator를 보자,

먼저 저자는 RNN이 아닌 CNN 구조를 차용했다는 것에 대해 설명한다.

RNN은 시작 – 현재까지의 state를 모두 사용한다는 점이 글자를 생성하는데 not trivial하다고 하며 좋지 않다고 지적한다. 하지만 CNN구조를 사용함으로써, 오직 양 옆에있는 글자만 연관되어 글자를 생성함으로 이런 문제를 해결했다고 한다. 이 오버랩된 receptive field는 글자간 상호작용하고 부드러운 변화를 만든다.

**Meet라는 글자를 만들때를 예시로 드는데 사진에서와 같이 filter bank에 각 해당하는 글자를 넣는다. 그럼 m,e,e 그리고 t 각 4개의 filter bank가 나오는 건데. 여기에 스타일을 나타내는 noise z를 더해주어 글자를 생성하기 위한 latent vector를 만든다. 그리고 위에 말한던 것 같이 각 필터뱅크를 입력으로 생성하는 네트워크에서는 양 옆 과 overlapped receptive field를 공유하면서 생성하게된다, 이런 방식은 길이의 제약이 없으며, 전체 글자의 스타일도 일관된다고 말한다.**

**또한 한 filterbank는 overlapped receptive field가 있다 하더라도 작은 부분이며, 따라서 생성한 글자는 해당 글자가 명확히 생성된다. 하지만 overlapped receptive field로써 양옆 글자가 달라짐에따라 다양성을 확보할수 있다고 말한다.**

**Style을 조절하는 vector z는 모든 각 글씨의 스타일도 일관되기 위해서 모든 filter bank에 동일하게 곱해진다.**

**3,2**

**다음으로는 Discriminator에 대해 설명한다. Discriminator의 역할은 앞서 말했듯 진짜 같은 이미지를 만드는 것과 여기서는 Discriminator의 역할이 이미지간 결과의 스타일을 분간하는 역할도 있다고 한다.**

**글자의 길이에 따른 동작은, 내가 이해하기로는, 한 필터뱅크에서 나온 (오버랩포함)글자마다 하나씩 넣고 평균을 내는 식으로 작동한다고 이해했다. 그러면 길이 변화에 따른 영향이 없이 학습이 가능하다.**

**3.3**

**Recognizer는 읽을 수 있는 텍스트를 만드는데 기여한다. Discrimminator를 손글씨 같은 정도를 만든다 치면 다른 일임에 이해하기 쉽다. Recognizer는 오직 라벨이 있는 real sample에서만 학습이 가능하다.**

**많은 모델들이 앞뒤 문맥을 볼 수 있는 bidirectional LSTM을 선택했지만, 이 모델은 글씨 자체가 제대로 쓰여있지 않아도 문맥상으로 때려 맞추는 문제가 있다고 지목한다. 자주쓰는 단어는 세 글자중 가운데가 이상해도 알아보듯이 말이다. 논문에서는 이 문제를 지목하며 한 글자 글자가 제대로 인식되는 Recognizer구조로 convolutional backbone을 사용했다고 한다.**

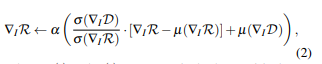
* **자세한건 SM 보라고 했음.**

**3.4**

**그 다음으로는 학습 과정에 대해서 설명한다.**

**학습은 Loss R 과 Loss D로 이루어지는데,**

**두 로스에 밸런스를 맞추기 위해서 먼저 제안된 방법은 다음과 같다.**

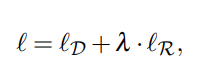
****

**시그마는 std고 뮤는 mean이다. 식을 보면 gradient R은 평균을 gradient D 와 같게 만들어주고 분산도 gradient D와 같게 하였다. 그 다음 알파로 두 로스의 밸런스를 조절하였다. 하지만 논문에서 제안한 방식은 std만 맞춰주는 방식을 택했다. 평균을 이동하면서 gradient 부호가 바뀌는 문제를 방지하고자 했다고 한다.**

**그럼 이동 안해서 생기는 문제는..? 알파로 해결하려 한듯? 근데 알파는 std를 늘리는건디.. ?**

**모르겠다.**

****

****

**결과**

**4.1**

먼저 논문에서는 생성하는 이미지를 높이 32로 고정하였고 넓이는 16 픽셀로 고정한다.

Filter bank의 크기는 32x8192인데 여기에 32차원 z 를 곱한다. 그럼 n개의 글자를 생성할 때 n x 8192가 된다고 하는데 나는 이걸 n 개의 Filterbank\*z((1x32) \* (32x8192))을 n개 concat한거라고 이해했다.

그 다음 reshape을 통해 512x4x4n (8192 = 512x4x4)가 되고, 이때 각 글자는 4x4 spatial size를 가지고 있다고 한다. 그 다음 3개의 residual blocks을 통과한 후에 업샘플링 후, 겹쳐진 영역을 만들어서 최종 아웃풋인 32x16n을 만든다고 한다. Overlapped receptive field에 대해서는 좀더 코드를 보거나 자세히 다뤄야 할 거 같다.

Discriminator 구조는 BigGAN 모델에서 차용했는데 4개의 residual blocks로 구성되고 마지막에 fc레이어가 하나 있는 구조이다. 앞서 이야기 한대로 conv layer로 구성되어있고, 각 패치의 평균이 최종 prediction이 된다. 여기서 Geometric gan에서 제안한 GAN hinge loss를 사용한다는데 체크가 필요하다.

**4.2**

다음으로는 DATASET과 METRICE을 설명하는데 데이터셋으로는 RIMES, IAM, CVL이라는 데이터셋을 사용했다.

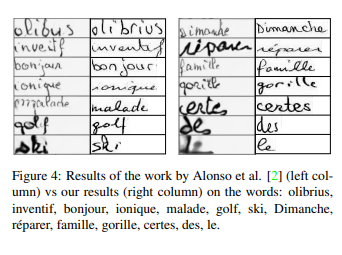
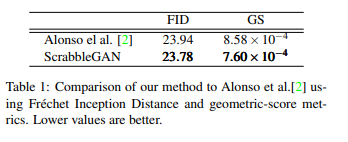
그리고 메트릭을 조금 자세히 보자. 메트릭에서는 두 gold standard metric을 사용했다하는데,

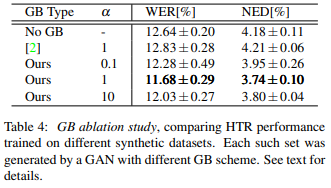
첫번째로는 word error rate(WER)이다. 말그대로 전체 단어중에 몇 개의 단어가 잘못 읽혔냐를 평가한다. 두번째는 normalized edit-distance(NED)인데, true와 prediction사이에 edit-distance를 측정한다고 한다.

**4.3**

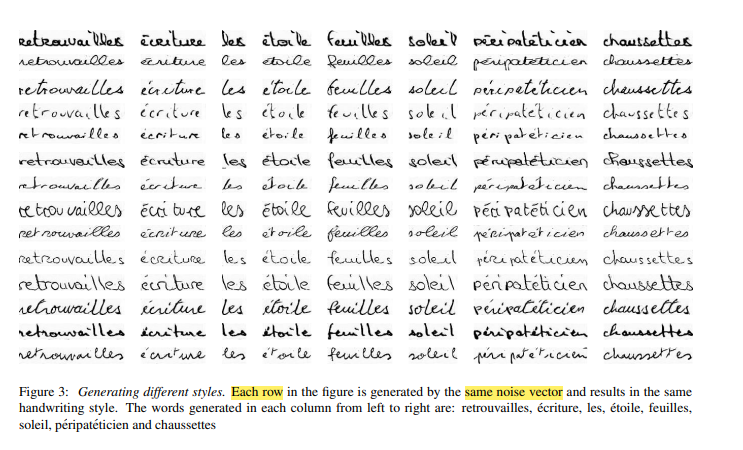
다음으로는 adversarial generation of handwritten text images conditioned on sequences 논문과 비교한다. 이 논문은 위에 말했듯 본 논문과 전체적인 구조가 매우 유사하고 논문 내용도 아주 유사하다. 이 논문을 넘는 것이 본 논문의 필수 조건이 아닌가 싶다.

**논문에서는 figure4로 정성적인 결과를, table1, table4 로 정량적인 결과를 보여주며 본 논문이 더 나음을 설명한다.**

**** ****

****

**4.4**

****

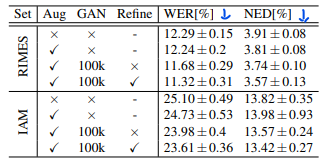
다음으로는 다른 스타일에 대한 생성을 설명한다. Figure3 를 통해 다른 각 다른 스타일의 글자가 잘 생성됨을 보이고, 인접한 글자마다의 interaction도 잘 되어 자연스러운 글씨가 생성되었다고 말한다.

**4.5**

다음으로는 제안한 네트워크로 생성한 dataset을 추가로 적용하여 기존의 HTR performance를 늘린 부분에 대해서 말한다. 예상할 수 있듯, 본 논문에서 제안한 방식으로 데이터셋을 추가로 구축한 결과가 더 나음을 설명한다.

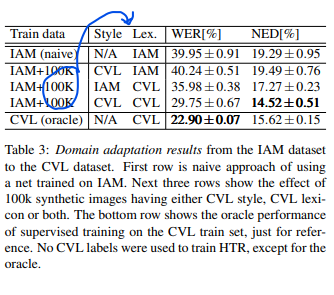
Augmentation을 한 데이터 셋보다 더 좋은 성능을 보이며, 생성한 이미지가 그저 기존데이터를augmentation한것과 다른 것임을 보인다.

하지만 여기서 3번째와 4번쨰 차이를 잘 모르겠어서 저자한테 물어본 상태다..

****

다음으로는 domain adaptation을 설명하는데,

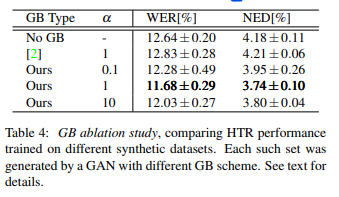
IAM에서 학습한 데이터는 아무래도 IAM에서 가장 잘되는 법이다. 논문에서는 IAM으로 학습된 데이터일지라도 본 논문에서 제안한 모델로 다른 데이터셋 스타일의 추가적인 데이터를 구성하여 학습함으로써 domain adaptation이 가능함을 보였다. Table3을 확인하면 IAM으로 학습했지만, 100k의 추가 데이터를 타겟으로 하는 CVL데이터셋 스타일로 구성하였을 때, IAM으로 학습하고 CVL로 테스트했을 때보다 더 좋은 성능을 보임을 보였다. WER성능상은 CVL 데이터 자체를 사용하는 것이 더 좋았지만, NED 성능상은 IAM에 스타일과 글자를 CVL로 추가 구성한 데이터셋으로 학습하였을 때, 더 좋은 성능을 보임을 알 수 있다.

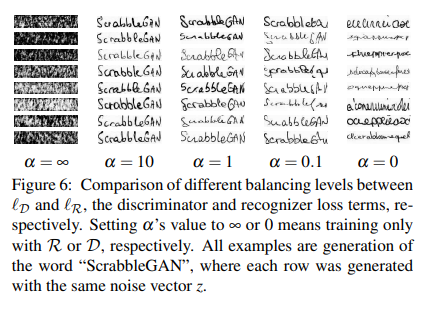
****

**4.6**

다음으로는 앞서 말했던 alpha를 조절함에 따라, 성능이 어떻게 되는지 ablation study 결과를 설명한다. Table 4에서 보면, Alpha가 1일 때 가장 좋은 성능을 보였다.

또 figure 6를 보면 alpha값을 조절함에 따라 얼마나 최종 결과물이 달라지는지를 보이는데 알파값이 lossR에 있다는걸 알고, 알파값을 높이면 글씨의 diversity가 줄어드는 것을 확인 할 수 있다. 하지만 너무 적으면 알아볼 수 없는 글씨들이 생성되는 것 또한 확인할 수 있다. 이에 저자는 적절한 alpha를 설정하는 것이 중요한 요소라고 말한다.

****



1. **Conclusion and Future Work**

결론이다.

본 논문에서는 RNN구조로 전체의 글자생성을 통으로 하나를 보는 것이 아니라 잘라서 local problem으로 만들었다고 한다.

이런 점으로 길이와 스타일에 제약받지 않은 이미지를 잘 생성할 수 있다고 주장한다.

그리고 오버랩된 receptive field로 인접한 글자간 자연스러움을 더했으며, Generator와 discrimmitor

또한 HTR의 향상을 도모한점을 짚고 넘어가고,

향후 연구 방향으로는 few shot learning으로의 방향성, style과 글씨체(굵기, 날림정도) controllable, 그리고 마지막으로는 각 글자마다 다른 receptive field를 적용시키는 방법을 제안한다. 나도 읽으면서 생각한 한계점인데 같은 글자에 스타일은 달라도 글씨의 한 글자에 해당한 길이가 일정해서 그런 측면에서 다양성이 없다는 것인데 저자도 이점을 지목했다. 이 문제를 위해 STN이라는 네트워크를 언급하며 논문을 마무리한다.

끝~~~~~~~~~~~~~~~~~~~