Neural Style Transfer Experiment

I developed an app that lets the user pick a content image and a style image at random and then attempt to simulate, by drawing, what a Neural Style Transfer Network would do: combine the content and style into one resulting image. I also present two different NST AIs and their outcome. As the last step, the user can view the results of the comparison between the 3 images (the drawing and the 2 stylized pictures), which will be a grade between 0 and 10.

Even though my project assembles more than one subject: Neural Style Transfer, Image similarity with Deep Learning, and User Interface Design, I consider that the main subject is Neural Style Transfer (NST). At this moment in time, NST has a single role in computer vision, capturing the style of a painting and representing photographs (content images) in said style in a convincing manner. It is a small, quite new domain. A bridge between technology and art.

It is not mandatory for the content image to be a photograph and for the style image to be a painting, it can be the other way around. The results are interesting nonetheless.

The main NST model used by me for this project is a model licensed by Apache-2.0, which I researched on. This model uses an architecture for extracting the style from the style image, and another one for applying the style to the content image, plus the old-fashioned VGG architecture for calculating the style and content losses. The first architecture uses layers from the InceptionV3 model, pretrained on image recognition. The second architecture is part of the *magenta* project, designed for music and painting processing in AI.

The original Neural Style Transfer architecture (VGG-19 in 2015) starts with the target image (copy of content image), extracting the gram matrixes which contain correlations between different filter responses of the style image (they contain the style, to put it briefly) and calculate the style loss in comparison to the style filter response of the target image. Same happens with the content correlations for the content loss, but the gram matrix is not computed in this case. The target image is optimized according to the loss until we get a stylized image as a result.

Our main model does a better job, as it extracts the style in a bottleneck with the first architecture trained on style prediction. It relies solely on the theory that a lot of paintings have textures in common, so no matter the image, its style can be generalized as a mixture of styles the model learned while training. The style bottleneck supplies a set of normalization constants for the style transfer network. The style transfer network is a Convolutional Neural Network with the structure of an encoder/decoder. While transforming, the architecture relies on residual blocks to solve the vanishing gradient problem caused by the big complexity of the model, which can slow down and

eventually stop the training. The style and content loss is calculated in the same fashion as the VGG-19 model, which takes as input the target, style and content images. It is believed that the style of an image is calculated by the gram matrix of the response of the first layer of each subblock of convolutions, and the content is given by a feature map resulted from a deeper layer. For calculating the content and style loss, we compute the L2-norm of said correlations and filter responses between the target, style and content images.

So the difference between the classic method and the new approach is that before calculating the loss, we make a style prediction and preprocess the content with it. The new method is an improved version of the VGG-19 NST and gives better and faster effects. In my app, I present the visible difference between the two methods and show both results.

Another AI which I used is the model that compares the resulting images and gives the similarity. I built it by training an Autoencoder on reducing the content of an image and building the image back based only on the reduced result. In order to compare two images, I use just the encoder part and differentiate the reduced content. The user will be able to compare the 2 stylized images with each other, and the stylized images with their own drawing.

I made the graphical user interface with PyQt5. I called the resulting app an “Experiment with NST”. After the user picks a content image and a style image randomly, they can either upload their own image, or choose to draw a picture with the said content and style. Upon clicking the “Start to draw” button, a new window will appear with a drawing app. For this one, I designed my own brush textures to resemble the different kinds of paint used in the abstract art picked by me.

There is also a button for the image style transfer that starts the process. I left the button enabled from the start in case the user is curious about the result of the NST models. However, the process may take around 100-140 seconds (from my experiments), so while the user is busy drawing the picture, the style transfer process will start automatically on a 2nd thread.

The last step will be the comparison between the picture the user drew or uploaded, and the NST result following the new approach.

Experimtent pe transferul neural de stil

Am dezvoltat o aplicație care permite utilizatorului să aleagă random o imagine stil (pictură) și o imagine conținut, și apoi să simuleze, prin desen, ce ar face o rețea neurală de transfer de stil: să combine într-o singură imagine stilul și conținutul. De asemenea, prezint rezultatele a două modele de transfer neural de stil. Utilizatorul poate vizualiza rezultatele comparației între cele 3 imagini: 2 stilizate și propriul său desen. Similaritatea imaginilor e dată de o notă între 0 și 10.

Deși lucrarea mea însumează mai multe subiecte: Transferul de stil pe imagini, Similaritatea între imagini calculată cu ajutorul unui model AI, și User Interface Design, consider că principalul subiect este Transferul de stil între imagini. Neural Style Transfer (NST), domeniu destul de nou în Computer Vision, are rolul de a capta stilul unei picturi și de a reprezenta diverse fotografii în acest stil. NST face o legatură între artă și IA, deci scopul este pur divertisment, dar rezultatele sunt de-a dreptul promițătoare.

Ca observație, nu este obligator ca imaginea stil să fie o pictură. Stilul se poate extrage din orice tip de imagine.

Modelul NST principal utilizat în aceasta lucrare este un model licențiat de Apache-2.0, pe care am făcut research. Acest model folosește o arhitectură pentru a extrage stilul din imaginea stil, și o altă arhitectură pentru a aplica stilul pe conținut, plus arhitectura veche utilizată pentru NST (pentru prima dată), VGG, pentru a calcula loss-ul pentru stil și conținut. Prima arhitectură utilizează straturi din modelul InceptionV3, preantrenat pe image recognition. A doua arhitectură este parte din proiectul *magenta*, destinat procesării de desen și muzică în IA.

Modelul original de NST (VGG-19, 2015) începe cu o copie a imaginii conținut, extrage matricile gram obținute în urma trecerii imaginii stil prin anumite straturi convoluționale (ne bazăm pe teoria că stilul imaginii e conținut de fiecare feature map obținut din primul strat a fiecărui sub bloc), și feature map-ul ce conține conținutul (procesat de un layer convoluțional mai adânc). Optimizăm imaginea rezultat după ce calculăm loss-ul pentru stil și conținut. Procesul se repetă până ajungem la efectul dorit.

Modelul NST principal pe care se bazează lucrarea este o versiune îmbunătățită, întrucât înainte de a calcula loss-ul și de a optimiza, stilul este extras separat într-un bottleneck. Rețeaua este preantrenată pe identificarea diverselor stiluri, și se descurcă bine și pentru picturi pe care nu a fost antrenată. Motivul este faptul că, după cum au presupus autorii, un număr limitat de picturi pentru antrenament conție toate texturile care ar putea exista în lumea artei, și orice pictură nouă e doar un amestec de texturi existente. Acest bottleneck rezultat conține diverse date de normalizare și

transformare a imaginii conținut, utilizate în arhitectura pentru transfer. Aceasta este o arhitectură convoluțională cu structura unui encoder/decoder. Aceasta folosește tehnica skip-connection pentru a rezolva problema gradientului dispărut, ce încetinește și pune piedică antrenamentului. Un ultim pas este calculul erorii (utilizând norma L2) și optimizarea în același mod cu modelul original VGG-19.

Diferența dintre modelul nou și cel clasic sunt cele 2 arhitecturi în plus pentru procesarea stilului și transformarea conținutului. Rezultatele ambelor modele sunt prezente în aplicație pentru a pune în evidență progresul față de metoda veche.

Un alt AI utilizat este cel pentru comparația între imagini. Pe acesta l-am construit cu ajutorul unui Autoencoder antrenat pe micșorarea informației dintr-o imagine pentru a o reconstrui corect doar din acel bottleneck primit. Pentru a compara 2 imagini, trec acestea numai prin prima parte – encoderul, și compar informația restrânsă. Utilizatorul poate compara cele 2 imagini stilizate, și fiecare imagine stilizată cu propriul său desen.

Interfața grafică, construită de mine cu ajutorul a PyQt5, plasează utilizatorul într-un experiment NST, unde el, după ce își alege imaginile conținut și stil, poate încărca o imagine proprie ca încercare de a simula rezultatul unui NST, sau poate desena o imagine. În acest caz, o fereastră nouă va aparea cu aplicația pentru desen. Pentru aceasta, am implementat propriile texturi pentru pensule pentru a reda acrilul și acuarela cât mai autentic. Taskul userului: să deseneze o imagine cu conținutul și stilul ales.

Fiindcă transferul de stil în ambele cazuri durează în jur de 100-140 de secunde (conform experimentelor mele), am decis să creez mai multe fire de execuție pentru fiecare model NST. Pe firele mele noi de execuție, procesul de transfer de stil începe în paralel cu utilizarea aplicației de desen de către user. Sigur, transferul de stil poate fi început înainte ca userul să decidă că va desena, doar în cazul în care e curios să vadă rezultatul modelelor NST.

Candidat: Stegarescu Ana-Maria