PLAGIARISM SCAN REPORT

Words	663	Date	April 22,2021
Characters	4412	Excluded URL	

8% Plagiarism	92% Unique	3 Plagiarized Sentences	34 Unique Sentences

Content Checked For Plagiarism

5.1 INTRODUCTION
The project is largely inspired by Christian Ledig's SRGAN paper [20] and Dong et. al [22] implementation of SRGANs using Tensorflow. Dahl et. al [10] introduction of residual encoding using VGG architecture and adaptation of GANs as conditional GANs by Mirza et al. [15] proved to be quite effective for implementing colorization of images. We provide detailed architectural design for each respective GANs and other networks that it will be compared with.
5.2 ARCHITECTURAL DESIGN
Generative Adversarial Networks (GANs) have two competing neural network models. The generator takes in the input and generates fake images. The discriminator gets the image from both the generator and the label along with the grayscale image and it determines which pair contains the real colored image. During training, the generator and the discriminator are playing a continuous game. At each iteration, generator produces a more realistic photo, while the discriminator gets better at distinguishing the fake photos. Trained in a minimax fashion, the goal is to train a generator that produces data that is indistinguishable from the real data.
5.2.1 Image Colorization
Both the generator and discriminator are conditioned on the input x with conditional GAN. Let the generator be parameterized by qg and the discriminator be parameterized by qd. The minimax objective function can be defined as:
Where, Gqg is the output of the generator and Dqd is the output of the discriminator. We're currently not introducing any noise in our generator to keep things simple for the time being. Also, we consider L1 difference between input x and output y in generator. On each iteration, the discriminator would maximize qd according to the above expression and generator would minimize qg in the following way:
With GAN, if the discriminator considers the pair of images generated by the generator to be a fake photo (not well colored), the loss will be back-propagated through discriminator and through generator. Therefore, generator can learn how to color the image correctly. At the final iteration, the parameters qg will be used in our generator to color grayscale images.
5.2.2 Image Super-resolution
We use the SRResNet as the generator in the SRGAN model as used by Ledig et. al [20]. It contains both the residual blocks and the skip connections, as seen in Figure 5.3. Within each residual block, there are two convolution layers followed by a Batch Normalization layer and a parametric ReLU layer. Finally, the image is then upscaled 4 times using two sub-pixel convolution layers [24].
The goal of the generator is to produce high resolution images that will fool the discriminator of the GAN into thinking that it is receiving real instead of fake images. On the other hand the discriminator's goal is to classify the images it has received as either
real images or generated images from the generator. The GANs objective function

is a minimax game as mentioned in the previous section. We define the minimax function for this task with trivial changes in notation and express it as:
where, IHRptrain(IHR) are the high resolution images. ILRpg(ILR) are the input low resolution images, Gqg is the output of the generator and Dqd is the output of the discriminator. We use the perpetual loss function for VGG based content losses introduced by Ledig et. al [20] which is a weighted sum of a content loss ISR
X and an
adversarial loss component (10□3ISR
Gen).
For the content loss, we aim to use the VGG loss introduced by Ledig et. al[20] which is the euclidean distance between the feature representations of a reconstructed image Gqg(ILR) and the reference image IHR:
where Wi; j and Hi; j represent the dimensions of the respective feature maps within VGG19 network. The adversarial generative loss ISR
Gen is defined on the probabilities
of the discriminator Dqd (Gqg(ILR)) over all the training samples as:
Dqd (Gqg(ILR)) is the probability that the reconstructed image Gqg(ILR)) is a natural HR image. For beter gradient behavior, we minimize □logDqd (Gqg(ILR))
instead of log

Sources	Similarity
Image Inpainting and Object Removal with Deep ... i With GAN, if the discriminator considers the pair of images generated by the generator to be fake (not well recovered), the loss will be back-propagated through discriminator and through generator. Therefore, the generator can learn how to recover the image to make it look real. 4. http://stanford.edu/class/ee367/Winter2018/fu_guan_yang_ee367_win18_report.pdf	5%
Colorization Using ConvNet and 2017-07-03ï grayscale and ... https://fdocuments.us/document/colorization-using-convnet-and-2017-07-03-grayscale-and-edge-only-from-color-images.html	5%
CS230 Deep Learning Within each residual block, there are two convolutional layers followed by a Batch Normalization layer and a parametric ReLU layer. Finally, the image is then upscaled by 4 … http://cs230.stanford.edu/projects_winter_2019/reports/15470581.pdf	5%