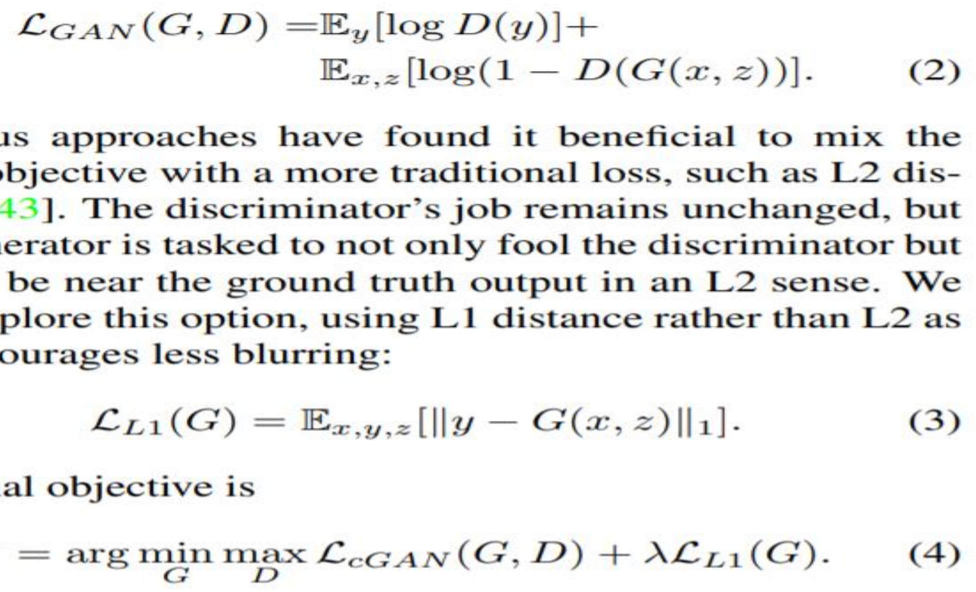
Image-to-Image Translation with Conditional Adversarial Networks

The Pix2Pix Generative Adversarial Network, or GAN, is an approach to training a deep convolutional neural network for image-to-image translation tasks.

The Conditional GAN, or cGAN, is an extension of the GAN architecture that provides control over the image that is generated, e.g. allowing an image of a given class to be generated. Pix2Pix GAN is an implementation of the cGAN where the generation of an image is conditional on a given image
The generator model is provided with a given image as input and generates a translated version of the image. The discriminator model is given an input image and a real or generated paired image and must determine whether the paired image is real or fake. Finally, the generator model is trained to both fool the discriminator model and to minimize the loss between the generated image and the expected target image.

As such, the Pix2Pix GAN must be trained on image datasets that are comprised of input images (before translation) and output or target images (after translation)

Loss Function:



$$\mathcal{L}_{GAN}(G, D) = \mathbb{E}_y[\log D(y)] + \mathbb{E}_{x,z}[\log(1 - D(G(x, z))]. \quad (2)$$

Previous approaches have found it beneficial to mix the GAN objective with a more traditional loss, such as L2 distance [43]. The discriminator's job remains unchanged, but the generator is tasked to not only fool the discriminator but also to be near the ground truth output in an L2 sense. We also explore this option, using L1 distance rather than L2 as L1 encourages less blurring:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|_1]. \quad (3)$$

Our final objective is

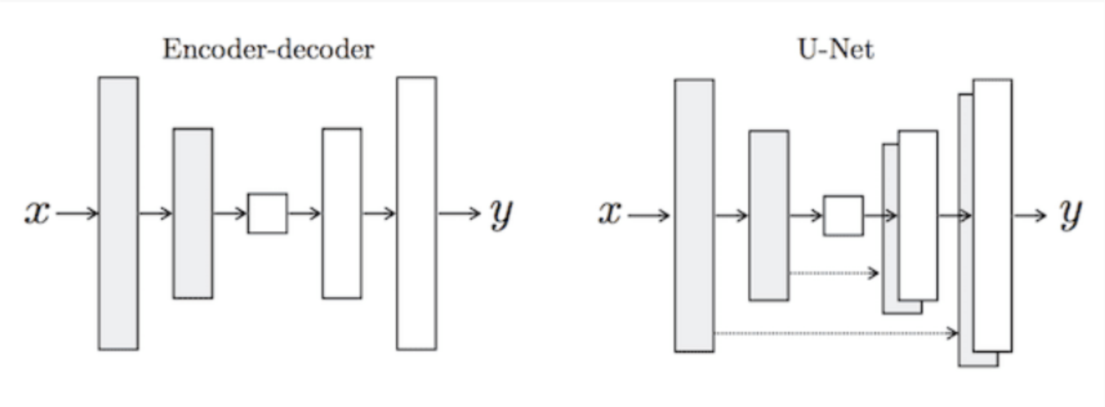
$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda\mathcal{L}_{L1}(G). \quad (4)$$

Structure:

U-Net Generator Model

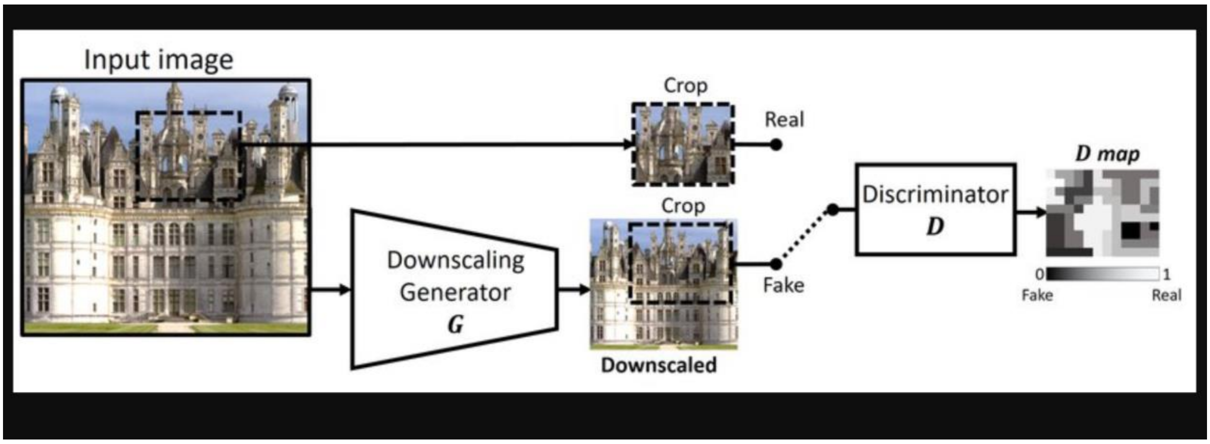
A U-Net model architecture is used for the generator, instead of the common encoder-decoder model.

The generator model takes an image as input, and unlike a traditional GAN model, it does not take a point from the latent space as input.
Instead, the source of randomness comes from the use of dropout layers that are used both during training and when a prediction is made.



Depiction of the Encoder-Decoder Generator and U-Net Generator Models.
Taken from: Image-to-Image Translation With Conditional Adversarial Networks.

PatchGAN Discriminator Model



The input to the discriminator model highlights the need to have an image dataset comprised of paired source and target images when training the model.

Unlike the traditional GAN model that uses a deep convolutional neural network to classify images, the Pix2Pix model uses a PatchGAN. This is a deep convolutional neural network designed to classify patches of an input image as real or fake, rather than the entire image.

Use Cases:

- Semantic labels <-> photo, trained on the Cityscapes dataset.
- Architectural labels -> photo, trained on Facades.
- Map <-> aerial photo, trained on data scraped from Google Maps.
- Black and White -> color photos.

- Edges -> photo.
- Sketch -> photo.
- Day -> night photographs.
- Thermal -> color photos.
- Photo with missing pixels -> inpainted photo, trained on Paris StreetView.

Dataset :
• Semantic labels↔photo, trained on the Cityscapes dataset [12].
•Architectural labels→photo, trained on CMP Facades
• Map↔aerial photo, trained on data scraped from Google Maps.

Cityscapes labels→photo 2975 training images from the Cityscapes training set.
Architectural labels→photo 400 training images .
Maps↔aerial photograph 1096 training images.
Thermal→color photos 36609 training images.
Day→night 17823 training images extracted from 91 webcams.
Edges→Handbag 137K Amazon Handbag images.

Similar models : cycle gan,lapgan,infogan.

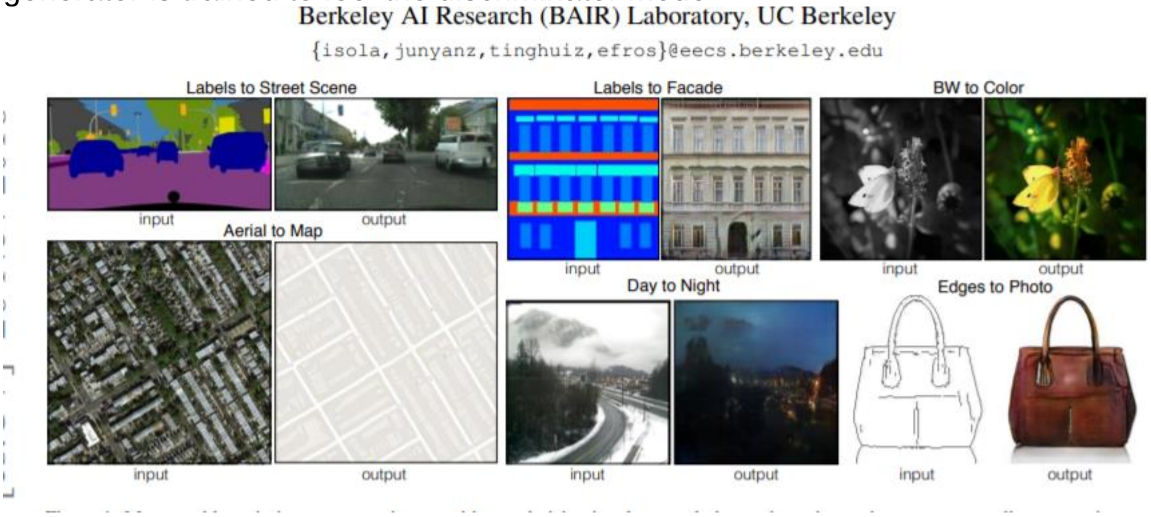
<u>Image Segmentation to Label Map</u>

Pix2Pix GAN is used for Image Segmentation to Label Map.

Details are as below:-

About:

Pix2Pix is a Generative Adversarial Network, or GAN, model designed for general purpose image-to-image translation.The GAN architecture is an approach to training a generator model, typically used for generating images. A discriminator model is trained to classify images as real (from the dataset) or fake (generated), and the generator is trained to fool the discriminator model.

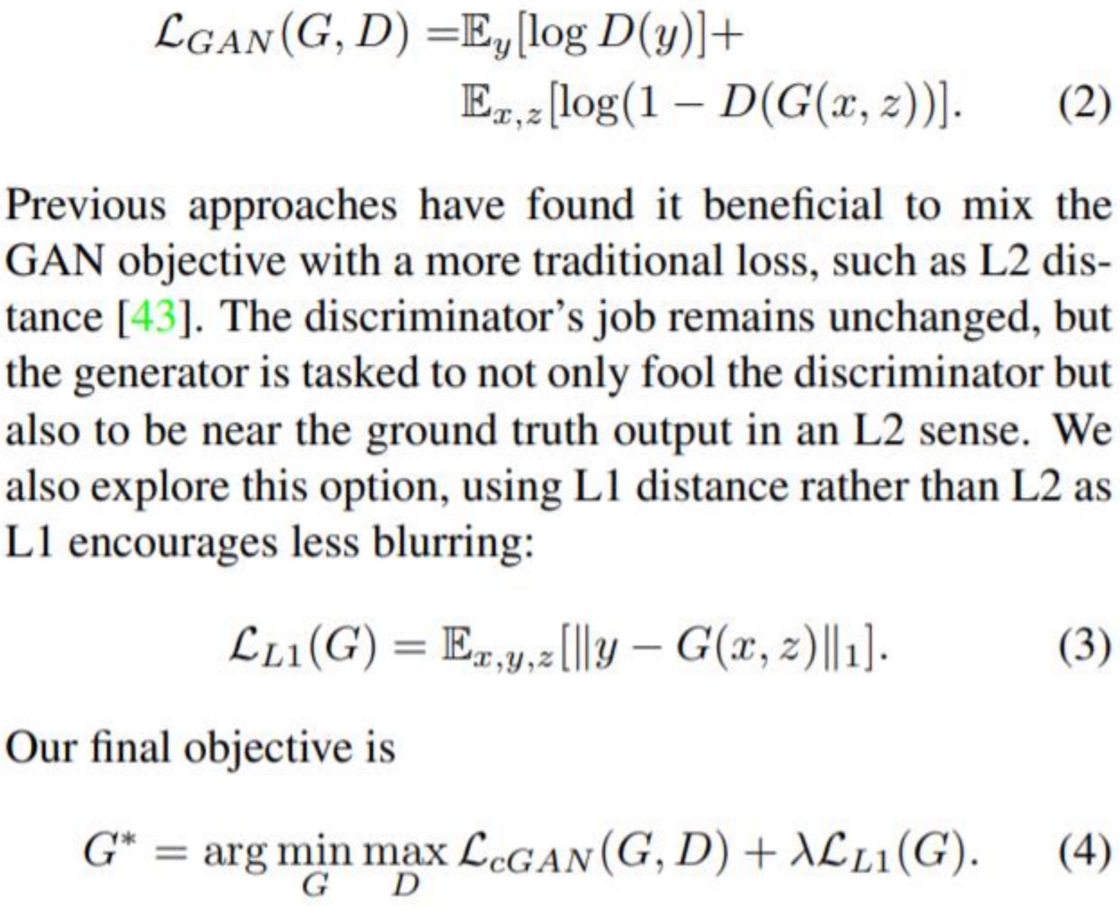


Berkeley AI Research (BAIR) Laboratory, UC Berkeley
{isola,junyanz,tinghuiz,efros}@eecs.berkeley.edu

The Conditional GAN, or cGAN, is an extension of the GAN architecture that provides control over the image that is generated, e.g. allowing an image of a given class to be generated. Pix2Pix GAN is an implementation of the cGAN where the generation of an image is conditional on a given image
The generator model is provided with a given image as input and generates a translated version of the image. The discriminator model is given an input image and a real or generated paired image and must determine whether the paired image is real or fake. Finally, the generator model is trained to both fool the discriminator model and to minimize the loss between the generated image and the expected target image.

As such, the Pix2Pix GAN must be trained on image datasets that are comprised of input images (before translation) and output or target images (after translation)

Loss Function:



$$\begin{aligned}\mathcal{L}_{GAN}(G,D) =&\mathbb{E}_y[\log D(y)]+\\ &\mathbb{E}_{x,z}[\log(1-D(G(x,z))].\end{aligned} \tag{2}$$

Previous approaches have found it beneficial to mix the GAN objective with a more traditional loss, such as L2 distance [43]. The discriminator's job remains unchanged, but the generator is tasked to not only fool the discriminator but also to be near the ground truth output in an L2 sense. We also explore this option, using L1 distance rather than L2 as L1 encourages less blurring:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1]. \tag{3}$$

Our final objective is

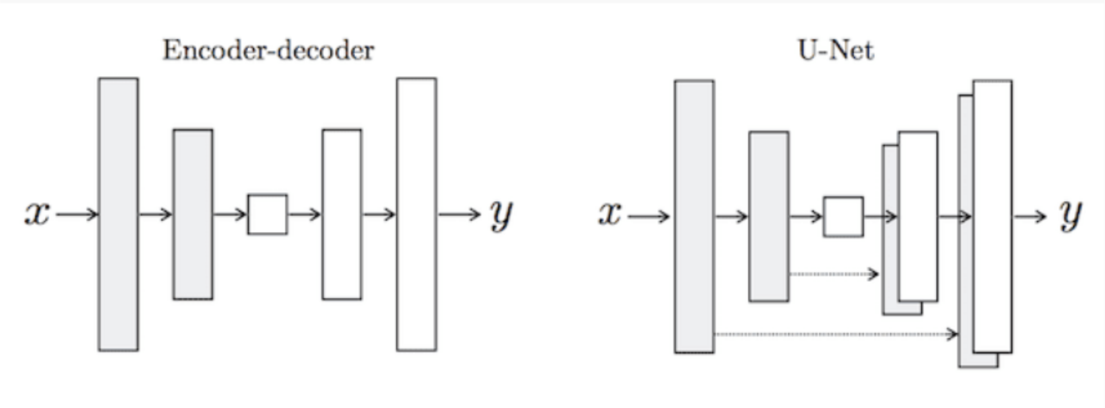
$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda\mathcal{L}_{L1}(G). \tag{4}$$

Structure:

U-Net Generator Model

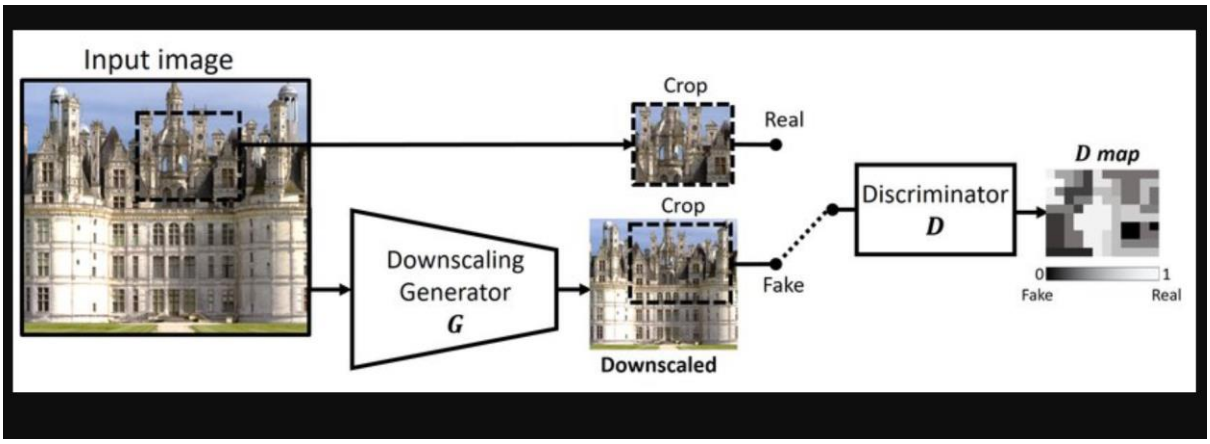
A U-Net model architecture is used for the generator, instead of the common encoder-decoder model.

The generator model takes an image as input, and unlike a traditional GAN model, it does not take a point from the latent space as input.
Instead, the source of randomness comes from the use of dropout layers that are used both during training and when a prediction is made.



Depiction of the Encoder-Decoder Generator and U-Net Generator Models.
Taken from: Image-to-Image Translation With Conditional Adversarial Networks.

PatchGAN Discriminator Model



The input to the discriminator model highlights the need to have an image dataset comprised of paired source and target images when training the model.

Unlike the traditional GAN model that uses a deep convolutional neural network to classify images, the Pix2Pix model uses a PatchGAN. This is a deep convolutional neural network designed to classify patches of an input image as real or fake, rather than the entire image.

Use Cases:

- Semantic labels <-> photo, trained on the Cityscapes dataset.
- Architectural labels -> photo, trained on Facades.
- Map <-> aerial photo, trained on data scraped from Google Maps.
- Black and White -> color photos.

- Edges -> photo.
- Sketch -> photo.
- Day -> night photographs.
- Thermal -> color photos.
- Photo with missing pixels -> inpainted photo, trained on Paris StreetView.

Face Generation:

Face generation is the task of generating (or interpolating) new faces from an existing dataset. These newly generated faces doesn't exist in real.

Face Generation GAN

Style Gan is preferred GAN architecture for face generation task.

Style GAN:

StyleGAN is a novel generative adversarial network introduced by Nvidia researchers in December 2018, and made source available in February 2019. StyleGAN depends on Nvidia's CUDA software, GPUs and Google's TensorFlow. The second version of StyleGAN, called StyleGAN2, was published on 5 February 2020.

The Style Generative Adversarial Network, or StyleGAN for short, is an extension to the GAN architecture that proposes large changes to the generator model, including the use of a mapping network to map points in latent space to an intermediate latent space, the use of the intermediate latent space to control style at each point in the generator model, and the introduction to noise as a source of variation at each point in the generator model.

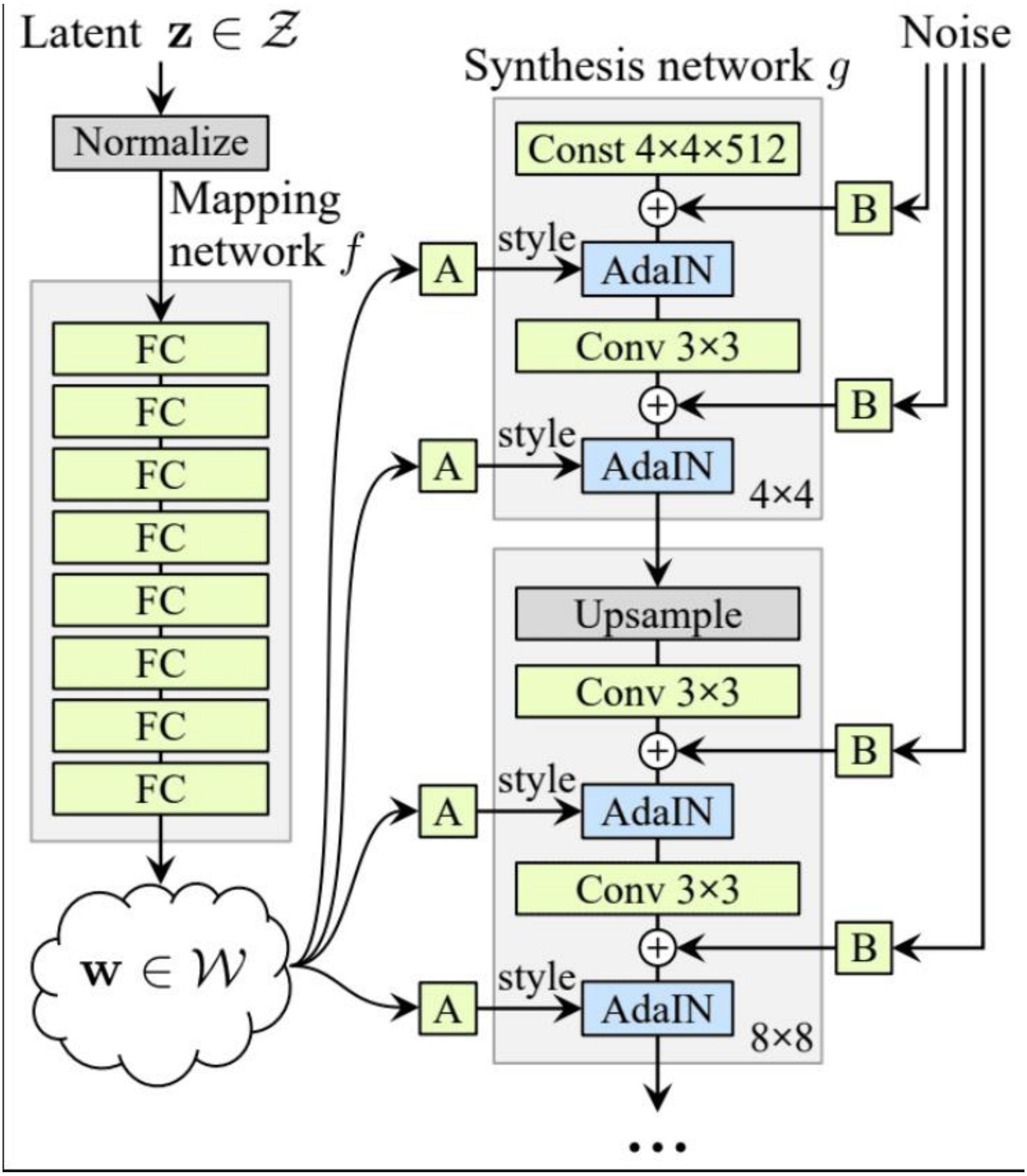
The resulting model is capable not only of generating impressively photorealistic high-quality photos of faces, but also offers control over the style of the generated image at different levels of detail through varying the style vectors and noise.

Architecture

The StyleGAN is described as a progressive growing GAN architecture with five modifications, each of which was added and evaluated incrementally in an ablative study.

The incremental list of changes to the generator are:

- Baseline Progressive GAN.
- Addition of tuning and bilinear upsampling.
- Addition of mapping network and AdaIN (styles).
- Removal of latent vector input to generator.
- Addition of noise to each block.
- Addition Mixing regularization.



1. Baseline Progressive GAN

The StyleGAN generator and discriminator models are trained using the progressive growing GAN training method.

This means that both models start with small images, in this case, 4×4 images. The models are fit until stable, then both discriminator and generator are expanded to double the width and height (quadruple the area), e.g. 8×8.

A new block is added to each model to support the larger image size, which is faded in slowly over training. Once faded-in, the models are again trained until reasonably stable and the process is repeated with ever-larger image sizes until the desired target image size is met, such as 1024×1024.

2. Bilinear Sampling

The progressive growing GAN uses nearest neighbor layers for upsampling instead of transpose convolutional layers that are common in other generator models. The first point of deviation in the StyleGAN is that bilinear upsampling layers are unused instead of nearest neighbor.

3. Mapping Network and AdaIN

Next, a standalone mapping network is used that takes a randomly sampled point from the latent space as input and generates a style vector. The mapping network is comprised of eight fully connected layers, e.g. it is a standard deep neural network.

4. Removal of Latent Point Input

The next change involves modifying the generator model so that it no longer takes a point from the latent space as input.Instead, the model has a constant 4x4x512 constant value input in order to start the image synthesis process.

5. Addition of Noise

The output of each convolutional layer in the synthesis network is a block of activation maps.Gaussian noise is added to each of these activation maps prior to the AdaIN operations. A different sample of noise is generated for each block and is interpreted using per-layer scaling factors.

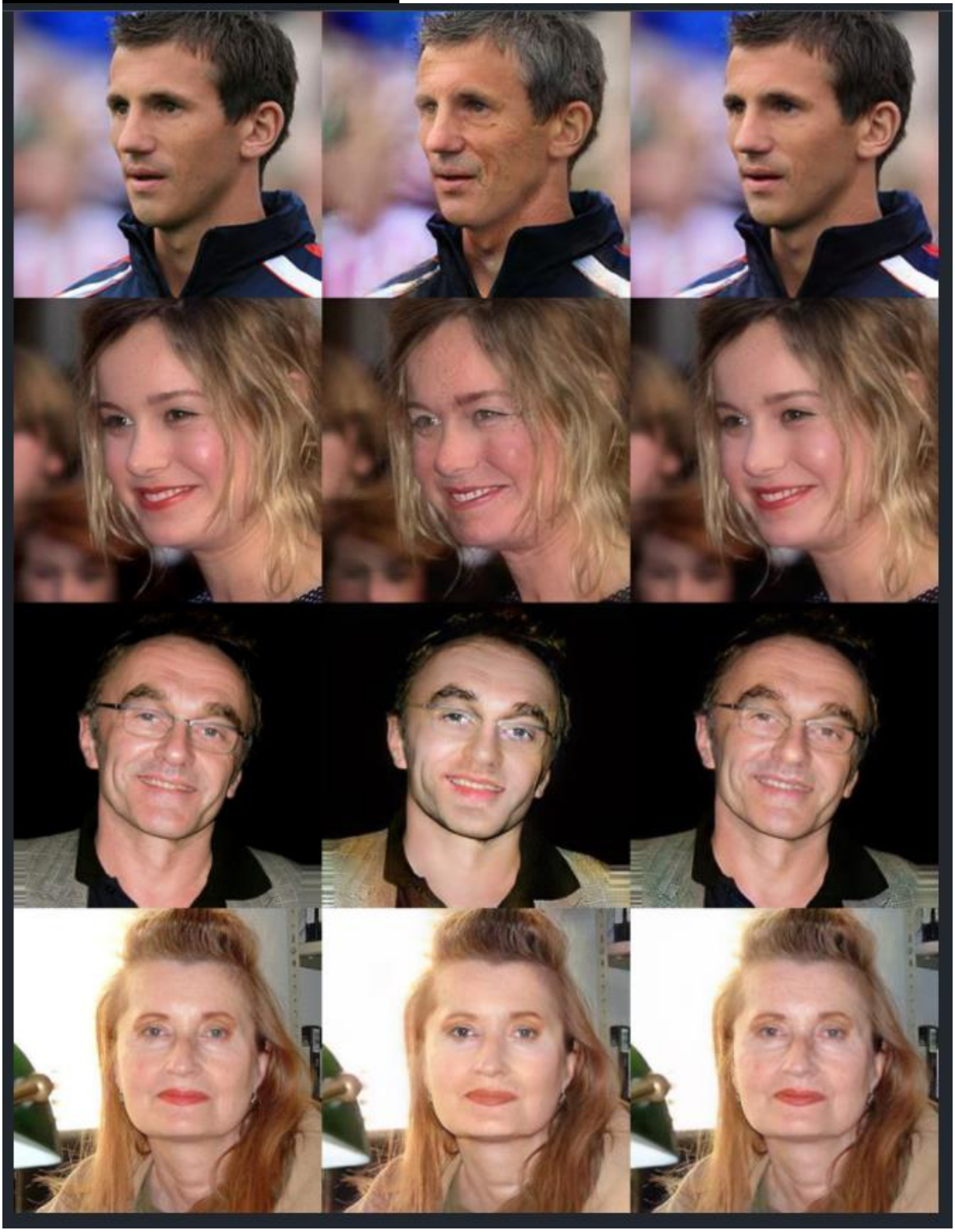
6. Mixing regularization

Mixing regularization involves first generating two style vectors from the mapping network.A split point in the synthesis network is chosen and all AdaIN operations prior to the split point use the first style vector and all AdaIN operations after the split point get the second style vector.

Datasets:

FFHQ dataset
CelebA-HQ dataset
Note: Till now no dataset found for indian faces

Face Aging GAN



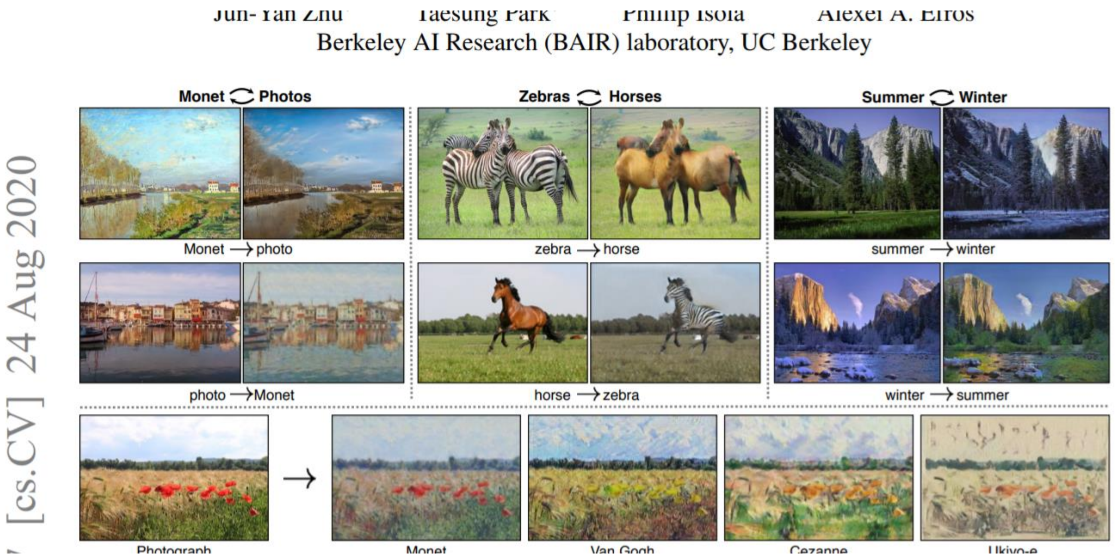
Facial aging refers to how the appearance of your face changes with age.

Model:

Cycle GAN is preferred model for face aging task. It can perform new to old and old to new face conversion.

CycleGAN

About:



Jun-Yan Zhu Taesung Park Phillip Isola Alexei A. Efros
Berkeley AI Research (BAIR) laboratory, UC Berkeley

There is a desire for techniques for training an image-to-image translation system that does not require paired examples. Specifically, where any two collections of unrelated images can be used and the general characteristics extracted from each collection and used in the image translation process.

For example, to be able to take a large collection of photos of summer landscapes and a large collection of photos of winter landscapes with unrelated scenes and locations as the first location and be able to translate specific photos from one group to the other.

This is called the problem of unpaired image-to-image translation.

One generator takes images from the first domain as input and outputs images for the second domain, and the other generator takes images from the second domain as input and generates images for the first domain. Discriminator models are then used to determine how plausible the generated images are and update the generator models accordingly.

The CycleGAN uses an additional extension to the architecture called cycle consistency. This is the idea that an image output by the first generator could be used as input to the second generator and the output of the second generator should match the original image. The reverse is also true: that an output from the second generator can be fed as input to the first generator and the result should match the input to the second generator.

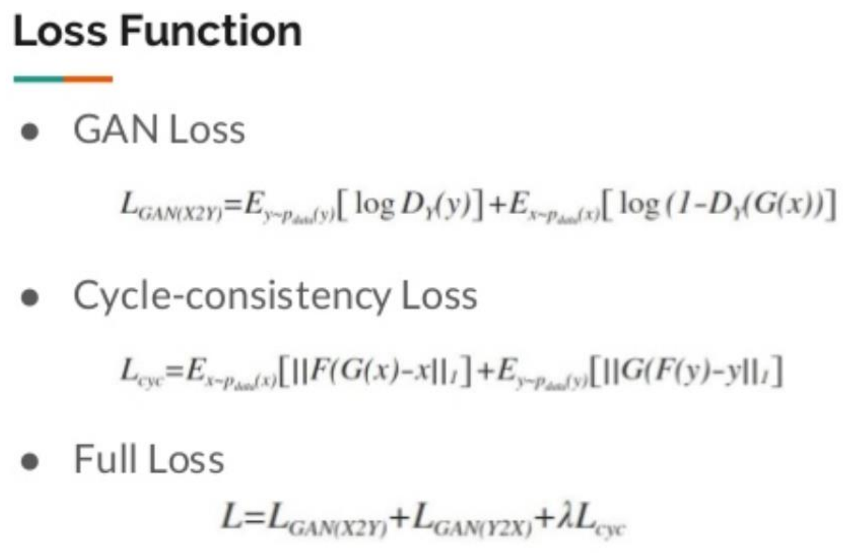
Cycle consistency is a concept from machine translation where a phrase translated from English to French should translate from French back to English and be identical to the original phrase. The reverse process should also be true.

The first GAN (GAN 1) will take an image of a summer landscape, generate image of a winter landscape, which is provided as input to the second GAN (GAN 2), which in turn will generate an image of a summer landscape. The cycle consistency loss calculates the difference between the image input to GAN 1 and the image output by GAN 2 and the generator models are updated accordingly to reduce the difference in the images.

This is a forward-cycle for cycle consistency loss. The same process is related in reverse for a backward cycle consistency loss from generator 2 to generator 1 and comparing the original photo of winter to the generated photo of winter.

Loss Function:

For cycle GAN actual loss function is combination of 2 Advercial Loss(we train two GAN models) and Cycle consistency loss.



Loss Function

- GAN Loss

$$L_{GAN(X2Y)}=E_{y\sim p_{data}(y)}[\log D_Y(y)]+E_{x\sim p_{data}(x)}[\log(1-D_Y(G(x))]$$

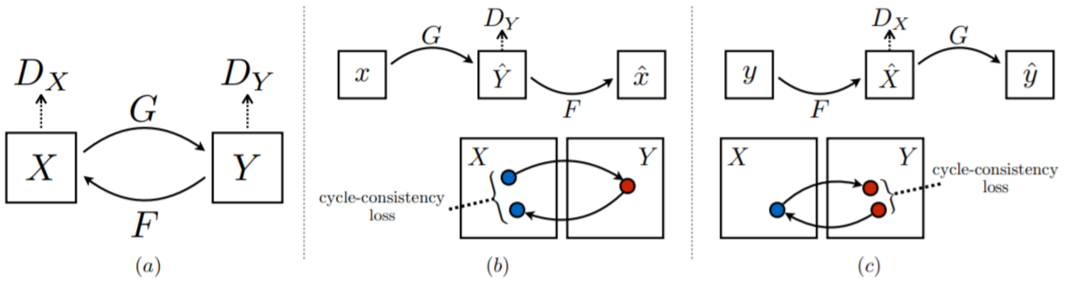
- Cycle-consistency Loss

$$L_{cyc}=E_{x\sim p_{data}(x)}[||F(G(x)-x||_1]+E_{y\sim p_{data}(y)}[||G(F(y)-y||_1]$$

- Full Loss

$$L=L_{GAN(X2Y)}+L_{GAN(Y2X)}+\lambda L_{cyc}$$

Structure:



Datasets:

IMDB-WIKI-500k+ face images with age and gender labels - contains 500k+ images of celebrities from IMDb and Wikipedia. The metadata of this dataset contains the date of birth of the person portrayed in the image and the date of which the image was taken.
Cross-Age Celebrity Dataset (CACD) - contains 163k+ images of 2,000 celebrities. "The images are collected from search engines using celebrity name and year (2004-2013) as keywords. We can therefore estimate the ages of the celebrities on the images by simply subtract the birth year from the year of which the photo was taken."

Similar Models:

PFA-GAN

Edge to image:

PIX 2 PIX GAN:

pix2pix uses a conditional generative adversarial network (cGAN) to learn a mapping from an input image to an output image.

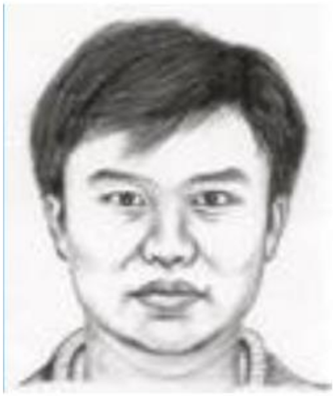
The network is composed of two main pieces, the Generator and the Discriminator. The Generator applies some transform to the input image to get the output image. The Discriminator compares the input image to an unknown image (either a target image from the dataset or an output image from the generator) and tries to guess if this was produced by the generator.

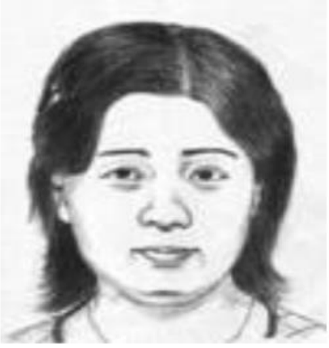
DataSet:

CUHK student data set, AR data set, XM2GTS data set

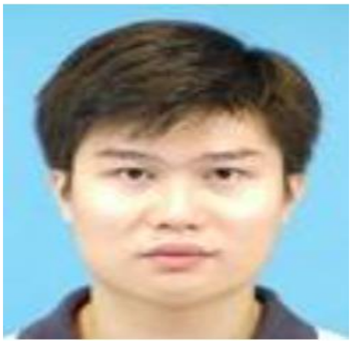
http://mmlab.ie.cuhk.edu.hk/archive/facesketch.html

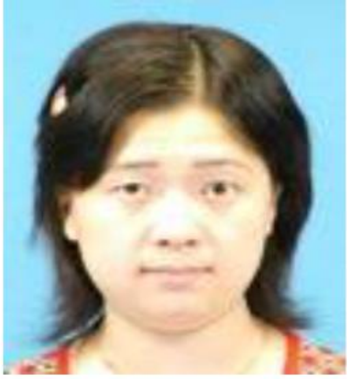
Input :



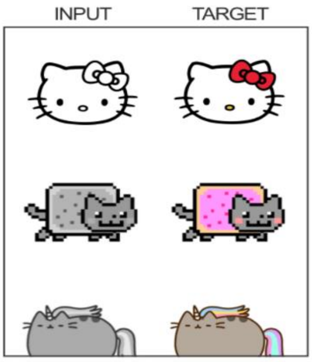


Output:



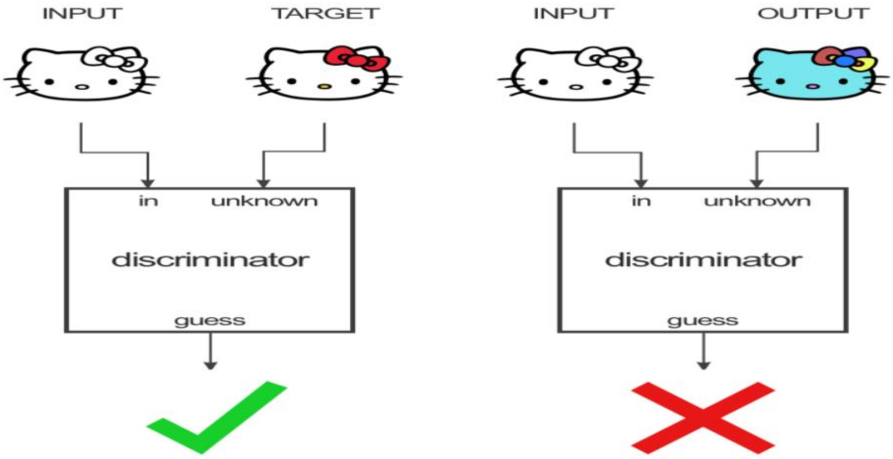


How PIX 2 PIX work:



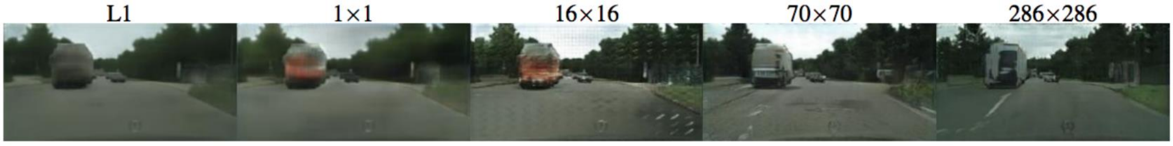
Discrimininator:

The discriminator is looking at the generator's colorization attempts and trying to learn to tell the difference between the colorizations the generator provides and the true colorized target image provided in the dataset.



PatchGAN Discriminator

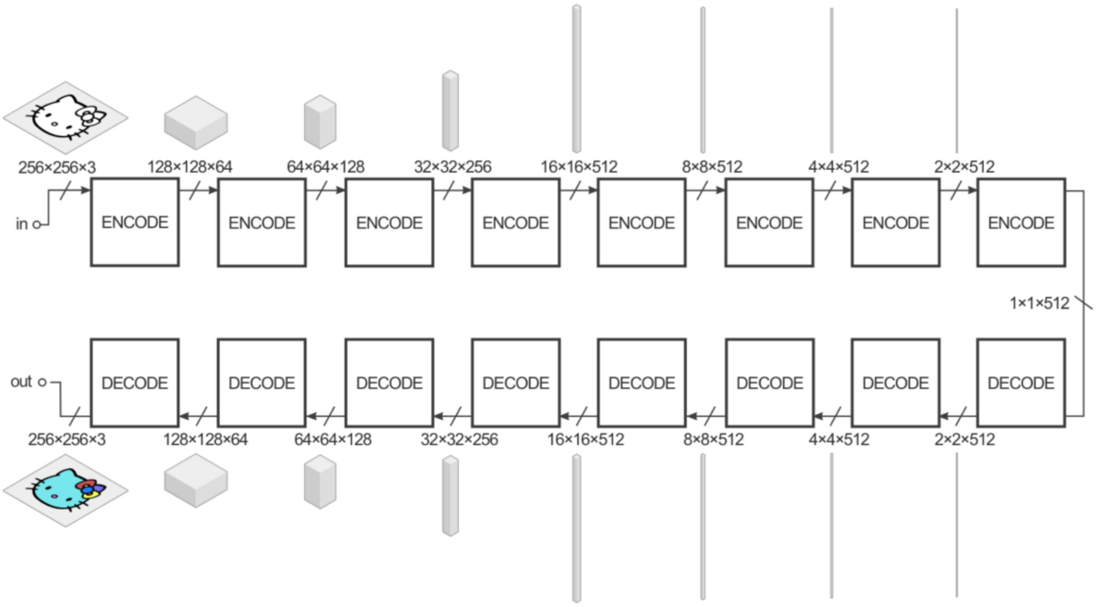
The PatchGAN discriminator used in pix2pix is another unique component to this design. The PatchGAN / Markovian discriminator works by classifying individual (N x N) patches in the image as “real vs. fake”, opposed to classifying the entire image as “real vs. fake”. The authors reason that this enforces more constraints that encourage sharp high-frequency detail. Additionally, the PatchGAN has fewer parameters and runs faster than classifying the entire image. The image below depicts results experimenting with the size of N for the N x N patches to be classified:



The 70 × 70 Patch is found to produce the best results

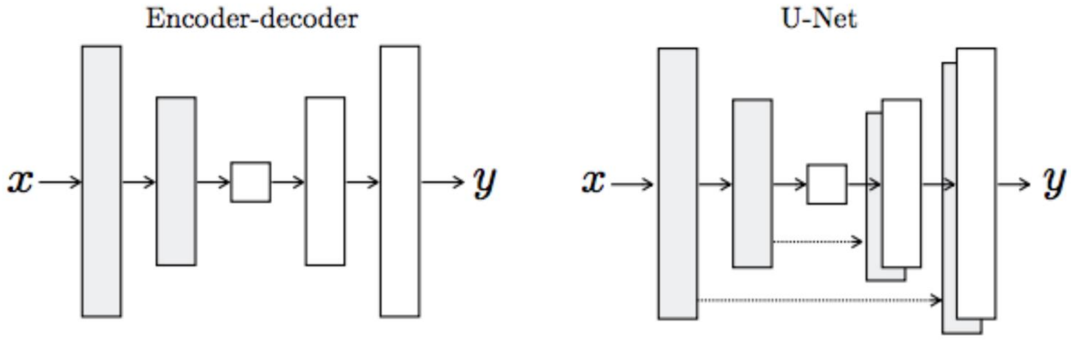
Generator:

The Generator has the job of taking an input image and performing the transform we want in order to produce the target image. An example input would be a black and white image, and we want the output to be a colorized version of that image. The structure of the generator is called an "encoder-decoder"



U-Net Generator:

The U-Net architecture used in the Generator of the GAN was a very interesting component of this paper. Image Synthesis architectures typically take in a random vector of size 100x1, project it into a much higher dimensional vector with a fully connected layer, reshape it, and then apply a series of de-convolutional operations until the desired spatial resolution is achieved. In contrast, the Generator in pix2pix resembles an auto-encoder.



ART Generation GAN

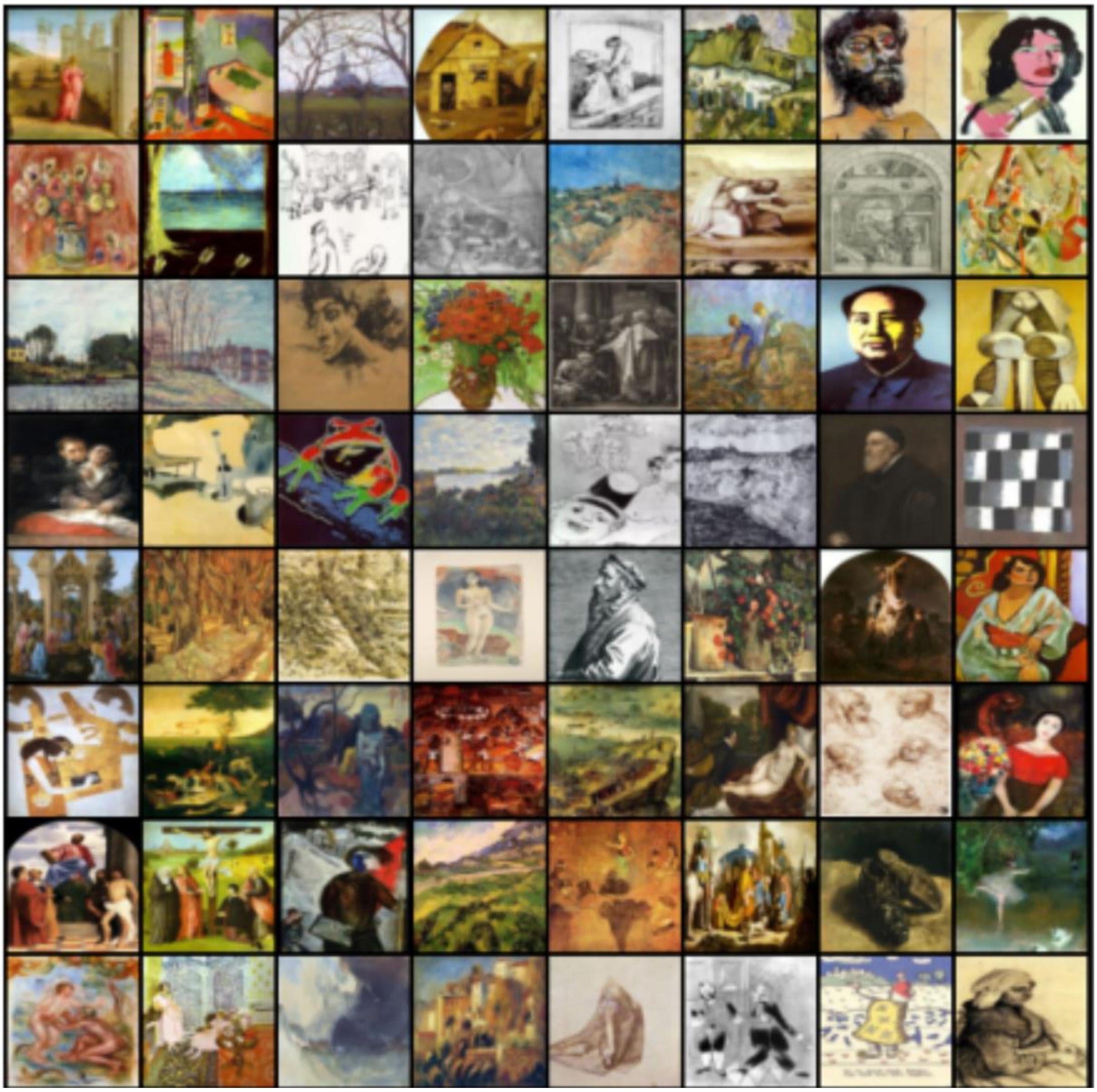
Data set:

best-artworks-of-all-timef-all

Input:

Latent noise

Output:



GANs consist of a pair of neural networks: a *Generator* and a *Discriminator*. They behave like Counterfeiter and a Cop where Generator (the Counterfeiter) is constantly creating fake data (in this case the paintings) and the Discriminator (the Cop) is constantly trying to catch the Counterfeiter.

In short, the Generator's job is to fool the Discriminator by creating realistic datasets which in this case are artworks. And Discriminator's job is to catch the real from fake artworks

Discriminator

The discriminator takes two sets of input; real images and fake images. Its job is to classify them properly whether the given image is fake or real. Since we are dealing with images we will use Convolution Neural Network(CNNs) for our discriminator's neural architecture.

Generator Network

The input to the generator is typically a vector or a matrix of random numbers (referred to as a latent tensor) which is used as a seed for generating an image. The generator will convert a latent tensor of shape (128, 1, 1) into an image tensor of shape 3 x 28 x 28. To achieve this, we'll use the ConvTranspose2d layer from PyTorch, which is performed to as a *transposed convolution* (also referred to as a *deconvolution*)

Pose Estimation

HUMAN pose estimation (HPE) aims to predict the locations of body joints from input images. It is fundamental for some other computer vision applications such as action recognition [1, 2, 3], human-computer interaction and video surveillance. The most recent methods for human pose estimation take advantage of convolutional neural networks (CNNs) to drastically improve the performance on standard benchmarks

Despite of these great progresses, there still exist some challenging cases, such as ambiguities caused by occluded body joints, invisible joints, nearby persons and clutter backgrounds, where even the state-of-the-art models may fail to predict the body joints correctly. The main reasons lie in: 1) these “hard” joints cannot be simply recognized based on their appearance features only; 2) these “hard” joints are not explicitly addressed during the training process.

One of the straightforward and efficient ways to handle these “hard” cases maybe combining body joints structural constraints into the training process to make the predicted pose plausible. GAN (Generative Adversarial Networks) [15] has been applied to learn the structural constrains of human body parts by adversarial training.

However, there are problems with existing GAN based pose estimation models. Since traditional convolutional GANs can only learn the spatially local constraints, previous GAN based HPE methods [10, 11] still cannot fully tackle these challenging cases when more complex body joints occlusion and crowded backgrounds occur.

Zhang and Goodfellow et al propose the SelfAttention Generative Adversarial Networks (SAGANs) [16], which introduce a self-attention mechanism into convolutional GANs. The self-attention module is complementary to convolutions and is capable of modeling long-range, multi-level dependencies across image regions. With self-attention, the discriminator can more accurately enforce complicated geometric constrains on the global image structure [16], which leads the generator to produce holistic consistent images.

Motivated by SAGANs, in this paper, we propose to apply self-Attention GAN to further improve the performance of human pose estimation. With attention mechanism in the framework of GAN, we can learn long-range body joints dependencies, therefore enforce the entire body joints H This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/. This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2019.2936709, IEEE Access VOLUME XX, 2017 9 structural constrains making all the body joints to be consistent. We evaluate the proposed approach on two HPE benchmarks, MPII and LSP. Experimental results show that our approach outperforms state-of-the-art methods.

HUMAN POSE ESTIMATION

Human pose estimation (HPE) is a challenging problem due to the large variations in configuration and appearance of body parts. Early works often tackle such problems by graphical models with handcrafted image features.

Similar as many other vision tasks, the progress on human pose estimation has been greatly advanced by deep learning , since Convolutional Neural Networks (CNNs) have the powerful ability to learn rich convolutional feature representations. Before CNNs were applied for HPE, the performance of previous works on the MPII benchmark [22] was only about 40% PCKh@0.5 [. CNNs pioneer works surprisingly improve it to about 80% . During the later three years, till now, it has achieved to more than 90% [8, 9, 10, 14]. The mAP (mean Average Precision) metric on more recent and challenging COCO human pose benchmark [23] has been increased from 60.5 (COCO 2016 Challenge winner to 72.1 and recently 78.1 (COCO 2017 and 2018 Challenge winner

Single person pose estimation.

DeepPose [4] is the first deep learning based approach for human pose estimation, which takes pose estimation as a body keypoints regression problem using Convolutional Neural Networks. Latter methods mostly predict heatmaps that characterize the probabilities of each keypoint at different locations [5]. The exact location of a keypoint is further estimated by finding the maximum in an aggregation of heatmaps. Heatmap-based methods better leverage the distributed properties of convolutional networks and are more suitable for training human pose estimation models.

Multi-person pose estimation.

The more practical problem is multi-person pose estimation, which is to estimate poses of multiple people in one image. There are two types of methods for multi-person pose estimation, bottom-up and top-down. Bottom-up methods first locate keypoints for all persons in the image and then group joints candidates for each person. Such as, DeepCut, DeeperCut [32, 33] and Openpose [7]. DeepCut and DeeperCut [32, 33] use CNNs (VGG [35] or ResNet [27]) to generate keypoint candidates and then run integer linear programming (ILP) to group them for each person. Cao et al. propose the Openpose [7], which is based on CPM [6] to simultaneously learn multi-person joints locations and their associations via Part Affinity Fields (PAFs), and then uses a greedy algorithm to group the joints that belong to the same person.

Pose tracking.

The more challenging task is simultaneous pose estimation and tracking [13, 39] or pose estimation in videos [40, 41]. Luo et al. adopt Long Short-Term Memory (LSTM) to impose geometric consistency among video frames while using CPM [6] to estimate person pose in images. 3D human pose estimation is also very important for practical applications, such as virtual reality or augmented reality [42, 43]. But 3D human pose estimation is based on 2D. Usually, 2D pose must be estimated first and then extended to 3D.

In summation: (1) Recent state-of-the-art human pose estimation methods are either improved hourglass networks [9, 10, 14, 84] , or take ResNet as their backbone [12, 13, 44]; (2) Among these tasks, 2D single person pose estimation is the basis. In this paper, we focus on 2D single person pose estimation from RGB images.

GENERATIVE ADVERSARIAL NETWORKS

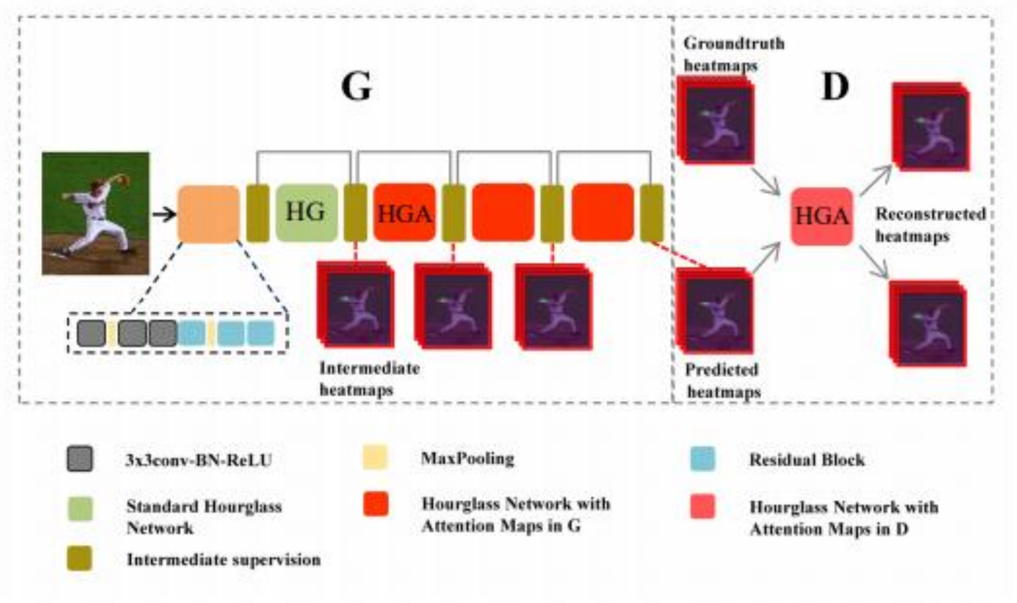
Recently, Generative Adversarial Networks (GAN) has been applied to various computer vision tasks, such as image super-resolution [59], image inpainting [60], object detection [61], person image synthesis [62], person Re-identification [63], and human pose estimation

[10, 11]. GAN is first proposed by Goodfellow et al. [15], which can generate natural images such as human faces and indoor scenes. It consists of generator and discriminator. The generator generates images to fool the discriminator, while the discriminator tries to distinguish the fake one from the real. In this way, the adversarial training can help generator to improve its product increasingly. The training of GANs may be unstable and sensitive to the choice of hyper-parameters

Researches on GAN may be considered mainly from three perspectives: (1) some works try to improve the training of GAN; (2) some works shine light on GAN theoretic analysis; (3) most of the works exploit various GAN applications.

Initially, GAN is used to generate synthetic images from input noises [15]. With rapid development in recent years, GAN has been able to generate amazing perfect images. Zhu et al. propose the state-of-the-art CycleGAN [54] to learn to translate an image from a source domain to a target domain in the absence of paired examples. CycleGAN learns bidirectional mappings between source and target domain with adversarial and cycle consistency losses to enforce the translation to be consistent. Bansal et al. propose the Recycle-GAN [55] for unsupervised video retargeting that enables the transfer of sequential content from one domain to another while preserving the style of the target domain. StarGAN [56] can perform image-to-image translations for multiple domains using only a single model. GANimation [57] enables continuous facial animation. Vid2vid [58] implements Video-to-Video Synthesis with GAN.

ATTENTION MODELS



Attention mechanism allows the model to differentiate irrelevant information so as to focus on the most relevant part of images or features as needed. Attention mechanism has been proven effective and successfully applied in many computer vision and natural language processing tasks e.g. image classification and action recognition image super-resolution , object detection .

Some recent works introduce attention mechanism into GAN for image synthesis , object transfiguration or face attribute editing , attention networks lead the generator to pay important attention to specific image regions

the SelfAttention GAN (SAGAN) which can model long-range dependencies for image generation tasks. The self-attention module calculates response at a position as a weighted sum of the features at all positions. Armed with self-attention, the discriminator can ensure detailed features in distant portions of the images to be consistent with each other. That is, the discriminator can more accurately enforce complicated geometric constraints on the global image structure. Note that, self-attention mechanism can learn long-range dependencies, but convolutions can only model local dependencies with local receptive fields.

As mentioned above, although human pose estimation has been significantly advanced by deep learning, still, all the difficulties lie in occlusion, overlapping with other people, or clutter background. In such cases, the model may find similar features which belong to the background or another person. So, body structural constraints are needed. Recent works try to improve the performance of human pose estimation by refinements , which are shown to be efficient, since such refinement processes are indeed to learn structural constraints of human body joints.

The generated poses can be refined by GAN , in which the discriminator checks the structural constraints of human body parts and distinguish implausible poses to guide and refine the generator training. But there is no attention mechanism in discriminator or generator. As stated in , the self-attention mechanism is powerful to model longrange dependencies in the feature maps. It is complementary to convolutions, which only models local dependencies with local receptive fields. So Self-GAN cannot fully learn the whole body structural constraints, which will be important in cases of occlusion, invisible joints or crowd background to ensure plausible poses.

HUMAN POSE ESTIMATION WITH SELF-ATTENTION GENERATIVE ADVERSARIAL NETWORKS

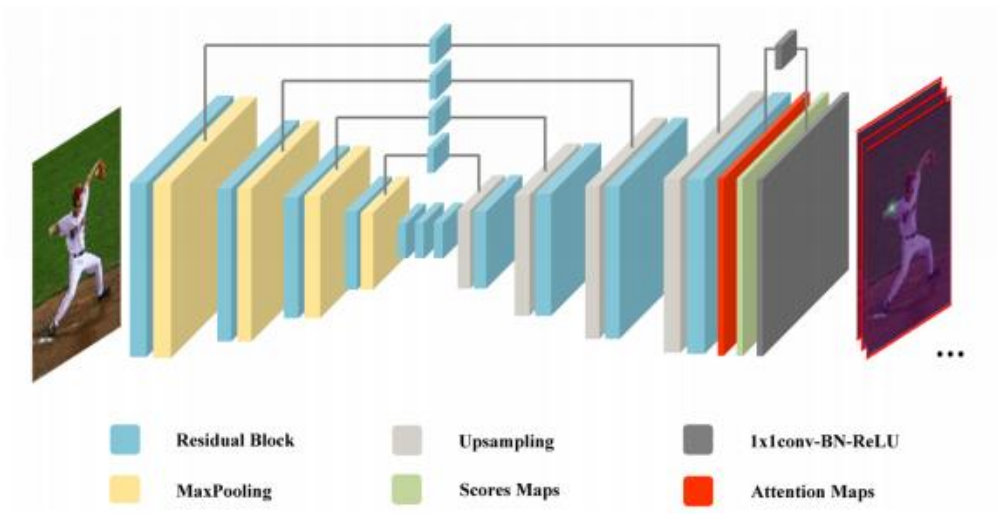


FIGURE 2. Overview of our Generator framework. We show one Hourglass network with self-attention architecture. The attention maps are generated by Self-Attention Residual Module (SARM) (See FIGURE 4 for more details).

THE NETWORK ARCHITECTURE

The framework is illustrated in Fig. 1. The model consists of two networks, the generator G and the discriminator D. Both use Hourglass networks as their backbone. Hourglass networks are fully convolutional networks constructed with residual blocks and conv-deconv architecture

. The generator generates heatmaps that indicate the confidence score at every location for all the body joint keypoints. The discriminator reconstructs both the predicted heatmaps and the ground truth heatmaps and distinguishes real from fake ones by adversarial training.

GENERATOR

We use Hourglass networks as the generator. Following previous works [8, 9], the input images are first warped to the same resolution of 256×256 and then fed into Hourglass network. The network starts with a 7×7 convolution layer with stride 2, followed by a residual module and a round of max pooling to reduce the resolution from 256 to 64. So the highest and the final output resolution is 64×64. Then, multiple hourglass modules are stacked to predict the body joint heatmaps. All residual modules output 256 features. The repeated bottom-up (from high resolutions to low resolutions), top-down (from low resolutions to high resolutions) structure together with skip connections allow processing features across all scales and capturing various spatial relationships associated with different body joints. Intermediate supervision at the end of each stack is also critical to the network’s final performance.

We use 4-stack hourglass networks as generator in our experiments. The first one is the standard hourglass network, and the next there hourglass networks are integrated with attention modules. One of the hourglass networks with selfattention architectures is shown in Fig. 2. We design a new Self-Attention Residual Module (SARM) by adding the selfattention structure into residual module, as shown in Fig. 4 (a) and (b). In generator, we use the attention structure of Fig. 4 (b) that is SARM-B. Indeed we have tried several different forms of attention modules and put them at the different parts of hourglass network. Finally we adopt the attention strategy by putting the attention module at the end of the hourglass networks (see Fig. 1 and Fig. 2), which can efficiently improve the performance.

2) DISCRIMINATOR

The framework of the proposed discriminator is illustrated in Fig. 3. We use 1-stack Hourglass network as the discriminator. In standard Hourglass networks [8], the building blocks are residual modules. In this work, we introduce self-attention mechanism into the discriminator. We utilize SARMs (that is SARM-A or SARM-B, See FIGURE 4 for more details) as the skip connections to connect blocks with the same semantic meanings, that is, the blocks in bottom-up and top-down processing with the same resolution scale of feature maps.

3) VARIANTS OF SARM STRUCTURE

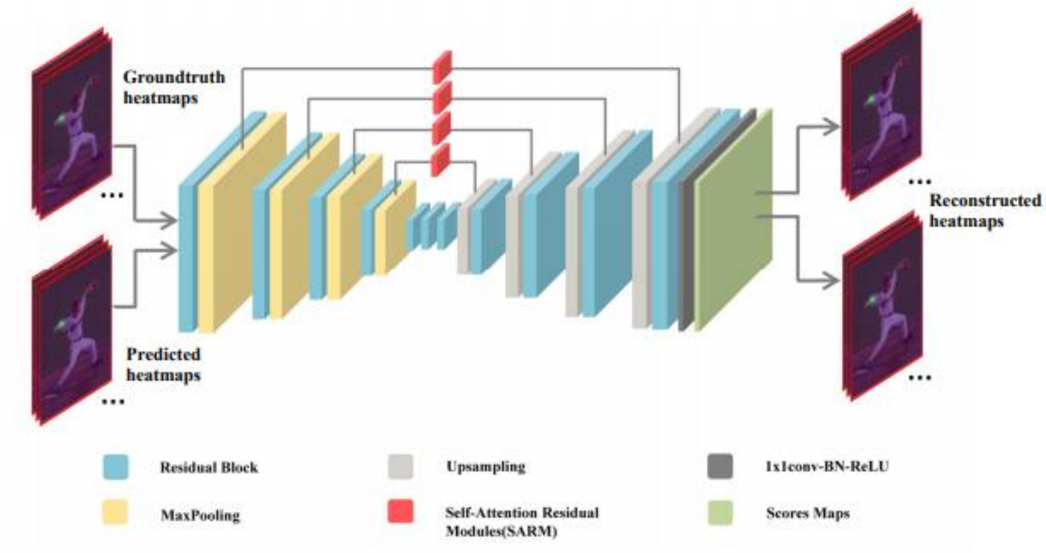


FIGURE 3. Overview of our Discriminator framework. The Discriminator is a single Hourglass network, in which the skip connections along blocks with the same semantic meanings adopt Self-Attention Residual Modules (SARMs).

Self-attention mechanism [16] calculates response at a position as a weighted sum of features at other locations, which is a good balance between modeling long-range dependencies and computational efficiency. So, SelfAttention can be complementary to convolutions and more suitable to capture widely separated spatial long-range multi-level dependencies among body joints in human pose estimation problem.

The ground-truth heatmaps and generated heatmaps together with the corresponding input image are all fed into discriminator. The discriminator reproduces a new set of heatmaps. By adversarial training, the discriminator checks that features in different positions of the heatmaps are consistent with each other. Armed with self-attention, the discriminator can more accurately enforce global geometric structural constraints on generated human pose.

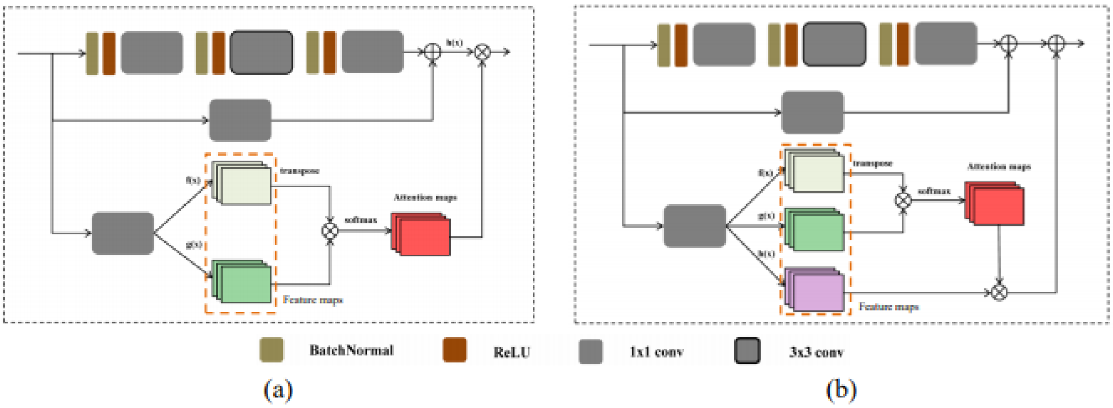


FIGURE 4. Our proposed Self-Attention Residual Modules (SARMs). (a) SARM-A produces the attention map and then matrix multiply it with the output of the residual module, while (b) SARM-B first matrix multiply the attention map with the input and then add such result to the output of the residual module. The ⊗ denotes matrix multiplication and ⊕ denotes matrix summation. The softmax operation is performed on each row.

EXPERIMENTS A. DATASETS AND IMPLEMENTATION DETAILS

We evaluate our method on two widely used human pose estimation benchmarks, Leeds Sports Pose (LSP) [76] and MPII Human Pose Dataset

The LSP and its extended dataset [76] contain total 12k images with poses in various sports. 11000 images are used for training and 1000 for testing. Each image is annotated with 14 keypoint locations. The center and scale of annotated person are calculated to be used in training. The dataset is challenging because of its noisy labels and various poses.

MPII dataset [22] contains about 25k images and over 40k annotated people, which covers a wide range of human activities. Each image may contain multiple persons. We focus on single person pose estimation. There exist some missing annotations for some persons. We just consider persons with full annotations. We follow previous works [9] to split the training set into train and validation subsets. Train set has 14679 images with 22246 persons, validation set has 2729 images with 2958 persons, and test set has 6619 images with 11731 persons. The test set has no annotations. Each h person is annotated with 16 joints, the center and scale.

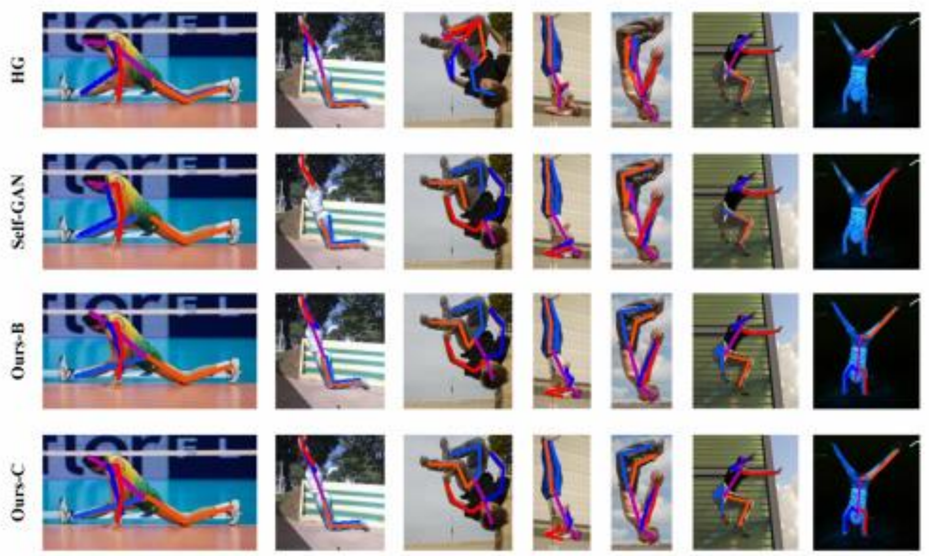


FIGURE 5. Qualitative comparisons on LSP.

FIGURE 5. Qualitative comparisons on LSP.

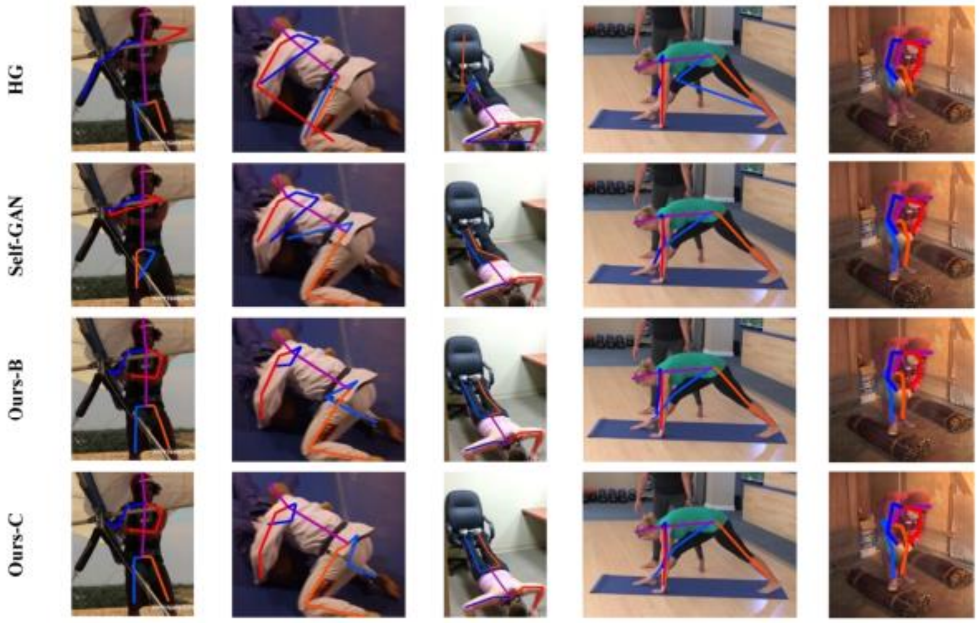


FIGURE 6. Qualitative comparisons on MPII.

We do data augmentation following previous works [8, 9, 10]. All input images are 256 × 256 cropped from resized images according to the annotated human body centers and scales. An input image is randomly flipped horizontally, rotated by an angle in [-30, 30] degrees, and scaled with factors in [0.75, 1.25].

We implement our models using Torch7 [77] deep learning libraries. The networks are optimized by RMSprop algorithm with a batch size of 6 for 200 epochs. Training is performed on a server with 16GB NVIDIA Tesla P100 GPU. The learning rate is initialized as 1 × 10 and dropped by 5 at 40th, 60th and 80th epoch. Our model takes about 4 days to train on the training set.

In the GAN framework, the generator and discriminator are all stacked Hourglass networks. The generator is responsible for predicting poses, while the discriminator acts to enforce structural constraints of human body joints to refine the poses. Since modeling long-range dependencies among image regions just by convolutions with local receptive fields is not efficient and enough, so we further introduce self-attention mechanism into the generator and the discriminator. The self-attention mechanism allows modeling longrange dependencies among body joints. With self-attention, the generator can pay more attention to salient body joints, while the discriminator can check that body joints in distant portions of the body are consistent with each other. So entire body joints geometry constraints can be further enforced during training of the generator, which will be important in cases of occlusion, invisible joints or crowd background to ensure plausible poses.