STTP- Multi-Modal Generative AI Generative Adversarial Networks & Its Variants

Praveen Kumar Chandaliya

Sardar Vallabhbhai National Institute of Technology, Surat
Department of Artificial Intelligence
pkc@aid.svnit.ac.in

December 18, 2024

Outline

Fundamentals of Convolution Neural Network

1. Convolutional Layer Size Calculate Formula

$$K = \frac{(W - F + 2P)}{S} + 1 \tag{1}$$

2. Pooling Layer Size Calulate Formula

$$K = \frac{(W - F)}{S} + 1 \tag{2}$$

- W: Input volume size
- F: Filter Size
- S: Stride (Sampling in the Case of Pooling)
- P: Padding

Convolution Neural Network Architecture

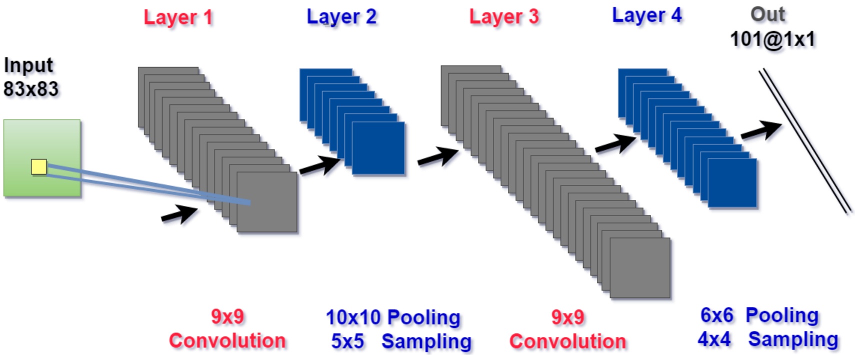


Figure 1: Convolution Neural Network

Convolution Neural Network Architecture

Input (W=83, F=9, P=0, S=1) => Layer 1 Size : 32 @ 75 x 75

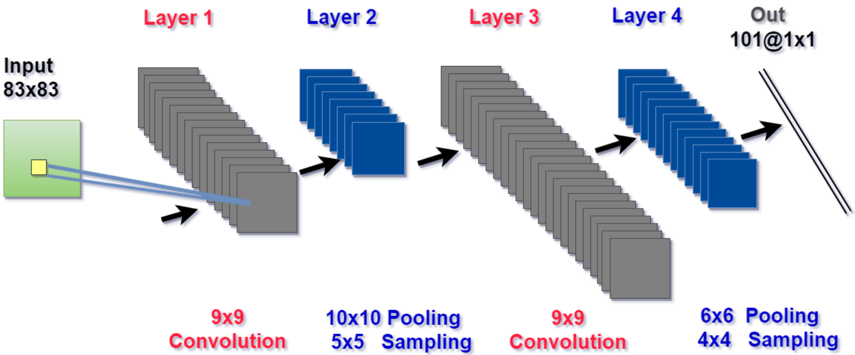


Figure 2: Convolution Neural Network

Convolution Neural Network Architecture

INPUT (W=75, F=10, P=0, S=5) => LAYER 2: 64 @ 14 x 14

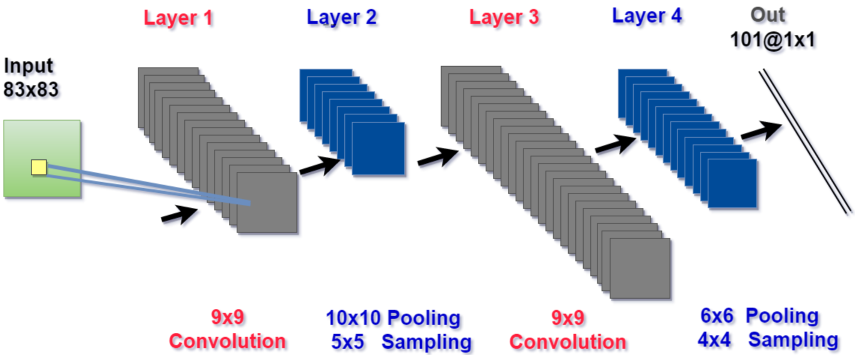


Figure 3: Convolution Neural Network

Convolution Neural Network Architecture

INPUT (W=14, F=9, P=0, S=1) => LAYER 3: 128 @ 6 x 6

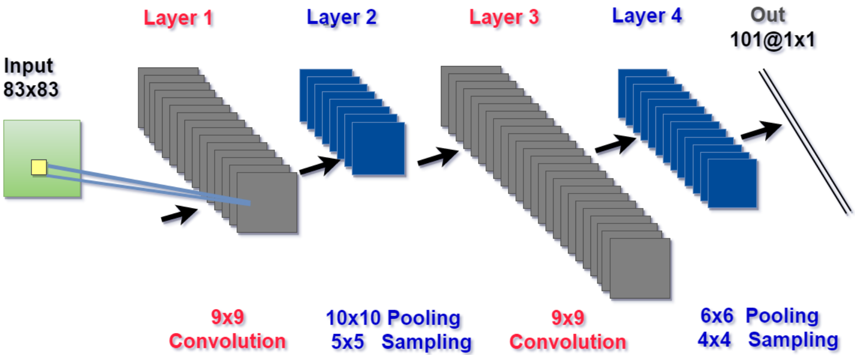


Figure 4: Convolution Neural Network

Convolution Neural Network Architecture

INPUT (W=6 F=6 P=0 S=4) => LAYER 4: 256 @ 1 x 1

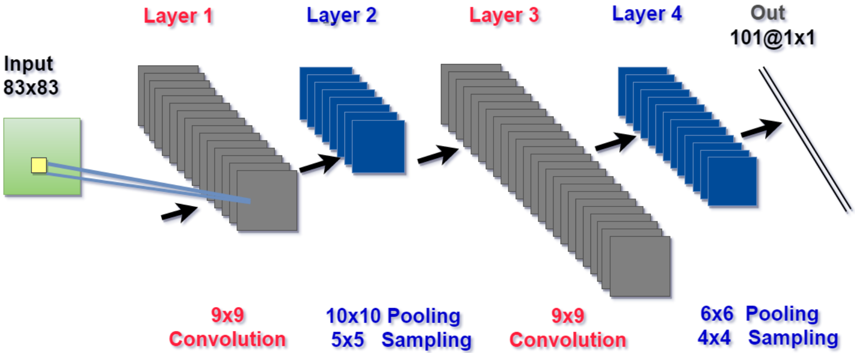


Figure 5: Convolution Neural Network

Convolution Autoencoder

Convolution Autoencoder (CAE) are Convolution neural network (CNN) trained to reproduce its input at the output layer.

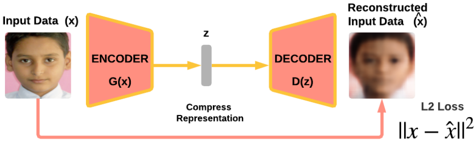


Figure 6: Autoencoder Block diagram

Pros:

- CAE do dimensionality reduction and can also remove distortions from images (both gray scale and color).

Cons:

- CAE not being able to determine what information is relevant.
- CAE is that it often has "holes" in its latent space. decoded a random vector, it might not result in anything realistic.

Convolution Autoencoder With Face Dataset

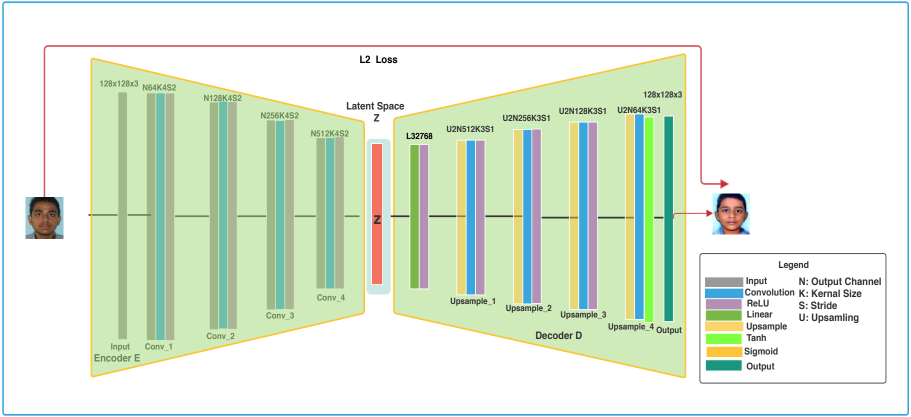


Figure 7: Convolution autoencoder architecture [1]

[1] Praveen Kumar Chandaliya, Neeta Nain, Conditional Perceptual Adversarial Variational Autoencoder,ICB2019,IEEE.

Convolution Autoencoder implementation on colab **STTP_CAE_Face.ipyb** (ChildGAN) [1]

Convolution Autoencoder Implementation Result

(a) Morph Data

(b) epoch-1

(c) epoch-15

(d) epoch-30

Figure 8: Convolution Autoencoder Implementation Results

Praveen Kr Chadnaliya, Vardhman, Mayank Harjani, Neeta Nain, “SCDAE: Ethnicity and Gender Alteration on CLF and UTKFace Dataset” CVIP 2019

SCDAE: Ethnicity and Gender Alteration

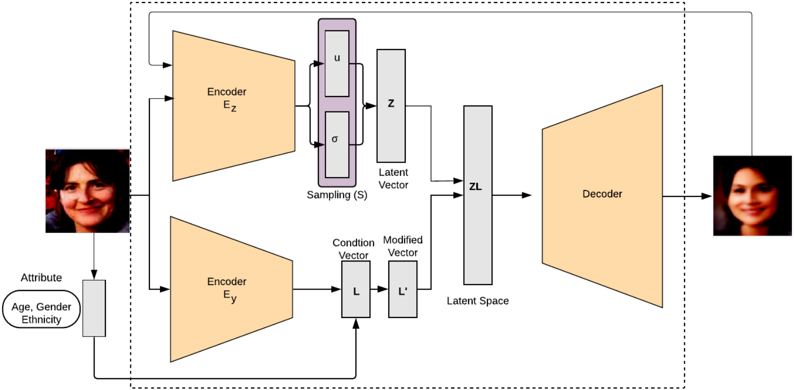


Figure 9: Sampling and Condition based Deep AutoEncoder [2]

---

[2] Praveen, Vardhman,Mayank, Neeta Nain, SCDAE:Ethnicity and Gender Alteration on CLF and UTKFace Dataset,CVIP2019

SCDAE: Ethnicity and Gender Alteration

Operations	Kernel	Stride	Filters	BN	Activation	Output Shape
*Conv*1	5×5	2×2	32	No	LReLU	$64 \times 64 \times 32$
*Conv*2	5×5	2×2	64	No	LReLU	$32 \times 32 \times 64$
*Conv*3	5×5	2×2	128	No	LReLU	$16 \times 16 \times 128$
*Conv*4	5×5	2×2	256	No	LReLU	$8 \times 8 \times 256$
*Conv*5	5×5	2×2	512	No	LReLU	$4 \times 4 \times 512$
*Conv*6	5×5	2×2	1024	No	LReLU	$2 \times 2 \times 1024$
Flatten	-	-	-	No	-	4096
$Dense_1$	-	-	-	Yes	LReLU	60
$Dense_2$	-	-	-	Yes	LReLU	60
Sampling	-	-	-	Yes	LReLU	60

Table 1: Encoder E_z network

Operations	Kernel	Stride	Filters	BN	Activation	Output Shape
*Conv*1	5×5	2×2	32	No	LReLU	$64 \times 64 \times 32$
*Conv*2	5×5	2×2	64	No	LReLU	$32 \times 32 \times 64$
*Conv*3	5×5	2×2	128	No	LReLU	$16 \times 16 \times 128$
*Conv*4	5×5	2×2	256	No	LReLU	$8 \times 8 \times 256$
*Conv*5	5×5	2×2	512	No	LReLU	$4 \times 4 \times 512$
*Conv*6	5×5	2×2	1024	No	LReLU	$2 \times 2 \times 1024$
Flatten	-	-	-	No	-	4096
Dense	-	-	-	Yes	LReLU	40

Table 2: Encoder E_y network

Operations	Kernel	Stride	Filters	BN	Activation	Output Shape
Dense	-	-	-	No	-	32768
Reshape	-	-	-	No	-	$4 \times 4 \times 2048$
DConv1	5×5	2×2	1024	Yes	LReLU	$8 \times 8 \times 1024$
DConv2	5×5	2×2	512	Yes	LReLU	$16 \times 16 \times 512$
DConv3	5×5	2×2	256	Yes	LReLU	$32 \times 32 \times 256$
DConv4	5×5	2×2	128	Yes	LReLU	$64 \times 64 \times 128$
DConv5	5×5	2×2	64	Yes	LReLU	$128 \times 128 \times 64$
DConv6	5×5	2×2	3	No	Tanh	$128 \times 128 \times 3$

Table 3: Decoder Network Architecture

Gender and Ethnicity alteration results

Figure 10: Gender alteration Male to Female

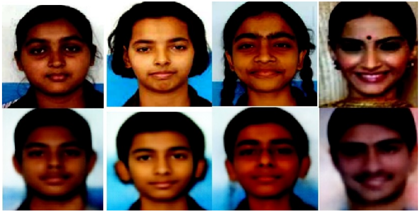


Figure 11: Gender alteration Female to Male

Figure 12: Asian ethnicity

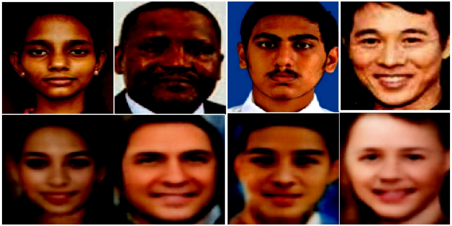


Figure 13: Indian ethnicity

Age alteration results

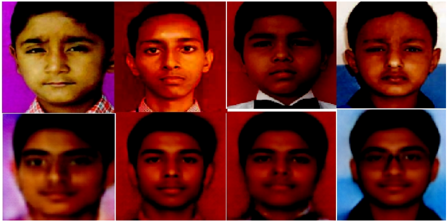


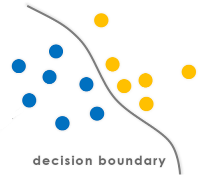
Figure 14: Results showing aging on child faces

Figure 15: Results showing de-ageing on child faces

Discriminative Models Vs Generative Models

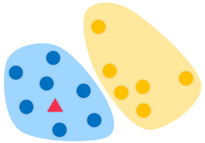
Discriminative Models

- Given X, Predict Y.
- Model learn P(Y|X) directly.
- Maximize conditional likelihood.
- Model cannot generate synthetic images.
- Examples : Logistic regression, SVM, NN.



Generative Models

- Given X, Predict P(X|Y).
- Model learn P(X,Y) directly.
- Maximize joint likelihood.
- Model can generate samples from P(X).
- Examples : RBM, GAN.

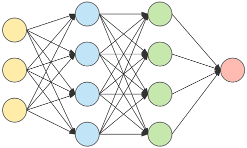


Definition

- Generative

- Adversarial

- Networks



Generative Adversarial Network Architecture

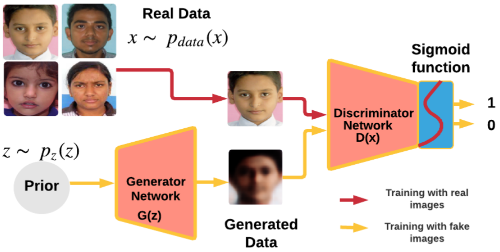


Figure 16: GAN Architecture

- Discriminator (D) distinguishing between real images and generated images and is responsible for classifying images as real(1) or fake (generated)(0).
- Generator (G) generating the images that fool the discriminator.

Loss Function

Binary Cross Entropy:

$$\mathcal{L}(\hat{y}, y) = y \log(\hat{y}) + (1 - y) \log(1 - \hat{y}) \tag{3}$$

Discriminator on real (D) :
y = original, $\hat{y}$ = generated image; The label for the data coming from $P_{data}(x)$ is $y = 1$ and $\hat{y} = D(x)$

$$\mathcal{L}(D(x), 1) = \log(D(x)) \tag{4}$$

Data coming from generator the label is $y = 0$ and $\hat{y} = D(G(z))$
Discriminator on fake (D):

$$\mathcal{L}(D(G(z)), 0) = \log(1 - D(G(z))) \tag{5}$$

Continue

Objective of the Discriminator is to correctly classify fake vs the real dataset. For this Equation 4 and 5 should be maximized.

$$\max_{D} \quad \log(D(x)) + \log(1 - D(G(z))) \tag{6}$$

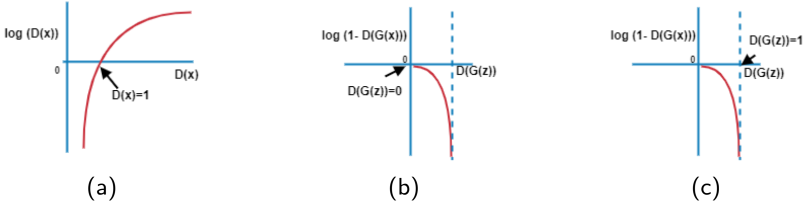


Figure 17: (a) and (b) Discriminator D Loss and (c) Generator G loss.

Objective of Generator to fool the discriminator. $D(G(z)) = 1$ then Generator fool the Discrimintor by producing the probability as 1.

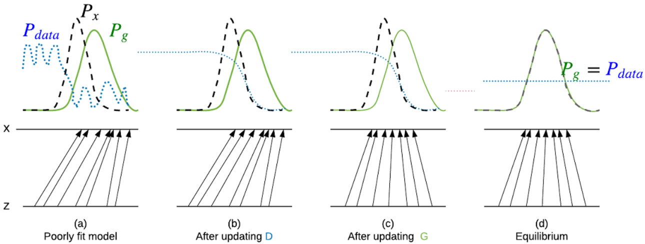
$$\min_{G} \quad \log(1 - D(G(z))) \tag{7}$$

Training GAN is a two-player minimax game problem

- Discriminator D tries to maximize it classification accuracy.
- Generator G tries to minimize the discriminator classification accuracy.

$$\min_G \max_D V(D, G) = \log[D(x)] + \log[1 - D(G(z))] \tag{8}$$

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim Pdata(x)} \log[D(x)] + \mathbb{E}_{z \sim Pz(z)} \log[1 - D(G(z))] \tag{9}$$



Training GAN

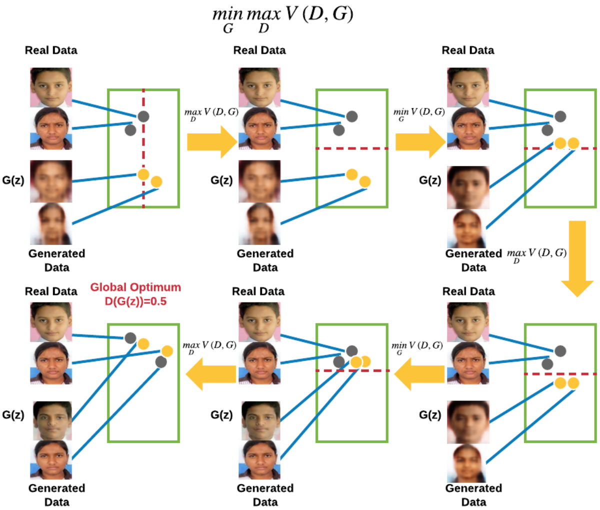
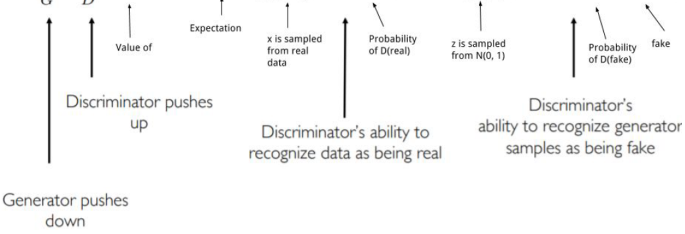


Figure 18: Training Steps in GAN

Value function

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$



1. Objective function derived from the cross entropy between real and generated distribution.
2. Generator tries to minimize objective function while the discriminator tries to maximize it.
3. $D(x)$ is the discriminator estimate of the probability that real data instance x is real.
4. E_x is the expected value over all the real data instances.
5. $G(z)$ is the generator's output when given noise z.
6. $D(G(z))$ is the discriminator's estimate of the probability that a fake instance is real.
7. E_z is the expected value over all random inputs to the generator (in effect, the expected value over all generated fake instances $G(z)$.

Algorithm

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
  - Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

  end for

  - Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
  - Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

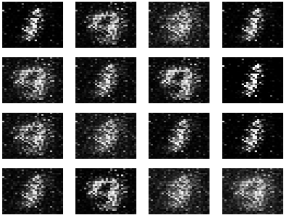
end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

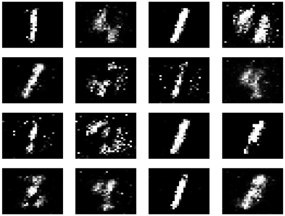
Figure 19: GAN Objective

Generative Adversarial Network implementation on colab (STTP_GAN.ipynb).

GAN Implementation Result: MNIST



(a) Epoch 5



(b) Epoch 20

Figure 20: DCGAN results on Celeb dataset

Challenges Faced by GANs

Training Instability

1. Non-convergence: GANs often fail to converge due to the adversarial nature of their training, as the generator and discriminator compete with each other.
2. Vanishing Gradients: When the discriminator becomes too good, the generator receives almost no gradient to improve itself.
3. Mode Collapse: The generator produces limited diversity, repeatedly generating similar outputs instead of a wide variety of data.
4. Sensitive Hyperparameters: Training requires careful tuning of hyperparameters like learning rates, batch sizes, and network architectures

DCGAN: MNIST

Generator

1. Hidden layers: Four 4x4 strided convolutional layers (1024, 512, 256, and 128 kernels, respectively) with ReLU.
2. Output layer: 4x4 strided convolutional layer (4096 nodes = 64x64 size image) with Tanh.
3. Batch normalization is used except for output layer.

Discriminator

1. Hidden layers: Four 4x4 convolutional layers (128, 256, 512, and 1024 kernels, respectively) with Leaky ReLU.
2. Output layer: 4x4 convolutional layer (1 node) with Sigmoid.
3. Batch normalization is used except for 1st hidden layer and output layer.

Deep Convolution Generative Adversarial Network

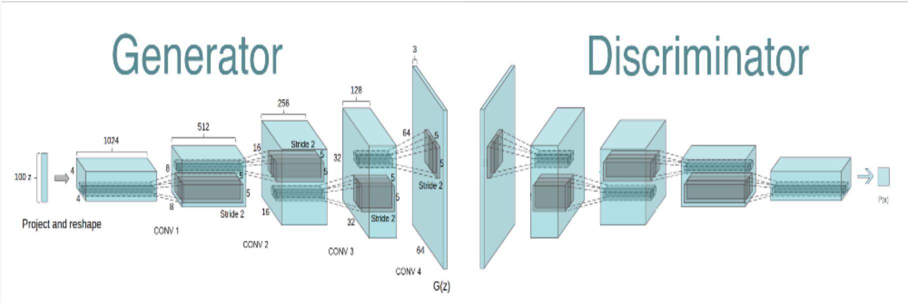
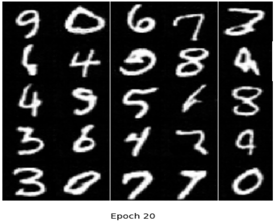


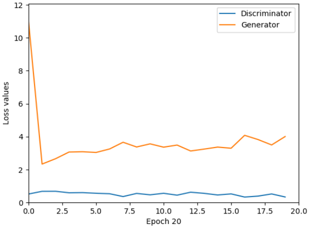
Figure 21: Deep Convolution Generative Adversarial Network [3]

1. Replace any pooling layers with strided convolutions (discriminator) and fractional-strided convolutions (generator).
2. Batchnorm in both the generator and the discriminator.
3. ReLU activation in generator for all layers except for the output, which uses Tanh.
4. LeakyReLU activation in the discriminator for all layers.

[3] Soumith et al. "Unsupervised representation learning with deep convolutional generative adversarial networks.



(a) MNIST Data



(b) Training Loss

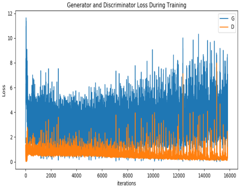
Figure 22: DCGAN results on MNIST dataset (STTP_DCGAN_MNIST.ipynb)

DCGAN Implementation Result: Celeb

(a) Celeb Data

(b) epoch-1

(c) epoch-15



(d) Training Loss

Figure 23: DCGAN results on Celeb dataset

Variants of GAN structure

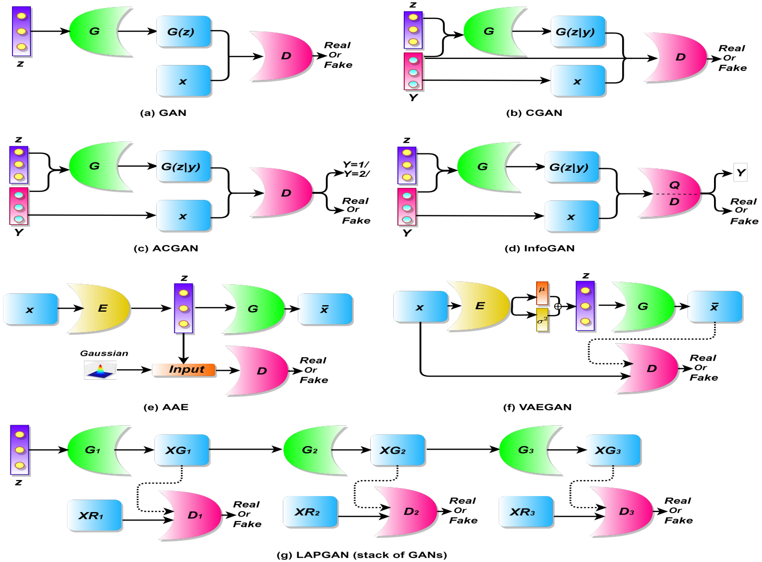


Figure 24: Variants of GAN structure

GAN Variants

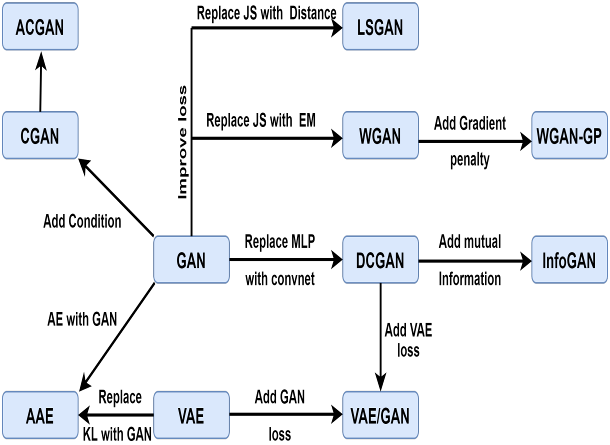


Figure 25: Variants of GAN structure

Conditional Generative Adversarial Network (cGAN)

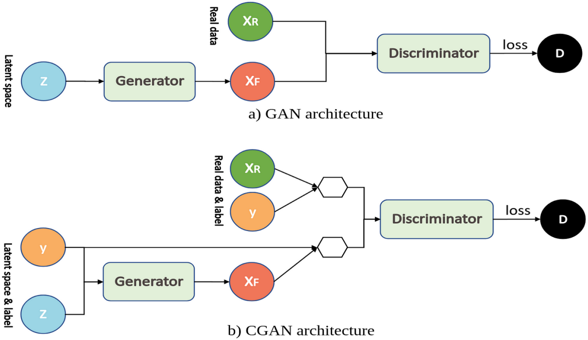


Figure 26: Conditional generative adversarial network

Conditional Generative Adversarial Network implementation on colab.

cGAN Result

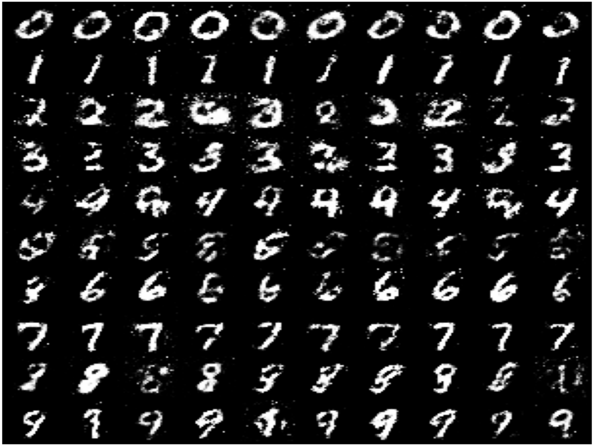


Figure 27: cGAN

Face Datasets

Table 4: Face datasets.

Dataset	Subjects	Images	Images Subject	Range	Avg.Age	Public
FGNET	82	1,002	6-18	0-69	16	Yes
Adience	2,958	26,580	NA	0-60	NA	Yes
Morph	13,000	55,134	2-53 (avg. 4.2)	16-77	42	Yes
CACD	2,000	163,446	22-139 (avg. 81.7)	16-62	31	Yes
IMDBWiki	20,284	523,051	avg. 25.79	0-100	38	Yes
UTKFace	NA	23,709	NA	0-116	33	Yes
AgeDB	568	16,488	avg. 29	1-101	50.3	Yes
AGFW-v2	36,325	27,688	-	10-64	–	Yes
ITWCC	754	7,990	3-37 (avg. 10.7)	0-32	13	Yes
CLF	919	3,682	2-6 (avg. 4.0)	2-18	8	No

Synthetic image generation models

Table 5: Synthetic image generation models

Study	Model	Model Architecture	Dataset	Findings
Yann LeCun et al. 1988 [3]	AutoEncoder	Encoder and Decoder	MNIST	Synthetic image
Welling et al. 2014 [4]	Variatioal AutoEncoder	Encoder and Decoder	MNIST,Frey Face datasets	Synthetic image
Goodfellow et al. 2014 [2]	Generative Adversarial Nets	Generator and Discriminiator	MNIST, TFD, CIFAR-10	Synthetic image
Mehdi Mirza et al. 2014 [5]	Conditional Generative Adversarial Nets	Generator and Discriminator	MNIST,MIR Flickr	Image tagging
Makhzani et al 2016 [6]	Adversarial Autoencoders	Encoder , Decoder, Discriminiator	MNIST,TFD,SVHN	Synthetic image
Soumith et al. 2016 [7]	Deep convolutional generative adversarial network	CNN base Generator, Discriminiator	LSUN, CACD, Image net	Synthetic image

Dataset Collection

Children Longitudinal Face [8]

1. The dataset contains $3,682$ face images of 919 children, in the age range of 2 to 18 years.
2. Each subject has an average of 4 images acquired over an average time lapse of 4 years (minimum time lapse of 2 years, maximum time lapse of 7 years).
3. Dataset is comprised of 604 (66%) boys and 315 (34%) girls.
4. The face images were captured with a resolution of 354×472 pixels.
5. Due to zoom variations, some faces occupy about only 70% of the image while some faces cover about 50% of the total image area.

Children Longitudinal Face

Figure 28: CLF dataset subject example, Age (in year) at image acquisition is given below the image.

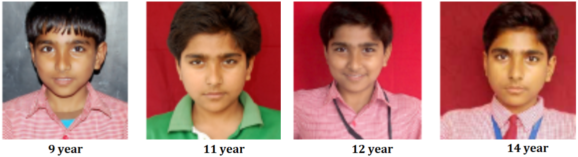


Figure 29: CLF dataset subject example, Age (in year) at image acquisition is given below the image.

Multi-task Cascaded Convolutional Networks Stages:

- (P-net): Produced candidate windows and bounding box vectors(Classification).
- (R-net): Refinement the windows(Regression of bounding box).
- (O-net): Finalized bounding box and facial landmarks position(Localization of facial landmarks point).

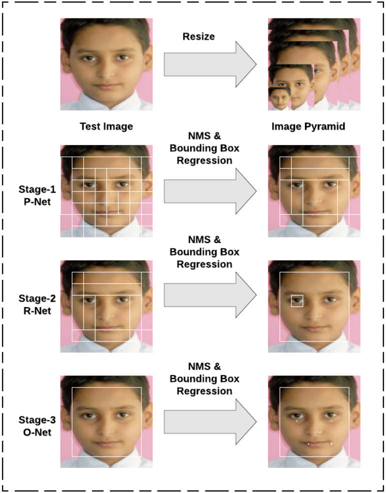


Figure 30: MTCCN Networks Stages

Preprocessing on Face dataset

Figure 31: First row shows the input image and second shows the preprocessed images. [4]

[4] Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks.

Praveen Kumar Chandaliya, Neeta Nain "Conditional Perceptual Adversarial Variational Autoencoder for Age Progression and Regression on Children Face" IEEE ICB 2019

Conditional Perceptual Adversarial Variational Autoencoder

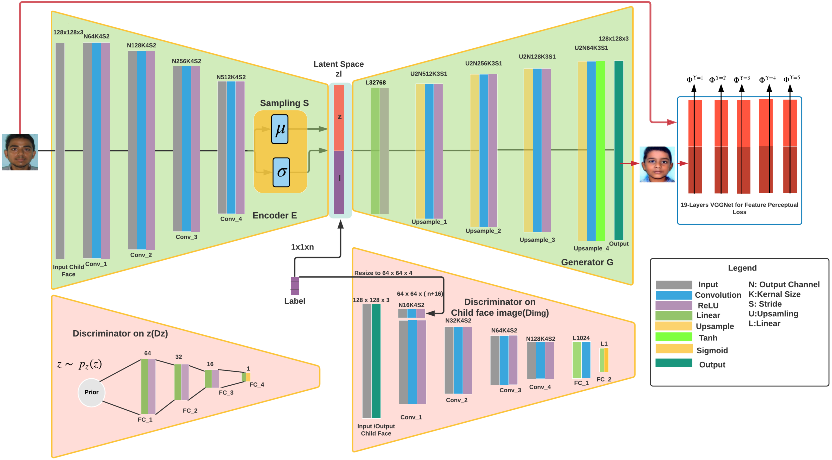


Figure 32: CPAVAE network for age progression and rejuvenation [5]

Conditional Perceptual Adversarial Variational Autoencoder

- CPAVAE model uses manifold learning approach and four component architecture.
- Encoder (E), Generator (G), Discriminator on z (D_z) and Discriminator on face image (D_{img}).
- Encoder take input images $x \in \mathbb{R}^{128 \times 128 \times 3}$ and applies stride convolution on them and produce output of encoder i.e. $E(x) = z$, with the help of encoder personal features of a child is preserved.
- Output of encoder is now conditioned with age and it is represented by $G(z, l) = x$. This conditioned age vector is sent to the generator as input.Generator uses deconvolution neural network and produces the required output.
- The D_z regularizes z to be uniform distributed, smoothing the age transformation.
- The D_{img} enforce Generator to generate realistic children faces for arbitrary z and l.

Objective function

A zero-sum non-cooperative objective function can be used to train E and D_z here D_z is weighted by 0.01.

$$\min_{E} \max_{D_z} \mathbb{E}_{z^* \sim p(z)}[\log D_z(z^*)] + \mathbb{E}_{x \sim P_{data}(x)}[\log(1 - D_z(S(E(x))))] \tag{10}$$

Discriminator D_{img} on child face image and G conditioned on l can be trained in adversarial fashion here D_{img} is weighted by 0.0001.

$$\begin{aligned} &\min_{G} \max_{D_{img}} \mathbb{E}_{x,l \sim P_{data}(x,l)}[log\ D_{img}(x, l)] \\ &+\mathbb{E}_{x,l \sim P_{data}(x,l)}[\log(1 - D_{img}(G(S(E(x)), l)))] \end{aligned} \tag{11}$$

Objective function

Perceptual Loss

$$\Phi^{\Upsilon} = \frac{1}{2 \times C^{\Upsilon} W^{\Upsilon} H^{\Upsilon}} \sum_{c=1}^{C^{\Upsilon}} \sum_{w=1}^{W^{\Upsilon}} \sum_{h=1}^{H^{\Upsilon}} (\Phi(x)^{\Upsilon}_{c,w,h} - \Phi(\bar{x})^{\Upsilon}_{c,w,h})^2 \qquad (12)$$

KL divergence loss $\mathcal{L}_{kl}$, used to regularize the distribution of latent vector z.

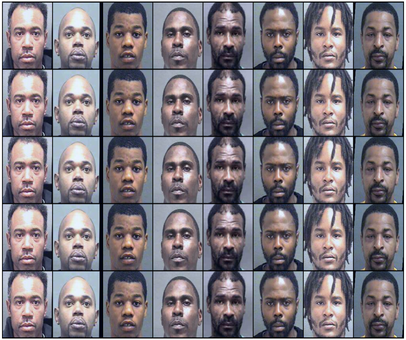
$$\mathcal{L}_{kl} = \sum_{i=1}^{N} q(z|x) \log \frac{q(z|x)}{p(z^*)} \qquad (13)$$

Final Objective function

Our resultant objective function becomes.

$$\begin{aligned}\min_{E,G} \max_{D_z, D_{img}} \mathbb{E}_{z^* \sim P(z)} \log\left[D_z\left(z^*\right)\right] \\ +\mathbb{E}_{x \sim P_{data}(x)}\left[\log\left(1 - D_z\left(S\left(E\left(x\right)\right)\right)\right)\right] \\ +\mathbb{E}_{x,l \sim P_{data}(x,l)}\left[log D_{img}\left(x, l\right)\right] \\ +\mathbb{E}_{x,l \sim P_{data}(x,l)}\left[\log\left(1 - D_{img}\left(G\left(S\left(E\left(x\right)\right), l\right)\right)\right)\right] \\ +\lambda_1 \mathcal{L}_{kl} + \lambda_2 \Phi\end{aligned} \tag{14}$$

CPAVAE Result on Morph



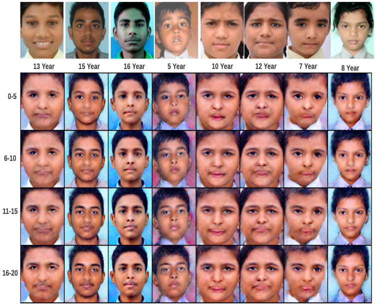
(a) Initial Input

(b) CPAVAE Result

Figure 33: CPAVAE Result on Morph.

VGG123 Vs VGG345

(a)



(b)

Figure 34: Shows face generated with (a) VGG123 and (b) VGG345 used as comparison layers by Perceptual loss.

Praveen Kumar Chandaliya, Aditya Sinha, Neeta Nain “ChildFace: Gender Aware Child Face Aging” IEEE BIOSIG 2020

ChildFace: Model Architecture

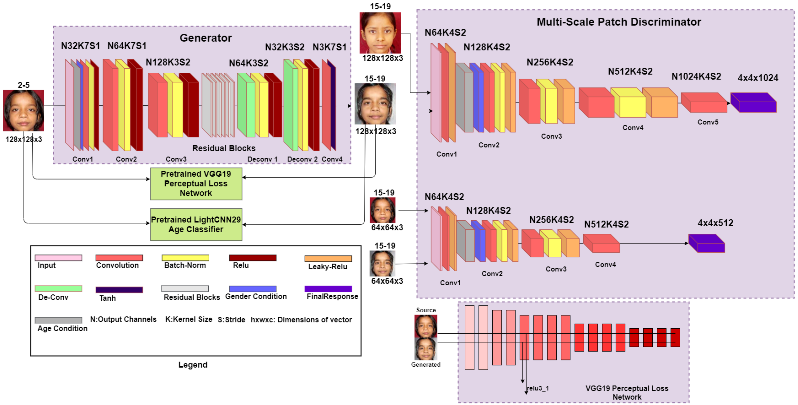


Figure 35: Detailed architecture of the proposed model [6]

[6] Praveen Kumar Chandaliya, Adtiya Shina, Neeta Nain, ChildFace,BIOSIG,2020,IEEE

Qualitative Evaluation: CLF Dataset

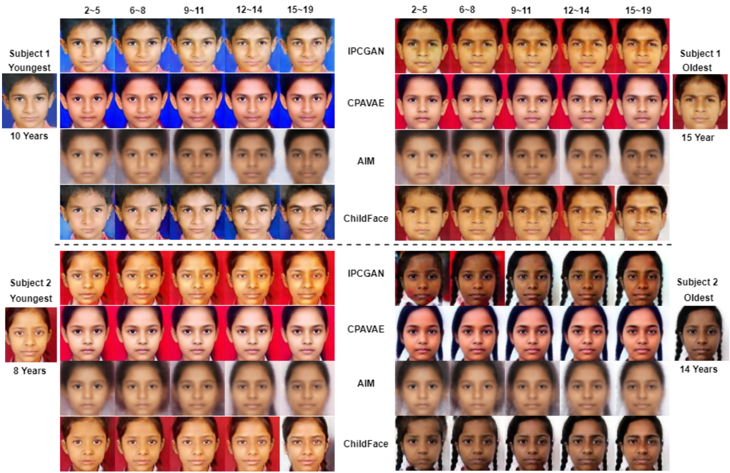


Figure 36: Comparison with prior work on aging: IPCGAN, CPAVAE, AIM and our ChildFace model, respectively.

Application of Generative Adversarial Network

Generate Photographs of Human Faces

- CycleGAN, StyleGAN

Text-to-Image Translation

- MirrorGAN

Super Resolution

- SRGAN, ESRGAN

Video Prediction

- RD-GAN

3D Object Generation

- 3D-GAN,MarrNet

Publication I

- Praveen Kumar Chandaliya, Preyas Garg and Neeta Nain, "Retrieval of Facial Images Re-rendered with Natural Aging Effect using Child Facial Image and Age," (SITIS), Las Palmas de Gran Canaria, Spain, 2018, pp. 457-463
- Praveen Kumar Chandaliya, Neeta Nain "Conditional Perceptual Adversarial Variational Autoencoder for Age Progression and Regression on Children Face" ICB 2019.**B Level**,
- Praveen Kumar Chadnaliya, Vardhman, Mayank Harjani, Neeta Nain, "SCDAE: Ethnicity and Gender Alteration on CLF and UTKFace Dataset" CVIP 2019
- Praveen Kumar Chandaliya, Aditya Sinha, Neeta Nain "ChildFace: Gender Aware Child Face Aging" BIOSIG 2020 **B Level**.

References I

Neeta Nain Praveen Kumar Chandaliya.
Childgan: Face aging and rejuvenation to find missing children.
In *Pattern Recognition*, volume 129, page 108761, 2022.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio.
Generative adversarial nets.
In *NIPS*, 2014.

P. Gallinari, Yann Lecun, S. Thiria, and F. Fogelman Soulie.
Memoires associatives distribuees: Une comparaison (distributed associative memories: A comparison).
In *Proceedings of COGNITIVA 87, Paris, La Villette, May 1987*. Cesta-Afcet, 1987.

Diederik P. Kingma and Max Welling.
Auto-encoding variational bayes.
CoRR, abs/1312.6114, 2013.

Mehdi Mirza and Simon Osindero.
Conditional generative adversarial nets.
CoRR, abs/1411.1784, 2014.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian J. Goodfellow.
Adversarial autoencoders.
CoRR, abs/1511.05644, 2015.

Alec Radford, Luke Metz, and Soumith Chintala.
Unsupervised representation learning with deep convolutional generative adversarial networks.
CoRR, abs/1511.06434, 2015.

References III

Debayan Deb, Neeta Nain, and Anil K. Jain.
Longitudinal study of child face recognition.
CoRR, abs/1711.03990, 2017.