INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 5, Issue 2)
Available online at: www.ijariit.com

Survey on generative adversarial networks

N. Yashwanth
yashwanthsa@gmail.com
Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, Andhra Pradesh

P. Navya
navyapatsamatla007@gmail.com
Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, Andhra Pradesh

Md. Rukhiya
rukhiya.mohammad@gmail.com
Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, Andhra Pradesh

K. S. V. Prasad
sivavaraprasadkakaraparthi@gmail.com
Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, Andhra Pradesh

Dr. K. S. Deepthi
selvanideepthi14@gmail.com
Anil Neerukonda Institute of Technology and Sciences, Visakhapatnam, Andhra Pradesh

ABSTRACT

GAN stands for Generative Adversarial Networks. GANs are the most interesting topics in Deep Learning. The concept of GAN is introduced by Ian Good Fellow and his colleagues at the University of Montreal. The main architecture of GAN contains two parts: one is a Generator and the other is Discriminator. The name Adversarial stands for conflict and here the conflict is present between Generator and Discriminator. And hence the name adversarial comes to this concept. In this paper, the author has investigated different ways GAN's are used in real time applications and what are the different types of GAN's present. GAN's are mainly important for generating new data from existing ones. As a machine learning model cannot work properly if the size of the dataset is small GAN's are here to help to increase the size by creating new fake things from original ones. GAN's are also used in creating images from the given words that are text-to-image conversion. GANs are also applied in image resolution, image translation and in many other scenarios. From this survey on GAN author aim to know what are the different applications of GAN that are present and their scope. The author has also aimed at knowing the different types of GAN's available at present. The author has also aimed at knowing the different applications of GAN and different proposed systems by various authors.

Keywords— *Generative Adversarial networks, Generator, Discriminator, Neural network, Deep Learning*

1. INTRODUCTION

The Artificial Intelligence engineer John McCarthy has quoted that AI is the science and engineering of making intelligent machines, especially intelligent computer programs. AI system has the capability to react and take decisions on its own based on the real-time events in nature. Machine Learning is a sub-part of AI and Deep Learning is a sub-part of ML. This Machine Learning field is rapidly growing in these days mainly in banking and health sectors. Deep Learning resembles the working of the human brain. As humans know the information by visualizing and process the information with neurons in the brain. Similarly, the concept of neural networks in deep learning works. The neural network has neurons which accept the information and interacts with other neurons in the other layer for processing. GAN's are an interesting topic in deep learning. GAN architecture is mainly made up of two parts Generator and Discriminator and both these are two separate neural networks. Always Generator tries to mimic and generate new things from the existing ones while the role of the discriminator is to identify whether that is fake or real. This is the actual conflict that is present between the Generator and Discriminator. Both of these resemble thief and police. These GAN's play an important role in generating new images from the existing ones as GANs reduce human time in generating new things. GANs are widely used in image synthesis, text-to-image conversion and many medical applications.

Both the networks in GAN need to be optimized and that should happen simultaneously. Both these networks try to reduce their loss and this is the most difficult part in creating GAN. Training the GAN is the most difficult part and many types of research are going on to make this process easy. Both the networks should find a point where both have minimum loss functions such that both do their job perfectly.

In the adversarial net framework, the generative model is made to observe the dataset thoroughly, a discriminative model that learns to distinguish whether a sample is from the original data as the generative model can be thought of as similar to a team of counterfeiters, trying to generate face currency and use without detection, while the discriminative model is similar to the police to detect the counterfeit currency [1].

The networks of generator and discriminator are typically implemented by multi-layer networks which are either convolutional or fully-connected layers.These models try to find all the statistical distributions of training data, thereby helping in creating or synthesizing new samples from the learned distributions.

Adversarial training is complicated as both generator and discriminator needs to get optimized simultaneously . And hence GAN's are very unstable. There will be many cases where the generator fails in understanding the distribution of the original data which the user wants. There are different techniques which are proposed to handle the instability of GAN. The different techniques are Laplacian Adversarial Networks and Deep Convolutional Generative Adversarial Networks and Adversarial Gradient Loss Predictors [2].

The Algorithm proposed by Ian GoodFellow in his paper is as follows:

For a number of training, iterations do
For k steps do
Sample minibatch of m noise samples from noise prior to pg(z).
Sample minibatch of m examples from data generating distributions pdata(x)
Update the discriminator by ascending its stochastic gradient.
End for
Sample minibatch of m noise samples from noise prior to pg (z)
Update the generator by descending its stochastic gradient
End for

Supervised machine learning means training data is going to have labels that are the output of every input is present and the model needs to predict the correct output for the new input. Coming to unsupervised machine learning the outputs of training data is not present here the model needs to predict that, to which class or cluster the given instance belongs to. Coming to semi-supervised learning some of the training data has labels and some others don't. All these types of problems can be solved with the help of neural networks. And the two important components of GAN are made up of these neural networks only. An adversarial model is used to learn conditional Probability function (P(y|x) i.e. given set of values x (observed values), the author can predict y (unobserved values) [2].

GAN's belong to implicit density models. In general, there are two different types of generative models Implicit and Explicit density models. In explicit density models, author need to express a model for generated samples explicitly whereas in the other case this needs to learn the model that can take samples without explicitly defining and hence GAN's belong to Implicit Density Models[2].

2. TYPES OF GAN'S

2.1 Fully connected GAN's

These are the first GAN's which are introduced and these uses fully connected neural networks. This model has many disadvantages and applicable only to simple datasets such as MNIST , CIFAR-10 etc.

2.2 DCGAN

After GAN's are implemented DCGAN's are implemented in order to improve the efficiency of GAN's.A DCGAN uses convolutional and convolutional-transpose layers in discriminator and generator.These type of GAN specifically use Convolutional layers in Discriminator and De-Convolutional layers in Generator. In order to understand the statistical distributions of the original samples, these convolutional layers are added.This CNN algorithm mainly used for finding the important features of a given image which are used for finding the patterns such as edge detection. DCGAN's are highly stable in generating new images than normal GAN's.In the paper [4], the author has proposed to prevent a GAN from its instability author must include:

(1) Replace any pooling layers with strided convolutions for discriminator and fractional-strided convolutions for generator [4].
(2) By removing the fully connected hidden layers [4].
(3) Use Batch Normalization for both generator and discriminator [4].
(4) All layers in the generator except output layer should use ReLU activation function and for the output, layer use Tanh [4].
(5) Use the Leaky ReLU activation function in the discriminator for all layers [4].

2.3 Improved DCGAN's

Even after the extension of GAN's with DCGAN's, there are still some problems with DCGAN's and hence different measures are proposed to overcome them.

(1) In general, the real data is labelled as 1 and fake data to 0 or vice versa but the improvised technique says that real data should be labelled as 0 and fake data to be 0.9 which is going to help in training called as one-sided label smoothing [5].
(2) Use virtual Batch Normalization instead of normal Batch Normalization [5].

2.4 Conditional GAN-CGAN

This is also an extension of generative adversarial networks by conditional setting. The main task of the generator is to fool the discriminator by making that to believe that the created samples by the generated to be original. The additional capability for each network to condition on some arbitrary data which describes the image being generated or discriminated [6] conditional GAN extends the formulation by providing the generator with extra labels. In Conditional GAN extend the formulation by providing the generator takes the form of an encoder-decoder network, where the encoder projects the label into a low-dimensional latent subspace a the decoder performs the opposite mapping [7].

The two networks Generator G(z) and Discriminator D(x) in normal GAN but in Conditional GAN Generator is G(z,y) and Discriminator is D(x,y) where y is the condition. This y is nothing but a vector which is a condition to both the networks.

2.5 Info GAN's
These are again the extension of GAN where an information-theoretic is present thereby GAN is able to learn disentangled representations in a completely unsupervised manner [8].

Instead of the latent variables being known a priori from a dataset, make parts of latent space randomly drawn from different distributions . Examples- Bernoulli, Normal, multiclass, etc. Make the discriminator reconstruct these arbitrary elements of latent space that are passed into the generator. Mutual Information between Generated samples and a small subset of latent variables C.The Author can fill the interesting features and aspects of the representation into C by forcing the high information content.

2.6 LAPGAN
LAPGAN stands for Laplacian Generative Adversarial Networks. LAPGAN has composed of a torrent convolutional GANs with the framework of a laplacian pyramid with N levels. At the stage of the level, N, a GAN is trained which maps a noise vector to an image with the coarsest resolution. At each level of the pyramid except the coarsestone, a separate cGAN is trained, which takes the output image in the coarser level (i.e., level N+ 1) as a conditional variable to generate the image at this level. Due to such a coarse-to-fine manner, LAPGANs are able to produce images with higher resolutions.

2.7 Vari GAN
Vari GAN stands for Variational GAN which was proposed to generate multi-view human images from a single view. This GAN follows a coarse-to-fine manner. Vari GAN has been composed of three networks: Coarse Image Generator, a fine image generator and a conditional discriminator. The coarse image generator GC uses a conditional VAE architecture where VAE stands for Variational Auto Encoder. Provided an input image i and a target view v, that was separately trained to generate a low resolution with the target view i-v (low resolution). The fine image generator GF is composed of dual-path U-Net architecture. The U-Net is named after its symmetric shape. This maps i-v(low resolution) to a high-resolution image conditioned on the input image. Discriminator D examines the high-resolution image conditioned on the input image. GF and Discriminator are trained with an objective function consisting of an adversarial loss and a content loss measuring the L1 difference between (i-v)high resolution and ground truth.

2.8 vGAN
Vondrick proposed a generative adversarial network for video (vGAN). Assumed the whole video is combined by a static background image and a moving foreground video. Hence, the generator has two-streams. The input to both of the streams is a noise vector. The background stream puts its effort to generate the background image with 2D convolutional layers, and the foreground stream tries to generate the 3D foreground video cube and the corresponding 3D foreground cover, with spatial-temporal 3D convolutional layers. The discriminator takes the total generated video as input and tries to distinguish from real videos. Since vGAN treats videos as 3D cubes. This requires large memory space. This can also generate tiny videos of about one-second duration.

2.9 TGAN
TGAN stands for temporal generative adversarial network. This GAN was proposed by Saito. TGAN is composed of a temporal generator, an image generator and a discriminator. The temporal generator produces a sequence of latent frame vector [V11, V12, V13, ... , V1S] from a random variable V0, where S is the number of video frames. The image generator takes V0 and a frame vector Z1t ($0 < t < S+1$) as input and produces the t-th video frame. Here also, the discriminator takes the total video as input and tries to distinguish from real ones. For stable training, TGAN follows WGAN, but further apply singular value clipping instead of weight clipping to the discriminator.

2.10 MGAN
MGANs. Li proposed a real-time texture synthesis method. The author first introduced Markovian deconvolutional adversarial networks (MDANs). Given a content image and a texture image. MDANs synthesize a target image e.g., a face image textured by leaves). Feature maps of an image are defined as feature maps extracted from a pre-trained VGG19 by feeding the image into it, and neural patches of an image are defined as patch samples on the feature maps. The discriminator is trained to differentiate neural patches from real and fake images. The objective function includes a texture loss as well as a feature loss. The texture loss is computed from the classification scores of neural patches of the target image from the discriminator. The feature loss considers the distance between the feature maps of the target image and content image. The target image is initialized with random noise and is continuously updated through back propagation by minimizing the objective function. The author further introduced Markovian generative adversarial networks (MGANs), which take feature maps of a content image as input to generate a texture image. MGANs are trained using content and target image pairs synthesized by MDANs. The objective function of MGANs is defined similarly to MDANs. MGANs have gained the ability to achieve real-time performance for neural texture synthesis, which is about 500 times faster than previous methods.

2.11 Other GANs
Berthelot proposed Boundary Equilibrium Generative Adversarial Networks (BEGAN), trying to maintain an equilibrium which can be adjusted for the trade-off between diversity and quality.

Zhao proposed an Energy-Based Generative Adversarial Network (EBGAN) which views the discriminator as an energy function instead of a probability function. The author shows that EBGANs are more stable in training.

To overcome the vanishing gradient problem, Mao proposed Least Squares Generative Adversarial Networks (LSGAN), which replace the log function by least square function in the adversarial loss.

Arjovsky proposed Wasserstein generative adversarial networks (WGAN). The author first theoretically show that the Earth-Mover (EM) distance produces better gradient behaviors in distribution learning compared to other distance metrics.

According to that, the author made several changes to regular GANs: (1) removing the sigmoid layer and adding weight clipping in the discriminator; (2) removing the log function in the adversarial loss. The author demonstrates that WGAN's generate images with the comparable quality compared to well-designed DCGANs.

SGAN and PSGAN: Regular GANs map a random vector to an image. Instead, Jetchev and Bergmann proposed Spatial GANs (SGAN), which extend to map a spatial tensor to an image. The network architecture follows DCGANs.The architectural properties of SGAN make suitable for the task of texture synthesis. Bergmann further extends SGAN to Periodic Spatial GAN (PSGAN). In PSGAN, the input spatial tensor contains three parts: a local independent part, a spatially global part, and a periodic part. PSGAN is able to synthesize diverse, periodic and high-resolution textures.

SRGAN: Ledig proposed super-resolution generative adversarial network (SRGAN), which takes a low-resolution image as input, and generates an upsampled image with 4 * resolution. The network architecture follows the guidelines of DCGAN, and the generator uses the very deep convolutional network with residual blocks. The objective function includes an adversarial loss and also a feature loss. The feature loss is computed as the distance between the feature maps of the generated upsampled image and the ground truth image, where the feature maps are extracted from a pre-trained VGG19 network. Experiments show that SRGAN has better performance at the state-of-art approaches on public datasets.

FCGAN: Based on Boundary Equilibrium Generative Adversarial Networks (BEGAN), Huang proposed Face Conditional Generative Adversarial Network (FCGAN), which focuses on facial image super-resolution. Within the network architecture, both the generator and discriminator uses an encoder-decoder along with skip connections. For training, the objective function includes a loss i.e., content loss, which is computed by the L1 pixel-wise difference between the generated upsampled image and the ground truth. FCGAN generates good results with 4 * scaling factor.

TextureGAN: Xian proposed TextureGAN, which converts sketch images to a realistic image with the additional control of object textures. The generator takes a sketch image, a color image, and also a texture image xt as input to generate a new image x. The network structure follows Scribbler. The objective function consists of a content loss, a feature loss, an adversarial loss as well as a texture loss. The content loss, feature loss and also adversarial loss are defined similarly to Scribbler. Following the CNN based texture synthesis method, the texture loss is computed as the distance between the Gram-matrix representation of patches in x and xt, enforcing the texture appearance of x close to a target image. Segmentation masks are also introduced to make computing texture loss and content loss only in the foreground region.

3. APPLICATIONS

3.1 Image Super Resolution

The main theme of this is to improve the quality of the low sampled image that is upsampling the given image. There are many other techniques which are proposed but those could not perform well in with very low sampled images. This super-resolution GAN uses deep learning concept to give higher resolution images. In the process of training, a high-resolution image is always converted into low-resolution image by downsampling. The Generator of the GAN is responsible for converting the low-resolution image to high-resolution image and discriminator is responsible for classifying the generated images.

Ledig et al. [8] have proposed to use SRGAN for this type of problems. The author has implemented a system which gives upsampled image with 4 resolution by taking a low-resolution image as input. The network architecture of SRGAN implemented by Ledig et al. [8] follows DCGAN architecture. The architecture of the generator contains both convolutional and residual networks.

3.2 Image Inpainting

The main concept of this application is to fill the gaps of an image. Many deep learning techniques have come to solve this problem and the major challenge is to fill the large gaps of an image to make a perfect one. There are convolutional networks for image inpainting but these are not good at filling the gaps with correct features and hence generative models are used for finding the correct features which are to be filled with and these features are known through the training process.

Pathak et al. [9] have proposed a new method for image inpainting called context encoders. These context encoders are based on convolutional networks trained mainly to generate images at an arbitrary. So these networks need to understand both full images and images with holes to identify the features with which need to replace with.

The network proposed by Pathak et al.[9] is based on encoder-decoder architecture.That system is capable of taking images with input size 128x128 with holes. The output of that proposed system is either the hole of the image or the entire image. The hole of the image size will be 64x64 and the full image is 128x128.

3.3 Face Aging

The main aim of this is to generate the human image which after some age. If the presence of the human is 20 years the GAN is ready to generate an image of that person at 40 years. Face ageing methods transform a facial image to another age, while still keeping identity.

Most of the GAN's used for Face ageing involves Conditional GAN's. The main aim is to generate an image with a target label age from a given initial face image. Conditional GAN's do not have an explicit mechanism of inverse mapping, therefore, that should contain an input image x with label y to a latent vector any z. Moez Baccouche et al [9]. Has proposed chan's for this Face Aging the main two steps:

(1) At first, the input face along with the person's age is given now the task of finding an optimal latent vector z should be done which allows generating a reconstructed face.
(2) The target age is given and now the image is generated xtarget=G(z∗,y) by simply changing the age in the generator. Zhang et al. [10] have proposed a new network for face ageing called Conditional Adversarial AutoEncoder (CAAE). This system consists of an encoder, a generator and two discriminators. The encoder is mainly responsible for getting the Z vector which contains personal features. The output vector Z together with a conditional vector C indicating a new age is fed into the generator G to generate a new face image.

3.4 Video generation

Video prediction stands for forecasting the future frame with respect to the existing frames. Different approaches were proposed for this purpose.

Since GANS have proved their importance and efficiency in image synthesis, many of the researchers have thought that GANs can also be applied for video synthesis. But when compared to image synthesis, video generation will be a bit tough since that has an extra-temporal dimension and also memory matters.

VGAN which was already discussed above is a video generation gan. Assuming that the whole video as a combination of a fixed background and a mobile foreground. Hence this gan consists of two generators. The input which will be a noise vector to both of these generators. These two produce an output which is sent as a whole to the discriminator to distinguish between the real and the generated. This gan generates a video of only a second of duration.

The TGAN which was also introduced before is a kind of video generation, Gan. In this, generator named temporal produces a sequence of latent frame vectors from a random variable. The image generator takes this random variable and frame vectors and produces the video frame. This was passes to the discriminator.

MoCoGAN[12], a video generation GAN, splits the video into content and motion The content says objects in the video and the motion describes their movement. This GAN produces a video by generating a sequence of random vectors to a continuous video frame. Every random vector consists of a content part and a motion part. In order to learn motion and content decomposition in an unsupervised manner, a new kind of learning scheme was introduced which is a novel adversarial learning scheme utilizing both image and video discriminators. Extreme experimental outcomes on various challenging datasets with qualitative and quantitative comparison to the state of art approaches, verify the effectiveness of the proposed one. Found that MoCoGAN allows one to produce videos of the same content but different motion and also videos with different content and same motion.

3.5 Text to image GAN

The phenomenon of producing an image for a given text description corresponds to Text-to-image Gan. Of course, this will be definitely going o tough call but recently this was able to produce images depicting simple scenes with the help of gans.

GAN-INT-CLS (GAN-Interpolation-Conditional latent space): This was proposed by Reed. This GAN is a text to image synthesis GAN. This GAN takes an input text which is encoded into a text embedding vector. Based on the condition on the text embedding vector, the generator maps a noise vector z to a synthesized image. The discriminator was used to check whether the input image is real or fake, and that matches the texture description. The network used in this gan is similar to DCGAN architecture.

GAWWN (Generative Adversarial What-Where Network): This was also proposed by Reed. The networks (both the generator and the discriminator) are conditioned on the bounding box and the text embedding vector which represents text description. The networks have two ways, a global pathway that operates on the full image, and a local pathway that operates on the region inside the bounding box. For keypoint constraints, a keypoint-conditional GAN is also introduced. The key point constraints are done by using binary mask maps.

3.6 Sketch to image

Sketches are the better what for users to draw what he wants, but lack detail, colour etc. Hence automatically mapping the input sketches to the user desired images is an attractive problem for researchers. Sketch to photo provides a better way to synthesize images sketch and text labels are necessary for their work Recently, GAN based methods are able to produce images from sketches without text labels showing better flexibility.

Scribbler: Sangkloy proposed GAN-based synthesis method named Scribbler that converts sketch images with colour strokes to realistic images. The generator employs an encoder-decoder architecture with residual blocks and generates a new image with the same resolution as the input sketch image. The objective function consists of a content loss, a feature loss, adversarial loss and a TV loss. The content loss measures L2 pixel-wise difference between the generated image xb and ground truth xG. Similar to MDANs, the feature loss is defined as the feature distance between xb and xG, where the features are extracted from a pre-trained VGG19 network. The TV loss is included in order to improve the smoothness of the generated images. Scribbler is able to generate realistic, diverse, and controllable images diverse, and controllable images.

TextureGAN: Xian proposed TextureGAN which converts sketch images to realistic images with the additional control of object textures. The generator takes a sketch image, a colour image, and a texture image xt as input to generate a new image xb. The network structure follows Scribbler. The objective function consists of a content loss, a feature loss, adversarial losses well as a texture loss. The content loss, feature loss and also adversarial loss are defined similarly to Scribbler. Following the CNN based texture synthesis method, the texture loss is computed as the distance between the Gram-matrix representation of patches in xb and xt, enforcing the texture appearance of xb close to xs. Segmentation masks are also introduced to make computing texture loss and content loss only in the foreground region.

4. CONCLUSION AND FUTURE SCOPE

There are still huge advances in image synthesis and other applications using GANs in the most recent time period are made. By moving through a large amount of datasets, GANs gained the ability to generate more meaningful, more semantically consistent results than previous traditional methods. GANs can produce texture details and realistic content, which will be an advantage to many applications, such as texture synthesis, super-resolution, image inpainting, etc.

However, GANs are still facing many challenges. First, difficult was to generate high-resolution images. At present, most GAN-based applications are limited to handle images with a resolution not larger than 256 ×256. When used for high-resolution images, blurry artifacts usually occur. Although some approaches use rough-to-fine iterative approaches to generate high-resolution images, those are not end-to-end and are usually slow. Recently, Chen et al.[94] introduced cascaded refinement networks for photographic image synthesis at 2-megapixel resolution, which gives us a novel perspective for high-resolution image generation. Secondly, the resolutions of input and output images are usually ensured to be fixed. In comparison, traditional image synthesis approaches are more flexible and could be adapted to arbitrary resolution. Recently proposed PixelRNN draws images pixel to pixel and allows arbitrary resolution, which gives a good insight. Also as a common issue in deep learning, ground truth data (for training) are crucial but hard to get. This is more important in GAN-based image synthesis and editing applications because usually, that was not easy to find ground truth of synthesized or edited images (or simply do not exist). CycleGAN and AIGN proposed to use unpaired data for training, which might be a feasible solution for similar problems but this needs more attention and exploration. Finally, although GANs have been applied to video generation and synthesis of 3D models, the results are far from perfect. That was still difficult to extract temporal information from videos or decrease memory costs.

5. REFERENCES

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio Département d'informatique et de recherche op erationnellen Universit ́e de Montr ́ealMontr ́eal, QC H3C 3J7
[2] Mahima Vuppuluri, Abhishek Dash, Survey on Generative Adversarial Networks
[3] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio‡D ́epartement d'informatique et de recherche op erationnelleb Universit ́e de Montr ́ealMontr ́eal, QC H3C 3J7
[4] Alec Radford & Luke Metz et al. Unsupervised representation learning with deep convolutional generative adversarial networks.
[5] Tim Salimans Ian Goodfellow Vicki Cheung et al.Improved Techniques for Training GANs
[6] Jon Gauthier, Symbolic Systems Program, Natural Language Processing Group Stanford University - Conditional generative adversarial nets for convolutional face generation.
[7] Grigorios G. Chrysos, Jean Kossaifi, Stefanos Zafeiriou - Robust Conditional Generative Adversarial Networks.
[8] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi-Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network.
[9] Grigory Antipov, Moez Baccouche, Jean-Luc Dugelay - Face aging with conditional generative adversarial networks.
[10] Xian Wu, Kun Xu, Peter Hall - A Survey of Image Synthesis and Editing with Generative Adversarial Networks.
[11] Sergey Tulyakov, Ming-Yu Liu, Xiaodong Yang, Jan Kautz - MoCoGAN: Decomposing Motion and Content for Video Generation.

