# Improving Generalization for Geometric Variations in Images for Efficient Deep Learning

Shivam Grover[1], Kshitij Sidana[1], Vanita Jain[1], Rachna Jain[2] and Anand Nayyar[3*]

[1*]Department of Information Technology, Bharati Vidyapeeth's College of Engineering, A-4 Block, Baba Ramdev Marg, Shiva Enclave, Paschim Vihar, New Delhi, 110063, Delhi, India.
[2*]Department of Information Technology, Bhagwan Parshuram Institute of Technology, PSP-4, Dr KN Katju Marg, Sector 17, Rohini, Delhi, 110089, Delhi, India.
[3*]School of Computer Science, Faculty of Information Technology, Duy Tan University, Da Nang, 550000 Vietnam.

*Corresponding author(s). E-mail(s): anandnayyar@duytan.edu.vn;
Contributing authors: shivumgrover@gmail.com; kshitijsidana@gmail.com; vanita.jain@bharatividyapeeth.edu; rachnajain@bpitindia.com;

**Abstract**

Deep Learning models for tasks such as image classification have a hard time adapting to the unseen geometric variations (such as scale, perspective, pose, etc) that the real-world offers in its data points. These variations can cause even a state-of-the-art model to perform poorly when used in real-world applications. This paper aims to solve this issue by presenting a general two-step method to cumulatively improve the generalization capabilities of the model when subjected to unseen geometric variations. The first step involves changing the model to be more dynamic while learning. This involves adding a deformable convolutional layer in our model. The second step of our method involves bringing realistic geometric diversity into our dataset

that the models are trained on. A combination of the traditional augmentation algorithms widely used (for example scaling, translation, rotation, etc.) and also Generative Adversarial Networks (GAN) for more realistic augmentation are used. The results show the efficacy of our methods on both non-generative and generative models. This method improves upon the existing traditional convolutional models adding a trainable offset parameter that helps the model adapt to geometric deformations in real world data. We show that our method improves the performance by 36% in the state-of-the-art model for conditional image translation. It is also observed that using the proposed method with a simple Convolutional Neural Network (CNN) classifier in the task of classification, the baseline accuracy is improved by 7%.

**Keywords:** Data augmentation, Deformable ConvNet, transformation, data augmentation, Image-to-image translation, Deep Learning

# 1 Introduction

Deep Learning practitioners often find themselves in situations where, even though they have trained their model on a large number of data points, the model consistently fails on real-world data of the same domain that wasn't part of their dataset. This is primarily because a real-world dataset practically offers an indefinite number of variations that weren't present in the training dataset. The number of variations that can be present in unseen data is huge and each type of variation requires a study of its own. In order to come up with a solution, one first needs to understand the root cause of the problem. An object can appear to have different visual geometric properties when the viewing conditions are altered. For example, a bicycle when seen from the top would geometrically appear to have a different shape from a bicycle viewed from the side (Fig. 1). Also a bicycle view from very close would not just be larger but also have a warped view due to perspective. Moreover, bicycles inherently come in various shapes and colors, and even if a really big dataset is curated, it's impossible to be able to accommodate all possible variations inside of it. When deep learning models have to work on several objects and perform more complex tasks, the impact of such unseen variations increases many folds and causes the model to fail on real-world data. A model when tested on a previously unseen variation of an object it has been trained to detect might fail since traditional deep learning models do not have an inherent sense of geometry [1].

General practices to avoid this involve having a larger and more diverse dataset preferably collected from the real-world, as shown in [2], [3], [4], [5], [6], or using heavy and realistic simulations that try to capture the variations one may see in real-world scenarios [7]. This effectively increases the quantity of the data but in most of the cases, the quality is reduced since these augmentations are done through some fixed set of operations that do not match

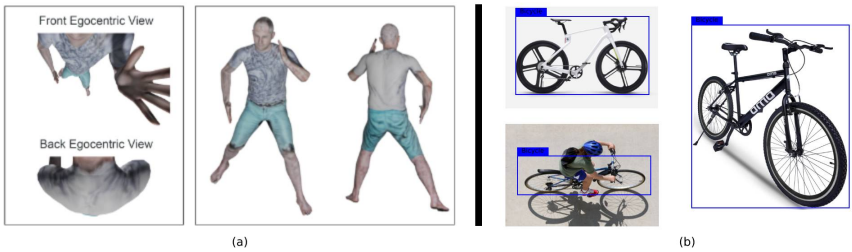


**Fig. 1** A visual overview of the geometric variations present in real-world data. (a) shows the egocentric views on the left and the third-person view on the right. Even though it is the same person in all images, the geometric variation is huge. The traditional generative model for generating third-person views using egocentric views as input would not be able to understand the relation between them very well. (b) shows different views of a bicycle. A classification model needs to understand how a bicycle looks regardless of its orientation, size, and position of the image source.

the variations the real-world data offers. Another common practice along with curating a large dataset is building larger models with more parameters [8]. A well known example would be the sequence of progressively complex structures from the Alexnet [9] to the subsequent state-of-the-arts VGG-16 [10], Resnet [11], Inception-V3 [12], and Densenet [13]. These two methods prove to be extremely difficult and expensive since collecting a large real-world dataset is very hard and inconvenient as well as training a larger model on such a dataset requires high computational power and takes more time. Our work presents a novel method for deep learning models to account for geometric variations in real-world data, that is especially helpful with limited datasets and smaller models that use less computational power.

Another task where this problem is more apparent is view translation [14], [15], [16], [17]. In this a generator is asked to take an image of an object and generate an image of the same object and scene but as seen from a different point of view. This is a task that belongs to the category of image-to-image translation and previously generative networks such as Generative Adversarial Networks (GANs) [18] and Variational Auto Encoders [19] were used for this. Traditional image-to-image translation networks use conditional GANs [20] and they strictly work only when the input and output images geometrically align. That roughly means if you extracted the edges of the input and output images and put one on top of the other they should overlap. This is because the encoder and decoder structure of a conditional GAN uses Convolutional Neural Network (CNN) layers[21] which inherently do not understand the geometric transformation that exists from the input to the output domain very well. These encoder-decoder layers are sometimes also joined with skip connections [22, 23] which force them further to assume a geometric alignment. For tasks where the input image does not align with the output image (Fig. 3), these networks fail.

A two-step approach towards solving the problem of geometric generalization is used. In the first step, the model is improved to be more dynamic to

geometric variations in the input without increasing its size. In the second step, the dataset is improved through the use of lightweight augmentation methods coupled with a Generative Adversarial Network (GANs) [18] for more realistic and high-quality augmented data. This allows our approach to combine the best of both worlds without compromising on performance and efficiency.

For improving the data, images that successfully capture a large portion of the possible geometric variations need to be added. A common solution vastly adopted is to use data augmentation [24]. This involves taking the existing data and making copies of it by incorporating various variations such as changing the scale and rotating the image. Apart from the previously mentioned simple transformation, in order to simulate the perspective change, warping is performed on the image within the size constraints. The aim is to simulate the possible variations and let our model train on them once. This effectively increases the quantity of the data but the quality is generally reduced since these augmentations are done through some fixed set of operations that do not match the variations the real-world data offers.

In order to increase the quality of the final dataset, GANs[18] are used since more recent advances in data augmentation use them. GANs are widely known for generating realistic unseen data samples which match the distribution the model was trained on [25, 26]. They have the capabilities of generating multiple unseen views of the same object in various environmental conditions based on the data the model was fed. Our method consists of integrating GANs with the traditional data augmentation for geometrical variations of our data.

While keeping the model size small, the aim is to change the model in a way that it is able to generalize among geometric variations better. Algorithms that make object detection invariant to geometrical changes have been studied for a long time, but they usually assume that the geometric variations are known and constrained, which is not the case for real-world samples and such hand-made algorithms do not work in unconstrained environments. Since our work is built to aid deep learning models for image operations, our method needs primarily to be usable with any CNN-based models. Hence, a more recent approach presented by [27] is adopted. The use of a deformable convolutional layer in our CNN model allows it to be robust towards geometric changes.

Traditional CNNs are not capable of modeling large and previously unseen geometric transformations. They largely use a fixed structure that doesn't work well for tasks that involve deformation of views. At the very basic, the locations at which a convolution unit does the task of sampling the input feature map are constant and fixed. [27] solves this issue by introducing 2D offsets to this grid for sampling locations and a similar offset for the deformable ROI (region of interest) pooling as well. The offsets that are added are learned dynamically for each task instead of hand tailoring, which allows the model to understand hidden geometrical properties within the dataset. By using a deformable convolutional layer for the first encoder layer in our generator network, its ability to generalize well to geometric variations is directly improved.

The main objective of our work is to improve the geometric generalization of an image based CNN model. We can further divide this objective in two parts, diversifying the dataset and making the model more dynamic, as can be viewed in Fig. 4. For diversifying the dataset, first our initial dataset is fed into a GAN that was built in order to generate more realistic samples and then algorithmic geometric data augmentation is performed on the enlarged dataset. For improving the model, the layers of the model are manipulated thereby making them more dynamic to the various geometric variations.

The final objectives of this paper come out to be:

1. To add a trainable offset parameter in the traditional convolutional neural network in order to make the model dynamic and allow it to learn in a way non-limiting with respect to geometric deformations
2. To implement adaptive ROI pooling to allow the convolutional model to have adaptive part localization in data points having varying sizes and shapes .
3. To diversify the dataset to match real world data points using Generative Adversarial Networks and algorithmic geometric data augmentation.
4. To keep this whole system lightweight to allow for efficient training and inferring.

The rest of the paper is organized as follows. Section 2 gives an overview of related work with examples. Section 3 discusses the datasets and methods we use. Section 4 gives a detailed explanation of the Proposed Methodology. Section 5 discusses several experiments that we performed, their results, and provides a walk-through of the demonstration. Section 6 concludes the paper with future scope.

# 2 Related Work

## 2.1 Geometrically Dynamic Image Models

We first explore various limitations in current deep learning models relevant to our proposal and the different approaches taken to solve them. For visual identification tasks such as classification of images [9], semantic segmentation [28], and object detection [29], Convolutional Neural Networks (CNNs) [21] have shown great promise. Traditional GANs [22, 25, 26] for image generation also use CNNs for their generator and discriminator. However, they are not inherently appropriate for understanding scene geometry unless explicitly trained to do so. In order to train them to understand geometric transformations and variations, one either needs to perform large amounts of data augmentation, build a larger model or use hand-tailored modules like max-pooling [8]. The former two methods make their training very expensive, and the latter is not that effective.

CNNs are constrained in their ability to model massive, unknown geometric transformations by default. The constraint stems from CNN modules' rigid geometric structures. Generally, a convolution unit will sample the feature

map in the input image at constant and fixed locations, then a pooling layer is used for reducing spatial resolution again at a fixed ratio, and finally, a region of interest (RoI) pooling layer divides a region of interest into fixed spatial bins. As is apparent, this causes the CNNs to be very limited in terms of capability to generalize. This shows that traditional CNNs do not have any internal method to understand geometric variations and transformations.

Jeon and Kim [30] proposed a method called active convolution which added offsets to the sampling points in the convolution and learns the offsets using back-propagation. It has been demonstrated to be useful in picture categorization tasks. These offsets are shared across the spatial places globally and offsets are fixed model parameters that must be learned for each task or training session.

The work by Luo et al. [31] showed that for a traditional CNN, the influence of the pixels near the center on the output is substantially greater than the ones on the peripheral. This shows that the effective receptive field doesn't cover the complete theorized receptive field and forms a Gaussian distribution. They also show that this effective receptive field's size increases much slower than what was expected.

Holschneider et al. [32] presented atrous convolutions which make the convolution filter's stride greater than one and stores the initial weights in sampling locations that are sparsified. This effectively increases the receptive field and does not compromise on the performance. This is widely used in several types of tasks, for semantic segmentation [28, 33, 34].

Previous works trying to adapt to geometric transformations have also tried to learn transformation invariant features. A prerequisite here is to inherently know the transformations that are being worked on and based on this information, a hand-tailored architecture of feature extraction algorithm is created. For example, Lowe et al. [35] performed object detection using local scale-invariant features. The feature extraction algorithm has fixed parameters in this which limits their work to a very specific problem. Our work, however, aims to solve the problem of real-world geometric variations that may work for all types of image-based tasks pertaining to all types of unseen geometric variations.

Dai et al. [27] proposed a Deformable CNNs which majorly consists of convolutional layers that have adaptable and dynamic sampling locations and also have RoI pooling. Compared to the other previous works, they have shown promising results in capturing the receptive fields in an adaptive manner and generalizing to various geometric tasks. Through our implementation (discussed in Section 3.2.3), the deformable layers are also easy to integrate with traditional CNNs making them an ideal fit for our case.

## 2.2 Classical Data Augmentation

Data augmentation refers to the process of expanding the size of a dataset by creating new data points which are created by doing various manipulations on the existing data. We can categorize the tasks of data augmentation

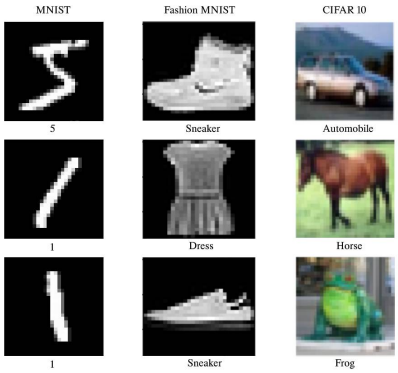


**Fig. 2** Difference in the geometry in images of the same scene but from different angles, distances, and perspectives. In the egocentric view the white car and the road looks highly warped due to perspective. In the bird's eye view, the objects are not warped and they look smaller since the distance of the camera increased.

**Fig. 3** Example pairs of egocentric views (left row) and their corresponding bird's eye view (right row). The egocentric views are highly warped due to perspective and a major part of the environment is out of the field of the camera's view. The bird's eye view shows a much more holistic view of the environment and the scaling is consistent.

methods inot two categories: geometric and non-geometric. Geometric data augmentation methods include cropping, flipping, rotation, translation, and perspective transformation of the original data points. Non-geometric data augmentation methods include manipulation of properties such as the brightness, contrast, color, and saturation of the original data. Non-geometric data augmentation may include other methods as well that apply photometric transformations such as Kernel filters, Random erasing, Mixing images, Color space transformations, etc.

While the various types of augmentations are all effective more or less in our case, since our work focuses only on the geometric transformations, to localize our analysis to the changes that are being proposed, our experiments employ only geometric transformations and do not use the non-geometric augmentation. Datasets like MNIST [36] or SVHN [37] require special consideration and constraints, when and which involve recognition of text and numbers. Only those augmentation methods are used which do not alter the data so much that their labels are invalidated. This factor is termed as safety of data augmentation. It refers to the likelihood of retention of the label post-transformation for the given data. Operations like flipping might end up changing the image such that the label does not provide the ground truth for the image. An example could be 6 flipped about the x and y axis simultaneously making it look like a 9.

Shorten et al. [24] elaborated evidence that showed mixing and combining different augmentation techniques is a lucrative proposition to further increase the size of the dataset, but it might come at a cost of further overfitting the model. Overfitting can be caused by augmentation in domains with very limited data. Another issue with classical data augmentation is that it works using simple handwritten algorithms and these algorithms do not reflect the various transformations one may see in the real-world.

## 2.3 GANs for Data Augmentation

Generative adversarial networks (GANs) [18] have proved to be an important tool for generating unseen datapoints that closely match the input domain. The structure of the network helps it produce better results. The two networks i.e. Generator (g) and Discriminator (d) play a sort of zero-sum game. The generator tries to create synthetic data photorealistic enough so that the discriminator cannot differentiate between the real and the synthesized data.

Using such a network one can augment the dataset by generating synthetic data that is virtually indistinguishable from the real data, effectively increasing the size and diversity of the dataset without any loss in the data's quality. GANs have been used extensively for reconstruction tasks in medical imaging such as for CT [38] and PET [39] denoising, accelerated magnetic resonance imaging [40], and using super-resolution for retinal vasculature segmentation [41]. The work by Adar et al. [42] for liver lesion classification aided by data augmentation using GANs improved the previous sensitivity performance of 78.6% and specificity of 88.4% using classical augmentation techniques to the sensitivity of 85.7% and specificity of 92.4% using GANs for the data augmentation, hence promoting the use of GANs for augmentation of data. Bowles et al. [43] used GANs for generating mixed images and showed that the inclusion of these mixed images in the training dataset improved the diversity of the author's data and decreased the training time. For data augmentation Shorten and Khoshgoftaar [24] say that DCGANs [26], Progressively Growing GANs [44], CycleGAN [23], and Conditional GANs [22] seem to possess the highest level of real-world application potential. [43] describe GANs to be the key

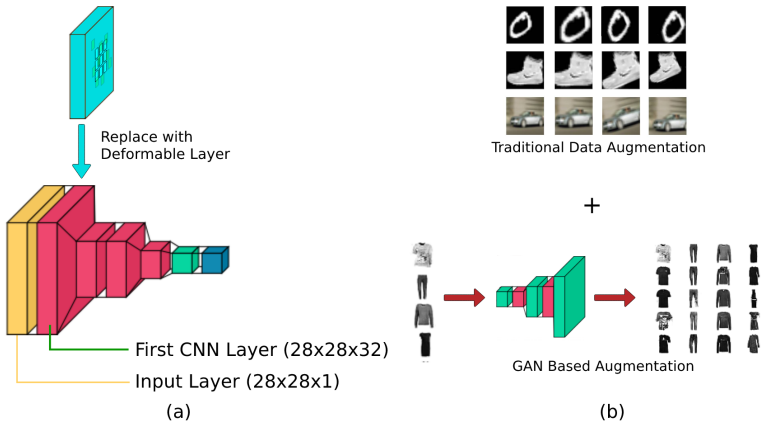


**Fig. 4** Visual overview of the proposed two-step method. In (a) we replace the first convolutional layer of the image-based model with the deformable layer. In (b) we perform a series of steps for data augmentation which include classical data augmentation and GAN based augmentation. The outputs from both are added to the original dataset improving the quality and quantity.

to unlock additional information from a dataset that we otherwise might not have access to.

## 2.4 Image to Image translation

The task of generating a novel viewpoint from a known point of view is known as view translation. Image to image translation pertains to the condition where the input and output domains are made up of images. In one of our experiments we demonstrate the efficacy of our proposed method on translating egocentric view to birds-eye view, a dataset made available by Palazzi et al. [45].

A classical approach to compensate for the camera angle is perspective transformation. Using a mathematical approach called homography [46–50], a plane is resolved and the transformation is applied to correct the perspective. The resulting image generally appear to be distorted and out of proportion.

Learning-based approaches have been gaining popularity as they provide promising results in similar applications. Generative adversarial networks (GANs) [18] have proved to be an important tool for generating the novel viewpoint. The structure of the network helps it produce better results for problems like Image to image translation [22, 23, 51–58] . This approach is best suited for isolated frames or images as it lacks temporal consistency. Video to video translation [59] is similar to the image to image translation but improves upon the temporal consistency.

The aforementioned generative model works well when input and output domain are geometrically aligned. This is a major drawback for applications where large geometric translations are needed.

# 3 Materials and Methods

## 3.1 Dataset

### 3.1.1 Image Classification

Our approach was trained and tested on three datasets Fig. 2, namely MNIST [36], Fashion MNIST [60], and CIFAR-10 [61]. MNIST is a dataset of handwritten digits labelled from 0 to 9. It initially had 70,000 images, split into 60,000 training images and 10,000 testing images. After performing the two augmentation steps, there are a total of 100,000 data points that are split it into a set of 80,000 training images and 20,000 testing images in which 10,000 are deformed through augmentation. The Fashion MNIST is a very similar dataset but of labelled fashion images with 10 classes (e.g. sneakers, dress, suit, etc.). Fashion MNIST dataset's split is the exact same as MNIST and the exact same augmentations are performed. CIFAR-10 is another dataset of 60,000 images of 10 classes of objects (such as automobile, truck, frog, horse, etc.) and initially split into exactly 50,000 training and 10,000 testing data. After performing these two augmentation steps, there are a total 80,000 data points and 20,000 testing images in which 10,000 are deformed through augmentation.

### 3.1.2 View Translation

To show the geometrical generalizing ability of our approach, the decision to test it on the task of view translation, which involves completely changing the geometrical properties of the same object, was taken. For this, a synthetic dataset was chosen, consisting of egocentric images (from a car's point of view) along with their corresponding bird's eye views [62]. The egocentric images are similar to what you would see from a dashcam of a car and the bird's eye view images are similar to what a bird looking down at the car would see. Since our work focuses on improving generalizations while keeping the data size low, our training split includes only 4000 images and the testing split includes 1000 images.

## 3.2 Methods

Our work shows how one can drastically improve the generalization capabilities of small-sized models by processing the data through the aforementioned two layers of data augmentation, and by using a deformable convolutional layers in the model. The proposed method is tested on two different computer vision tasks. For the first experiment, the task of classification of objects from images is undertaken. This majorly involves categorizing an image into a specific category and requires the model to understand what features represent which object/category. The efficacy of our work is shown using three popular datasets, namely MNIST [36], FashionMNIST [60], and CIFAR-10 [61]. For the second experiment, the task of view translation using generative adversarial networks. In this, an image-to-image translation-based generator is trained

to take an input image of an object and asked to output the same object but viewed from a different point of view. For our experiments, egocentric images from a car are used for the input domain and their corresponding bird's eye view is used for the output domain. The efficacy of the proposed method is shown in comparison to the state-of-the-art by [22].

### 3.2.1 Diversifying Dataset

Since our aim is to solve for the geometric deformations in images, only augmentation of the geometric terms (Fig 4) were performed. Images in our dataset are randomly scaled between the realistic range of values [1,2.5]. This means that each image would be scaled between 100% to 250% of its original size. Also, translation of the images randomly within the range [0,0.2] and rotation in either direction within the range of rotation [-45°,45°] was performed. To simulate perspective change, the images were warped keeping one edge fixed and the shrinking or stretching the opposite edge within the range [-1.25,1.25]. Finally, centre crop was performed on all the images to make them of the same size.

### 3.2.2 GAN augmentation

GANs are well known for being able to generate unseen and realistic data based on a distribution that it was trained on. The general architecture of a GAN includes two components, $g$ and a discriminator $d$. The job of the generator is to generate realistic examples relative to the training dataset and the job of the discriminator is to classify an image as realistic or fake. $g$ and $d$ are both trained together in a two-player min-max situation. The input to the generator is a noise vector $v$ which is generally smaller in size as compared to the images in the dataset. The generator learns to map the noise vector to a realistic image that could belong to the dataset. The job of the discriminator is to classify the generated image into one of two categories, namely real (belongs to the dataset) or fake (does not belong to the dataset).

A dataset may contain various combinations of geometric variations for the same object. To explain it simply, as an example, in one image a bicycle is shown up close from the side view, and in another image, a bicycle is shown from a distance and from the top. A GAN trained on such a dataset has the ability to generate results in which a bicycle may be shown from a distance and top view, which wasn't initially present in the dataset. The use of GAN indirectly generates a cartesian product of all such variations present in the dataset, allowing the model to understand how the same object might look in different combinations of transformations which would not be possible with the original dataset alone. The implementation details of our GAN for augmentation is explained in Section 4.1.

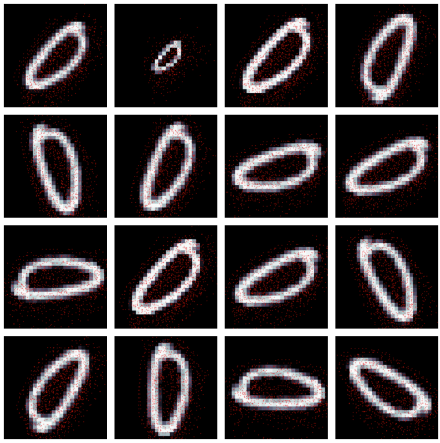

**Fig. 5** The sampling locations of the deformable convolutional filter in the same image but with different geometric orientations. The locations dynamically adapt and generalize for various geometric deformations and orientations.

### 3.2.3 Deformable Layer

The final step in our approach involves making changes to the model directly that makes it more dynamic and adaptable to the several variations in the unseen real-world data. Since the major concern is about generalizing to the geometric variations a deformable convolutional layer from [27], is implemented into our CNN. This layer replaces the very first layer in the CNN (Fig. 7). Fig. 5 shows how the deformable layer generalizes to various geometric deformations in the same image and sets the offsets in order to have dynamic and more accurate sampling locations. For the view translation task, only the deformable layer is employed in the generator since the discriminator works only on the images of the output domain and isn't subject to severe geometric deformation. The involvement of the deformable layer in our view translation experiment is discussed below.

### 3.2.4 View Translation

This section presents a slight theoretical insight for one of our experiments discussed in Section 5. The major task for this experiment is to generate a bird's eye view $y$ given an egocentric input $x$. Simple GANs are only effective in generative image synthesis applications if there is a need to generate new examples of images. There is no real control or access to alteration over the data being generated. To be able to control the outputs and to make use of information additionally supplied to the model, such as class labels, or in our case an input image of egocentric domain $a$ which has to be translated into an image of bird's eye domain $b$, Conditional GANs [20] are used.

In conditional generative adversarial networks, the generator $G$ learns to generate fake samples with a conditioned data point of domain $x$ instead of

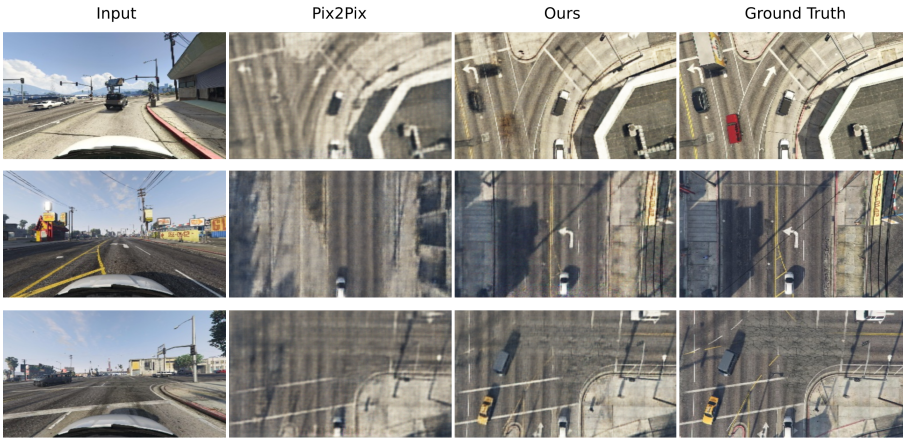


**Fig. 6** A comparison of the state-of-the-art Pix2Pix with our method on the task of egocentric to bird's eye view translation. The pix2pix results are often blurry and lack the geometric understanding of the input and output domains which causes it to generate samples with several artifacts. However, our approach understands the geometric variation much better and generates more realistic looking results.

unknown noise distribution as in simple GANs. We discuss our model and implementation for this in Section 4.1.

# 4 Proposed Methodology

## 4.1 System Model

The generative adversarial network for performing dynamic augmentation consists of a generator $g$ and a discriminator $d$. For this, a GAN architecture was built. The model was provided with the original dataset and was trained separately on each class to generate realistic and unseen data points which are the same as the original data for that class. The input is a random noise vector, and its resulting vector space is called a latent space, or a vector space with latent variables. Latent variables, often known as hidden variables, are variables that are relevant to some domain of data but cannot be directly observed. Points in this vector space will correspond to points in the actual dataset's domain after training, resulting in a compressed representation of the distribution of data. In a similar manner, the discriminator is trained simultaneously to compete with the generator and force it to generate more realistic results.

For view translation, we categorised the task as an image-to-image translation problem. In recent times, image-to-image translation tasks have been explored dominantly by Conditional GANs to achieve tasks like colorization of black-and-white images by Isola et al. [63], future frame prediction [64], image prediction from normal maps [65] etc. We built a conditional GAN whose task was to generate the Bird's eye view of the environment based on the egocentric input it had received. The generator is based on the state-of-the-art

network by Isola et al. [22] and the first convolutional layer of the encoder in the generator is replaced with the deformable layer, and all skip connections are removed (Fig. 7). This ensures that the model is dynamic to the geometric transformations and that it does not assume geometric alignment in the input and output images. The final objective of a our conditional GAN was:

$$\begin{aligned}\mathcal{L}_{cGAN}(G,D) =& \mathbb{E}_{a,b}[\log D(a,b)] + \\ & \mathbb{E}_{a,z}[\log(1 - D(a, G(a,z)))]\end{aligned} \tag{1}$$

Along with the cGAN loss in equation (1), our model also uses the traditional L1 loss. This forces the generator $G$ to generate images near the ground truth output in an L1 sense while also trying to get the discriminator $D$ into believing the generated images are real and belong to the input domain.

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{a,b,z}\left[\|y - G(a,z)\|_1\right] \tag{2}$$

This results in the final objective function as,

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G,D) + \lambda \mathcal{L}_{L1}(G) \tag{3}$$

## 4.2 Architecture and Working

The architecture of the GAN for image augmentation can be seen in Fig. 8. There are two neural network models (as discussed in Section 4.1): generator and discriminator. The generator takes as input a zero dimensional noise vector which is randomly generated. This is sent through a Dense Layer, and is further sent through 3 Convolutional Layer which upsamples the generated vector image into the final dimension (for example $28 \times 28 \times 1$ for the MNIST dataset, $32 \times 32 \times 3$ for the Cifar dataset). In Fig. 8 we show the architecture used for the MNIST and Fashion MNIST dataset. For the CIFAR dataset, we used an altered version of the same architecture where instead of 1 channel, we expect 3 channels in the generator output, and the discriminator input.

The architecture of our conditional GAN can be seen in Fig. 7(b). The whole model can be divided into 2 parts, the generator and the discriminator. The generator, which is a U-net, can further be divided into 2 major sections, encoders (downsampling) and decoders (upsampling). The input to the generator is an egocentric image of dimension $256 \times 256 \times 3$. This input goes is subjected to downsampling through the encoders until the bottleneck layer in the middle. After this, the low dimensional vector is subjected to upsampling through the decoders and the final image (bird's eye view) is received with the dimensions of $256 \times 256 \times 3$. The encoder and decoder consists of 7 Convolutional layers each. Traditionally these encoder and decoder layers are connected using skip connections, however we remove them as these skip connections can force geometric alignment which would be regressive for us.

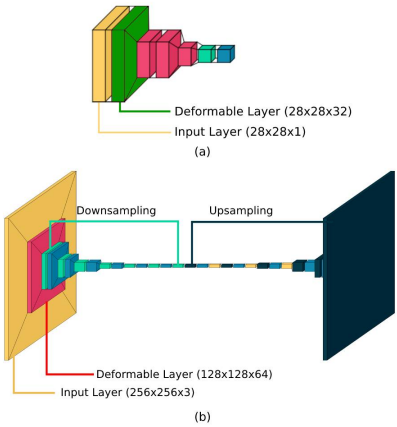


**Fig. 7** The changes to the CNN model (a) and the Pix2Pix model (b). In (a) we replace the first Convolutional layer with the deformable Convolutional layer and in (b) after this step, we also remove the skip connections so that the downsampling and upsampling stages do not assume geometric alignment in each other.

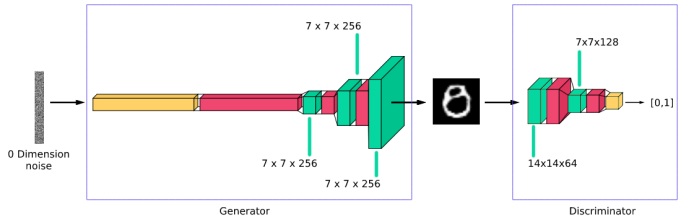


**Fig. 8** The training architecture of our image augmentation GAN model. The generator performs up-sampling on the noise and generates a random image. The generated output is sent to the discriminator for validation.

# 5 Experimentation and Results

## 5.1 Experimental Setup

To perform experimentation, Google's Jupyter Notebook Environment titled "Google Colab" is used with configuration: Nvidia K80 / T4 GPU, 12GB of memory, and 2 CPU Cores. The code for training both the GAN model and the CNN model was implemented in a way to utilise as much GPU power as possible.

To test our approach properly, we performed experiments on 2 computer vision applications: Image Classification and View Translations. Image classification for the past four decades has been a well researched area with real-world applications that impact us everyday. View translation on the other hand is a relatively more complex and less explored area, gaining more popularity as more research on generative adversarial networks have been done. The purpose of choosing these two experiments was to test on a diverse type of tasks.

### 5.1.1 Image Classification

To show the efficacy of our results, several models were trained with different training conditions and compared thereafter. Two variants For the model were used, i.e.- a CNN without the deformable layer and a CNN with the deformable layer. These models were trained and tested on three datasets. For each of the three datasets, two types of training and testing splits were created, one with regular images and one with augmented images. Each model was trained for 200 epochs and was completed in 25 minutes. Inference using the generator model takes on average 130ms (calculated on a set of 50,000 test images).

### 5.1.2 View Translation

Similar to the classification experiment, experiments were conducted with and without the data augmentation step on both the models: ours and Pix2Pix [22] for comparison. Since there is already a severe amount of geometric deformation between the input and output images, unseen augmentation was not performed explicitly for testing the models. If a model has a better and more general understanding of geometric transformations, it'll be able to learn the mapping from the egocentric views and the bird's eye view. This was used to evaluate the models and calculate the distance between the generated images and the ground truth bird's eye view images. The model was trained on the training dataset of 4000 image pairs of egocentric and corresponding bird's eye images. The training was done for 45000 steps and was completed in 182 minutes. Inference using the generator model takes on average 589ms (tested on a set of 1000 test images).

## 5.2 Results

### 5.2.1 Image Classification

Training and testing the models on all variations of the dataset as mentioned in Section 5.1.1 was performed and our findings are presented in Table 1 (DA in the table refers to the Data Augmentation which includes both the classical and the GAN based data augmentation steps). It should be noted that the models were tested on a dataset with added geometric variations in order to simulate real-world samples. We used the same data augmentation step to simulate these variations. Please note that this test dataset is referred to as the 'deformed dataset'. The first column shows the accuracy values of a simple CNN trained on the original versions of each dataset. In the second column, the accuracy of the CNN trained on the augmented datasets is shown. Similarly in the third and fourth columns show the accuracy of the CNN model with the deformable layer trained on original and augmented datasets respectively.

From Table 1 it can be seen that CNN trained without any data augmentation have the worst accuracy for the dataset with deformed images. After being trained on additionally supplied augmented data, it can be seen that the accuracy improves suggesting a higher generalizing capability of the model.

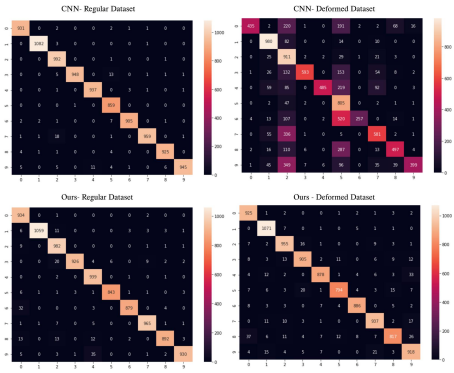


**Fig. 9** A comparison of confusion matrices for the performance of simple CNNs and our method on regular and deformed datasets.


**Fig. 10** Generator and discriminator loss from our method and Pix2Pix. The generator loss converges faster for our method consistently compared to Pix2Pix. This shows that our generator is able to understand and overcome the geometric variations better than Pix2Pix. The discriminator loss for us converges slower than Pix2Pix, however, this is desirable because if the discriminator is confused between real and generated images concluding that the generator works well.

However, the trend from the third column of the table shows that a CNN model with a deformable layer works better than the former even without using the data augmentation. After training the Convolutional Neural Network with the deformable layer on the augmented dataset, much better performance is observed.

Our method works equally well on the original test data when compared with its simple CNN counterpart. Fig. 9 shows the classification accuracy of the simple CNN (top row) and our method (bottom row) on the regular non-augmented test data (left column) and the deformed dataset (right dataset). It can very easily be seen from the top right subfigure that the simple CNN fails in correct classifying images with geometric variations, while our method performs much better than the deformed data points. Table 3 shows the recall and precision values of the CNN model and our method on the MNIST regular and deformed dataset. Both recall and precision values show that our method clearly outperforms the CNN when working on the deformed dataset. On a

**Table 1** Accuracy scores of the various methods trained on different levels of data augmentation (DA). All tests were performed on their respective deformed (Classical and GAN based

| | CNN | | | | Ours | | | |
|---|---|---|---|---|---|---|---|---|
| | Without DA | Classical DA | GAN DA | Both | Without DA | Classical DA | GAN DA | Both |
| MNIST | 0.89 | 0.87 | 0.89 | 0.90 | 0.89 | 0.90 | 0.96 | 0.97 |
| F-MNIST | 0.85 | 0.84 | 0.86 | 0.86 | 0.87 | 0.87 | 0.90 | 0.92 |
| CIFAR 10 | 0.88 | 0.90 | 0.89 | 0.91 | 0.91 | 0.93 | 0.95 | 0.96 |

**Table 2** Quantitative Comparison of our method with the state-of-the-art for image-to-image translation tested on deformed data, both trained on different levels of data augmentation (DA).

| | Pix2Pix | | | | Ours | | | |
|---|---|---|---|---|---|---|---|---|
| | Without DA | Classical DA | GAN DA | Both | Without DA | Classical DA | GAN DA | Both |
| RMSE | 45 | 43 | 44 | 42 | 32 | 31 | 29 | 28 |
| SSIM | 0.65 | 0.66 | 0.68 | 0.69 | 0.78 | 0.82 | 0.88 | 0.89 |

**Table 3** Quantitative comparison of the recall and precision values of a CNN trained without DA and our method on the MNIST dataset.

| | CNN | | Ours | |
|---|---|---|---|---|
| | Recall | Precision | Recall | Precision |
| Regular Dataset | 0.987 | 0.987 | 0.974 | 0.973 |
| Deformed Dataset | 0.615 | 0.788 | 0.945 | 0.947 |

regular dataset, both methods have comparable recall and precision values with our method being very slightly inferior.

### 5.2.2 View Translation

A visual comparison of the image-to-image translation state-of-the-art [22] with our method can be seen in Fig. 6. It can be clearly interpreted that the pix2pix network does not understand large geometric transformations very well and while its results look realistic in terms of textures and colors, it often fails in reconstructing the geometric details accurately in the generated image. Whereas our method significantly improves the geometric accuracy and shows a good understanding of the geometric transformations in the input and output images without compromising on the quality of the textures and retention of the details. The quantitative comparison of the experiments is also shown in Table 2. It can easily be observed that our method outperforms Pix2Pix [22]. While data augmentation helps Pix2Pix a little, by making the model more dynamic to changes in the geometry, the RMSE decreases by 35% and the SSIM increases by 36%.

A rule of thumb in training GANs is that the generator loss should go down faster and the discriminator loss should go down as slow as possible. This is because if the generator loss doesn't go down that means the generator is not able to learn the mappings between the input and output images and if the discriminator loss goes down fast, it means that the discriminator is easily able to classify the generated images as fake and the real images as real, which indirectly means that the generator isn't producing realistic enough results. Fig. 10 shows how the generators and discriminators converge as training progresses. The generator in our method converges faster and the discriminator converges slower than Pix2Pix [22], which is desirable.

## 6 Conclusions and Future Work

Through this paper, our work addressed the major issue of geometric variations in the real-world data and presented a broad strategy to account for the variations and improve the geometric generalization capability of an image-based deep learning model when subjected to unseen geometric variations. Our two step approach which includes data diversification and making the model dynamic makes allows us to keep the dataset and model lightweight. Our work

opens up new avenues for deep learning practitioners to improve their models' generalization efficiently on geometric variations introduced when using real-world data. Our method was tested on two different tasks: image classification and view translation (which are both heavily subjected to unknown geometric variations) and showed that it outperforms the state-of-the-art when subjected to data with geometric variations.

While our work fulfills all objectives mentioned in Section 1, there is a scope for improvement in future research. One possible improvement is using deformable skip connections in the generator network introduced by Siarohin et al. [66]. In our implementation, we had modified the pix2pix network by removing the skip connections as the skip connections assume geometric alignment between the input and output images, which forces the model to be more restrictive to geometric variations. While removing the skip connections completely solved the issue, by using the aforementioned deformable skip connections, we might be able to leverage their dynamic nature and improve the geometric understanding of network.

The method proposed by us can be used as a general approach to solve geometric deformation in real-world images, however we have only shown it's efficacy on two tasks, classification and view translation. More experiments need to be done on tasks such as object detection, video-based deep learning tasks, image segmentation, etc. to show to the practical feasibility of our method as a general approach to solve geometric deformation.

## Data Availability Statement

Authors declare that all the data used in the design and the production cum layout of the manuscript is declared in the manuscript.

## Funding Statement

The authors received no specific funding for this study

## Conflicts of Interest

The authors declare that they have no conflicts of interest to report regarding the present study

## References

[1] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. 2017 IEEE International Conference on Computer Vision (ICCV), 764–773 (2017)

[2] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S.M., Krivokon, M., Gao, A., Joshi,

A., Zhang, Y., Shlens, J., Chen, Z., Anguelov, D.: Scalability in perception for autonomous driving: Waymo open dataset. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2443–2451 (2020)

[3] Yang, G., Song, X., Huang, C., Deng, Z., Shi, J., Zhou, B.: Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 899–908 (2019)

[4] Gao, L., Biderman, S.R., Black, S., Golding, L., Hoppe, T., Foster, C., Phang, J., He, H., Thite, A., Nabeshima, N., Presser, S., Leahy, C.: The pile: An 800gb dataset of diverse text for language modeling. ArXiv **abs/2101.00027** (2021)

[5] Shoeybi, M., Patwary, M.A., Puri, R., LeGresley, P., Casper, J., Catanzaro, B.: Megatron-lm: Training multi-billion parameter language models using model parallelism. ArXiv **abs/1909.08053** (2019)

[6] Raffel, C., Shazeer, N.M., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. ArXiv **abs/1910.10683** (2020)

[7] James, S., Davison, A.J., Johns, E.: Transferring end-to-end visuomotor control from simulation to real world for a multi-stage task. In: CoRL (2017)

[8] Boureau, Y.-L., Ponce, J., Lecun, Y.: A theoretical analysis of feature pooling in visual recognition. In: 27TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING, HAIFA, ISRAEL (2010)

[9] Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 25. Curran Associates, Inc., ??? (2012). https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[10] Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition (2015)

[11] He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition (2015)

[12] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the Inception Architecture for Computer Vision (2015)

[13] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261–2269 (2017). https://doi.org/10.1109/CVPR.2017.243

[14] Yin, M., Sun, L., Li, Q.: Novel view synthesis on unpaired data by conditional deformable variational auto-encoder. In: ECCV (2020)

[15] Lai, Z., Tang, C., Lv, J.: Multi-view image generation by cycle cvae-gan networks. In: ICONIP (2019)

[16] Zhu, X., Yin, Z., Shi, J., Li, H., Lin, D.: Generative adversarial frontal view to bird view synthesis. 2018 International Conference on 3D Vision (3DV), 454–463 (2018)

[17] Weng, C.-Y., Curless, B., Kemelmacher-Shlizerman, I.: Vid2actor: Free-viewpoint animatable person synthesis from video in the wild. ArXiv **abs/2012.12884** (2020)

[18] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 27. Curran Associates, Inc., ??? (2014)

[19] Kingma, D.P., Welling, M.: An introduction to variational autoencoders. Found. Trends Mach. Learn. **12**, 307–392 (2019)

[20] Mirza, M., Osindero, S.: Conditional generative adversarial nets. ArXiv **abs/1411.1784** (2014)

[21] Lecun, Y., Bengio, Y.: In: Arbib, M.A. (ed.) Convolutional networks for images, speech, and time-series. MIT Press, ??? (1995)

[22] Isola, P., Zhu, J.-Y., Zhou, T., Efros, A.: Image-to-image translation with conditional adversarial networks, pp. 5967–5976 (2017). https://doi.org/10.1109/CVPR.2017.632

[23] Zhu, J.-Y., Park, T., Isola, P., Efros, A.: Unpaired image-to-image translation using cycle-consistent adversarial networks, pp. 2242–2251 (2017). https://doi.org/10.1109/ICCV.2017.244

[24] Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. Journal of Big Data **6**(1), 60 (2019). https://doi.org/10.1186/s40537-019-0197-0

[25] Brock, A., Donahue, J., Simonyan, K.: Large scale GAN training for

high fidelity natural image synthesis. CoRR **abs/1809.11096** (2018) arXiv:1809.11096

[26] Radford, A., Metz, L., Chintala, S.: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks (2016)

[27] Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable Convolutional Networks (2017)

[28] Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**, 640–651 (2017)

[29] Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)

[30] Jeon, Y., Kim, J.: Active convolution: Learning the shape of convolution for image classification. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1846–1854 (2017)

[31] Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. In: NIPS (2016)

[32] Holschneider, M., Kronland-Martinet, R., Morlet, J., Tchamitchian, P.: A real-time algorithm for signal analysis with the help of the wavelet transform. (1989)

[33] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**, 834–848 (2018)

[34] Yu, F., Koltun, V.: Multi-scale context aggregation by dilated convolutions. CoRR **abs/1511.07122** (2016)

[35] Lowe, D.: Object recognition from local scale-invariant features. Proceedings of the Seventh IEEE International Conference on Computer Vision **2**, 1150–11572 (1999)

[36] LeCun, Y., Cortes, C.: MNIST handwritten digit database (2010)

[37] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading Digits in Natural Images with Unsupervised Feature Learning. NIPS Workshop on Deep Learning and Unsupervised Feature Learning (2011). http://ufldl.stanford.edu/housenumbers

[38] Wolterink, J., Leiner, T., Viergever, M., Igum, I.: Generative adversarial networks for noise reduction in low-dose ct. IEEE Transactions on Medical Imaging **36**, 2536–2545 (2017)

[39] Wang, Y., Yu, B., Wang, L., Zu, C., Lalush, D., Lin, W., Wu, X., Zhou, J., Shen, D., Zhou, L.: 3d conditional generative adversarial networks for high-quality pet image estimation at low dose. NeuroImage **174**, 550–562 (2018)

[40] Shitrit, O., Riklin-Raviv, T.: Accelerated magnetic resonance imaging by adversarial neural network. In: DLMIA/ML-CDS@MICCAI (2017)

[41] Mahapatra, D., Bozorgtabar, B.: Retinal vasculature segmentation using local saliency maps and generative adversarial networks for image super resolution. ArXiv **abs/1710.04783** (2017)

[42] Frid-Adar, M., Diamant, I., Klang, E., Amitai, M., Goldberger, J., Greenspan, H.: Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. Neurocomputing **321**, 321–331 (2018). https://doi.org/10.1016/j.neucom.2018.09.013

[43] Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A., Dickie, D., Hernández, M.V., Wardlaw, J., Rueckert, D.: Gan augmentation: Augmenting training data using generative adversarial networks. ArXiv **abs/1810.10863** (2018)

[44] Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. ArXiv **abs/1710.10196** (2018)

[45] Palazzi, A., Borghi, G., Abati, D., Calderara, S., Cucchiara, R.: Learning to map vehicles into bird's eye view. In: International Conference on Image Analysis and Processing, pp. 233–243 (2017). Springer

[46] Agarwal, A., Jawahar, C.V., Narayanan, P.J.: A survey of planar homography estimation techniques. Technical report (2005)

[47] Jain, P., Jawahar, C.V.: Homography estimation from planar contours, pp. 877–884 (2006). https://doi.org/10.1109/3DPVT.2006.77

[48] Li, X., Fang, X., Wang, C., Zhang, W.: Lane detection and tracking using a parallel-snake approach. Journal of Intelligent I& Robotic Systems **77**, 597–609 (2015)

[49] Kholopov, I.S.: Bird's eye view transformation technique in photogrammetric problem of object size measuring at low-altitude photography. In: AIME 2017 (2017)

[50] Abbas, A., Zisserman, A.: A geometric approach to obtain a bird's eye view from an image. 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), 4095–4104 (2019)

[51] Huang, X., Liu, M.-Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation (2018)

[52] Bousmalis, K., Silberman, N., Dohan, D., Erhan, D., Krishnan, D.: Unsupervised pixel-level domain adaptation with generative adversarial networks, pp. 95–104 (2017). https://doi.org/10.1109/CVPR.2017.18

[53] Liu, M.-Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks (2017)

[54] Liu, M.-Y., Tuzel, O.: Coupled generative adversarial networks (2016)

[55] Shrivastava, A., Pfister, T., Tuzel, O., Susskind, J., Wang, W., Webb, R.: Learning from simulated and unsupervised images through adversarial training, pp. 2242–2251 (2017). https://doi.org/10.1109/CVPR.2017.241

[56] Taigman, Y., Polyak, A., Wolf, L.: Unsupervised cross-domain image generation (2016)

[57] Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans, pp. 8798–8807 (2018). https://doi.org/10.1109/CVPR.2018.00917

[58] Zhu, J.-Y., Zhang, R., Pathak, D., Darrell, T., Efros, A., Wang, O., Shechtman, E.: Toward multimodal image-to-image translation (2017)

[59] Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Liu, G., Tao, A., Kautz, J., Catanzaro, B.: Video-to-video synthesis. In: Advances in Neural Information Processing Systems (NeurIPS) (2018)

[60] Xiao, H., Rasul, K., Vollgraf, R.: Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms (2017)

[61] Krizhevsky, A., Nair, V., Hinton, G.: Cifar-10 (canadian institute for advanced research)

[62] Palazzi, A., Borghi, G., Abati, D., Calderara, S., Cucchiara, R.: Learning to map vehicles into bird's eye view. In: International Conference on Image Analysis and Processing, pp. 233–243 (2017). Springer

[63] Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016)

[64] Mathieu, M., Couprie, C., Lecun, Y.: Deep multi-scale video prediction beyond mean square error (2015)

[65] Wang, X., Gupta, A.: Generative image modeling using style and structure adversarial networks, vol. 9908, pp. 318–335 (2016). https://doi.org/10.1007/978-3-319-46493-0_20

[66] Siarohin, A., Sangineto, E., Lathuilière, S., Sebe, N.: Deformable gans for pose-based human image generation. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 3408–3416 (2018)