Enriching Diversity of Synthetic Images for Person Detection

Graduation Thesis report

Chinmay Polya Ramesh

Delft University of Technology

Enriching Diversity of Synthetic Images for Person Detection

by

Chinmay Polya Ramesh

To obtain the degree of Master of Science in Robotics at the Delft University of Technology.

Delft University of Technology

Student Number:	5135125
Thesis Duration:	August 02, 2021 - October 20, 2022
Thesis Committee:	Dr. L. Zhang, Philips, supervisor
	Dr. H. Caesar, TU Delft, supervisor
	Dr. J.F.P. Kooij, TU Delft

Abstract

Camera-based patient monitoring is undergoing rapid adoption in the healthcare sector with the recent COVID-19 pandemic acting as a catalyst. It offers round-the-clock monitoring of patients in clinical units (e.g. ICUs, ORs), or at their homes through installed cameras, enabling timely, pre-emptive care. These are powered by Computer Vision based algorithms that pick up critical physiological data, patient activity, sleep pattern, etc., enabling real-time, pre-emptive care. In this work, we develop a person detector to deploy in such scenarios. These algorithms require huge quantities of training data which is often in shortage in the healthcare field due to stringent privacy norms. Therefore looking for solutions to enrich clinical data becomes necessary. An alternative currently popular among the Computer Vision community is to use synthetic data for training, created using 3D modeling software pipelines. However, this type of technique often has limitations in data diversity and data balancing as desired variations need to be provided explicitly. In this thesis, we propose a data augmentation method for enriching diversity in synthetic data without using any additional external data or software. In particular, we introduce a pose augmentation technique, which synthesizes new human characters in poses unseen in the original dataset using Pose-Warp GAN. Additionally, a new metric is proposed to assess diversity in human pose datasets. The proposed method of augmentation is evaluated using YOLOv3. We show that our pose augmentation technique significantly improves person detection performance compared to traditional data augmentation, especially in low data regimes.

Acknowledgements

Firstly I would like to express my gratitude to my supervisor at Philips, Lu Zhang, for providing this topic and the internship. Our weekly meetings helped me stay motivated and bounce off ideas and navigate this thesis.I would also like to thank my supervisor at TU Delft, Holger Caesar for his extensive feed-backs and guidance to instill academic rigor.

Lastly, I want to thank my girlfriend Srinidhi, my brother Chethan, my parents(Ramesh, and Shashikala), and my friends Godwin and Darshan who were my comic relief during this time.

Chinmay Polya Ramesh
Den Haag, October 2022

Contents

Nomenclature

Abbreviations

Abbreviation	Definition
YOLO	You Only Look Once
PCA	Principal Component Analysis
t-SNE	t-Distributed Stochastic Neighbor Embedding
GAN	Generative Adversarial Networks
SMOTE	Synthetic Minority Oversampling Technique
GMM	Gaussian Mixtures Model
SMPL	Skinned Multi-Person Linear Model
RGB	Red Green Blue
mAP	Mean Average Precision
CCTV	Closed Circuit Television
GPU	Graphic Processing Unit
Re-ID	Re-identification
AIC	Akaike Information Criteria
PgGAN	Progressive Generative Adversarial Network
DCGAN	Deep Convolutional GAN
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
COCO	Common Objects In Context
SURREAL	Synthetic hUmans foR REAL tasks
RCNN	Region-based Convolutional Neural Network
VOC	Visual Object Classes
CycleGAN	Cycle Consistent Adversarial Networks

Symbols

Symbol	Definition
μ_{nb}	Mean Neighbours
t	Similarity Threshold(between 0,1)
P_I	Pose dataset Interpolated
P_{IF}	Pose dataset Interpolated & Filtered
S_{ij}	Similarity Score between pose i and pose j

1 Introduction

1.1. About the Thesis

This thesis topic was offered by the Philips Research, Patient care Informatics and Analytics team, part of the Patient care & monitoring Department which based out the High Tech Campus in Eindhoven, Netherlands.

1.2. Context

Remote/Contact-less Patient Monitoring is revolutionizing the patient care sector. Video cameras installed inside hospital wards, Intensive Care Units (ICUs), or at homes are used to monitor patients at real time. The Image feed is processed using Computer Vision algorithms to measure vital signs (e.g. heart rate, respiration rate, blood oxygenation saturation, body temperature, etc.) and also non vital signs (fall prevention, sleep monitoring, activity monitoring etc,.). Periodic observations by staff proves costly in nurse hours. It can be subject to inconsistencies, disturb the patients (especially at night) and has the potential for cross-contamination. Camera based monitoring on the other hand, facilitates round the clock monitoring, thus allowing sudden and unexpected deterioration to be recognised early, and treated quickly. Owing to these advantages, Philips is interested in developing its camera-based monitoring solutions.

In this research we are only interested in person detection, as a first step towards towards more fine-grained tasks. The objective is to develop a general purpose, efficient person detector that is trained on minimum real images. The detector is not just limited to hospital wards but can also be used in hallways, common rooms, lobbys and at homes. However developing algorithms for these camera based techniques required large amount of sensitive data. To train state-of-the-art person detectors requires high quality, high quantity images of patients in hospital rooms sometimes in vulnerable states. This causes several problems:

- Privacy of patients is breached.
- Stringent norms like HIPAA (US) and GDPR (EU) regarding collection, handling, storing and utilizing patient data.
- These norms allow de-identification (removing all direct identifiers from patient data) but the process is expensive. There are huge penalties if data is leaked[4].
- Lack of sufficiently large datasets.
- Lack of diversity in datasets.

To combat problems surrounding real data, the use of synthetic data in Machine Learning is currently on the rise with many field adapting and proving its effectiveness.

1.3. Synthetic data

The Remote Sensing team at Philips has been exploring methods to generate synthetic data with the intention of training person detectors using transfer learning on real data. The figure 1.1 shows the current pipeline being used to generate synthetic lab images containing people.

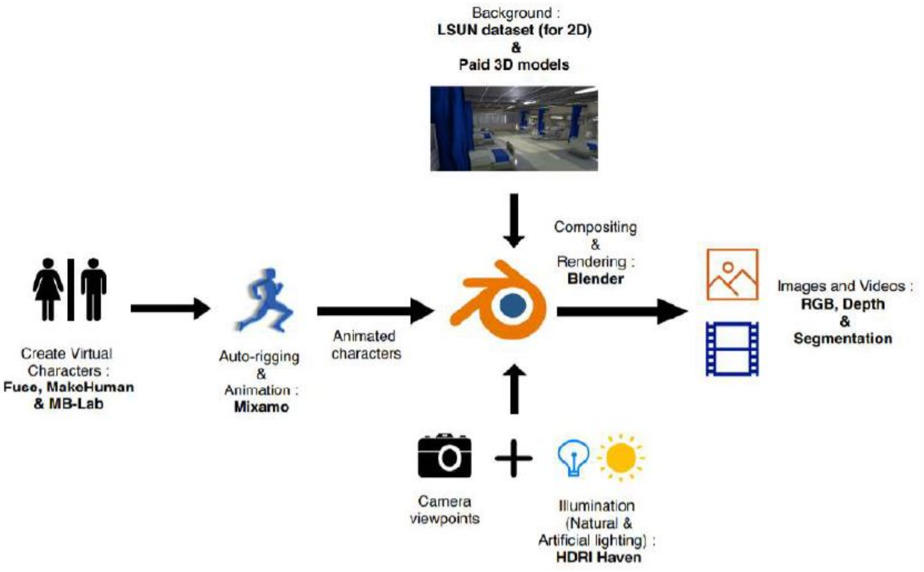


Figure 1.1: Pipeline for generating synthetic data. Image borrowed from Philips.

A set of characters (patients, hospital staff, and visitors) are initially created using modelling tools having different attributes (gender, age, built, and ethnicity). The characters are exported to an animation software to introduce different actions like walking, sitting, standing etc. These characters are imported to blender along with 2D image backgrounds and rendered to produce RGB images, segmentation masks and annotation text files in different camera views as output. Figure 1.2 and 1.3 show an example outputs from the pipeline.

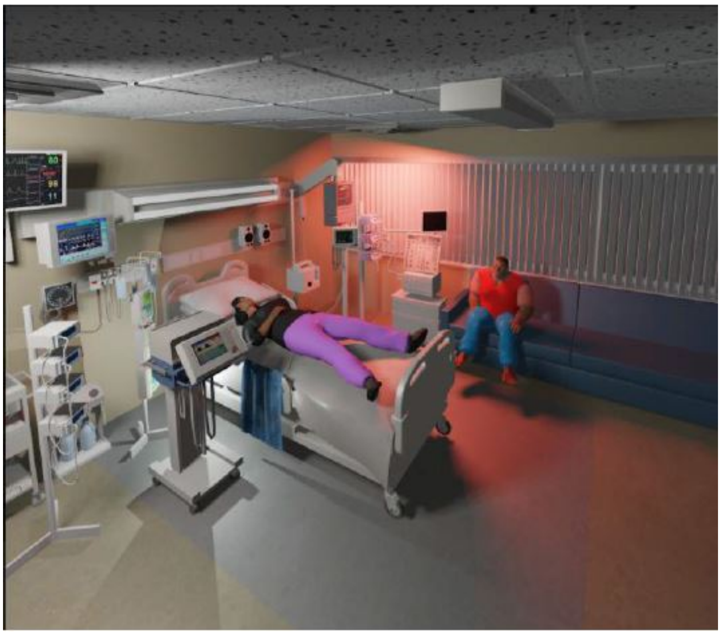
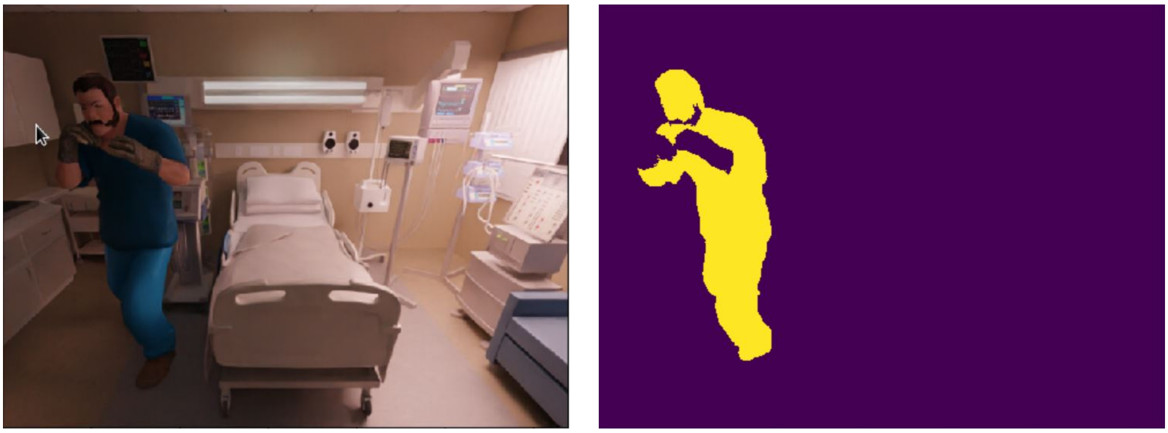


Figure 1.2: Example RGB output from Blender. Image borrowed from Philips

Although the generated images look promising, it has only a few variations of human poses, camera viewpoints, illumination, textures and colors. In order to include more permutations and combinations manual work is necessary. Also certain poses and effects like blankets draped on the body of a sleeping person do not seem natural in a way that appears in real world. Therefore the idea is to investigate potential elegant methods that requires less hard coding of variables to enrich the synthetic dataset.



(a) RGB Image

(b) segmentation mask

Figure 1.3: Example output from Blender pipeline. Image borrowed from Philips.

Another approach is to use Generative Adversarial Networks(GAN's). GANs have been used extensively to augment Medical Image datasets. It would be interesting to see if it could be applied to our case.

1.4. Generative Adversarial Networks

Generative adversarial networks(GANs) are being used increasingly across fields for different tasks like generating images of people[25], fruits[5], mammograms [37] [3], data augumentation[23], image translation from segmentation map to RGB[47], outline sketches to full pictures[21] to mention a few examples. All these examples utilize GANs to generate fake images that try to mimic the distribution of real images present in the training set. Some examples are given below in fig??.

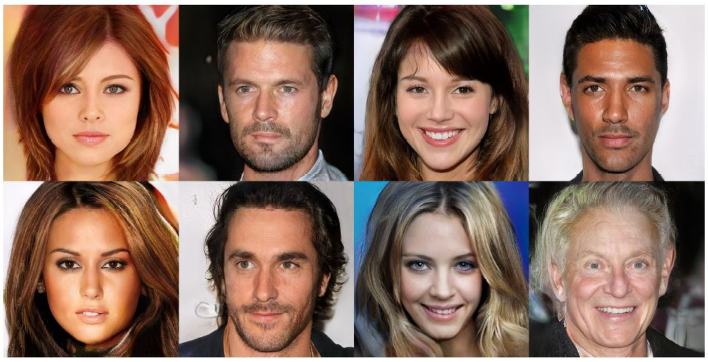


Figure 1.4: High resolution(1024^2) images of fake persons obtained from PGGAN[26] network

The revolution regarding generative networks was spurred by the landmark paper[16] that introduced GAN to the DL community by Ian Goodfellow et al., in 2014. The vanilla GAN consists of two fully connected networks that are pitted against each other in a zero sum game. The two networks are the Generator network and the Discriminator network respectively. Refer to fig1.5.

- Noise vector/Latent vetor(Z): A vector is drawn from randomly from a Gaussian distribution $N(0,1)$, and the vector is used to seed the generative process. After training, points in this multidimensional vector space will correspond to points in the input data but in a compressed low dimensional format.
- Generator(G): The job of the Generator is to learn a function that maps this random input noise vector to an image that is within the distribution of the training data.

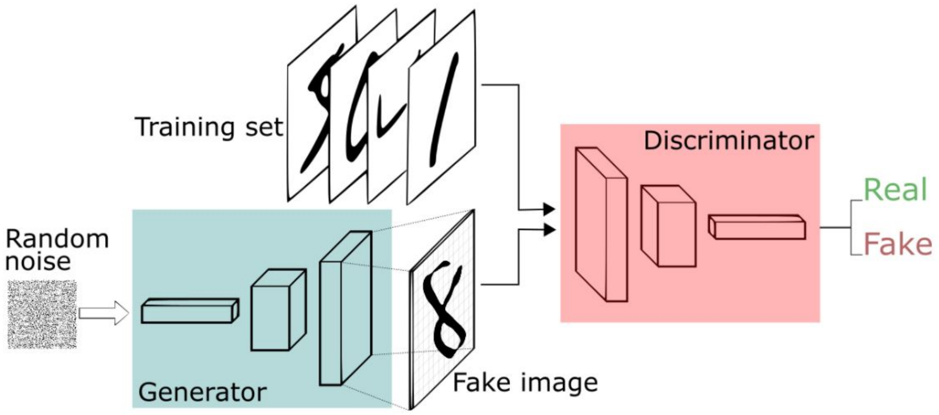


Figure 1.5: Vanilla GAN architechture, source

G(Z) corresponds to the fake image. The generator update equation is given by:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \tag{1.1}$$

This has the effect of encouraging the generator to generate samples that have a low probability of being fake.

- Discriminator(D): The job of the discriminator is to distinguish the fake images generated by the generator from real samples drawn from the training data. D(G(Z)) is either 0 or 1 depending on the Discriminators assumption of the source of the image.
  The Discriminator update equation is given by:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(x^{(i)}\right) + \log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right] \tag{1.2}$$

This has the effect of the discriminator trying to maximize the average of the log probability of real images and the log of the inverse probability for fake images.

The architecture is called adversarial because the gain in performance of one network is at the cost of deterioration of the other while playing the minmax(D,G) game. In the end, the networks are said to converge when the accuracy of the discriminator is 50%. That is when the Discriminator is not able to tell the fake image from the real. While GANs can be a viable solution to create images, there are some inherent problems with GANs such as:

- Training is sensitive to hyperparameters.
- Generated images are low quality.
- Generator fails to create diverse samples because it only cares that the discriminator is fooled. There is no control on output of the generator.

The last two points are of importance, given that our aim is to enhance the diversity of synthetically generated images.

1.5. Goal

The end goal of this thesis is to improve the performance of a person detector trained on synthetic images. Figure 1.6 shows the existing pipeline. Figure 1.7 shows the proposed pipeline. In the introduction we explained how the Philips dataset being used currently is lacking in diversity. A possible direction is to explore GANs as a method for augmenting synthetic datasets. However, Vanilla GANs are notorious for mode collapse and struggle with handling complex data(images of humans in different poses). Therefore, in the next Section, we look at the kind of synthetic datasets being used currently for computer vision tasks in the literature. We also look at how GANs are modified for augmenting datasets, particularly human image synthesis. We then proceed to formulate our research questions.

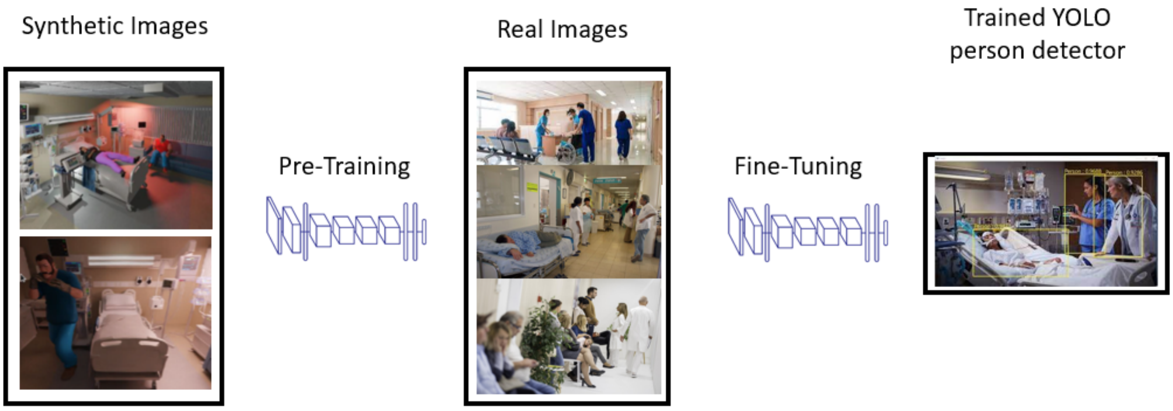


Figure 1.6: Existing pipeline for training a person detector

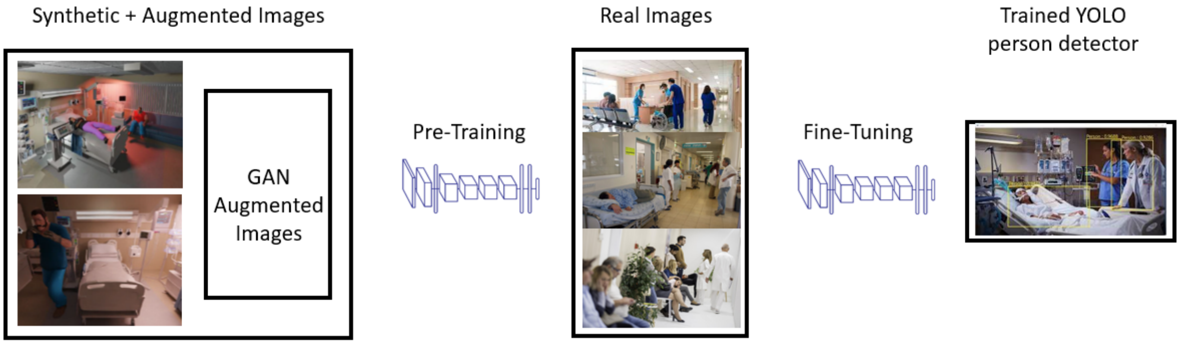


Figure 1.7: Proposed pipeline for training a person detector

2
Literature Study

2.1. Synthetic datasets in object detection

In this section we look at some examples of synthetic data being used in object detection training and how it fares against real data.

In [33], the authors take a comprehensive look into the effects of replacing real data with synthetic data for two types of training; mixed training and fine-tuning. It is reported that fine-tuning synthetic training model with limited real data provides better results than mixed training. The authors claim that for the task of pedestrian detection, using only 10% of real data while fine tuning provides decent results when pretrained on synthetic data (Figure 2.1). Also, It is shown that the photo-realism is not as important as the diversity of the data. Figure 2.2 shows example synthetic images used.

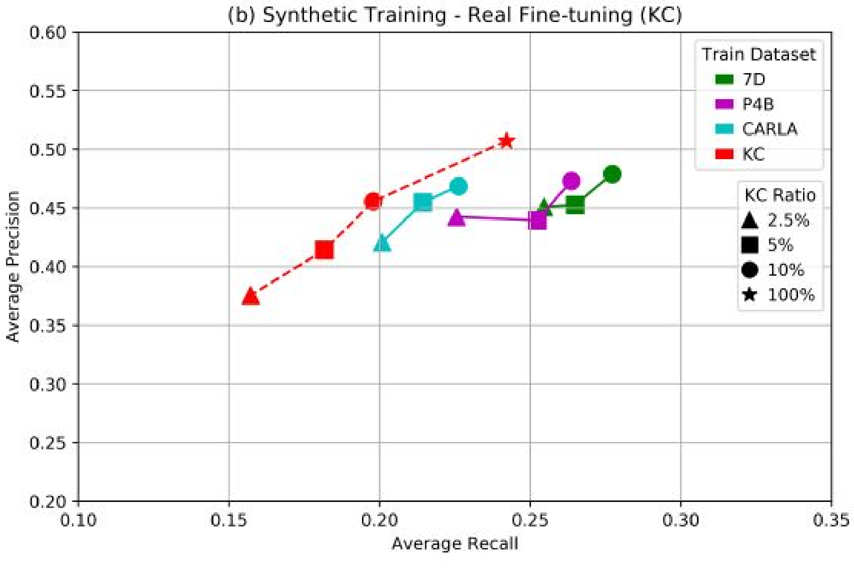


Figure 2.1: Graph shows that fine-tuning model trained on synthetic data using only 10% of real data still performs relatively well. However model performance deteriorates drastically on further reduction on real data proportion[33].

The domain randomization technique[41] introduced by NVIDIA plays with lighting, pose, object textures, in randomized and non-realistic ways to force the neural network to learn the essential features of the object of interest. The approach is evaluated on bounding box detection of cars on the KITTI dataset using Faster-RCNN. The authors observed that when using synthetic dataset to train, it resulted in an AP(average precision) of 83.7% compared to an AP of only 56.1% when only COCO weights were used.

Similarly, [29] tested synthetic data on performance of remote sensing image aircraft detection. Although synthetic data could not overtake the performance of 100% real data, the authors suggest that image translation

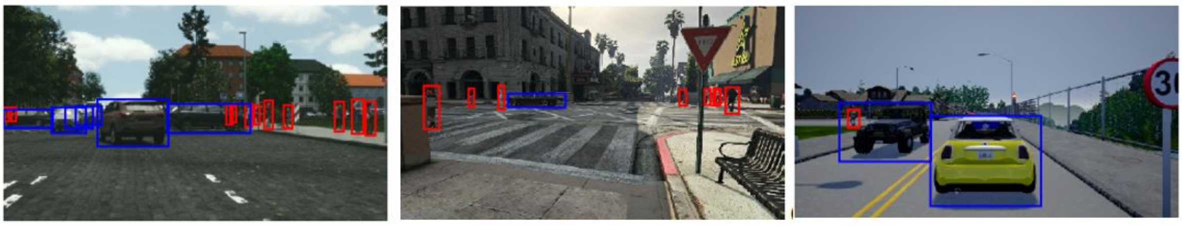


Figure 2.2: Synthetic images of street scenes sourced from CARLA driving simulator and SYNSCAPES 7D[33]

from synthetic to real domain improves mAP by 40%.

With regards to usage of synthetic data in the context of person detection, [44] used a method of pasting blender generated person image patches on randomly sampled images from original data (Figure 2.3). A similar approach has been followed by [18] for the task of pose estimation. Multiple synthetic humans sourced from the SURREAL[42] dataset are pasted on to the training set(Figure 2.4). The authors claim that networks trained on such augmented datasets perform significantly better compared to that trained on the original dataset. Synthetic data could effectively include diverse instances of occlusions and special artefacts and this increased model performance. Surprisingly it was found that improving visual appearance of synthetic humans decreased the accuracy.

Similar approach is followed for the tasks of pedestrian detection, re-identification, segmentation, and tracking in [12] and MOTsynth [13]. The authors show that real data can be completely replaced by 100% synthetic data without a drop in detection performance. In Figure 2.5, an unannotated image set is taken as input. The scale ratio of the pedestrian and vanishing points are extracted to calculate camera parameters. Synthetic 3D characters basic animations are rendered with location-aware lighting and extracted camera parameters. The synthetic pedestrians are overlayed on real backgrounds to produce an annotated training dataset, which is further used to train a pedestrian detector. The trained pedestrian detectors outperform other general-purpose pedestrian detectors by 5–13%.

In [20], authors improved multi-person 2D pose estimation by adding extra 3D modelled human images depicting ”rare” poses from MPII human pose dataset. Backgrounds are randomly cropped patches from randomly selected samples of VOC2012 dataset(Figure 2.6). The reason for improvement in pose detection is attributed to an even distribution in the diversity of poses along with difficult occlusions present in the training set.

Overall these examples show that there is a potential to use synthetically generated datasets for training our YOLO person detector, and expect reasonable performance.

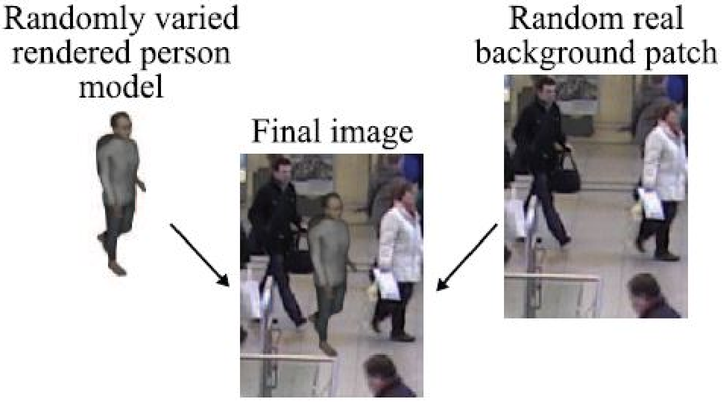

Randomly varied rendered person model
Final image
Random real background patch

Figure 2.3: Data is augmented by pasting blender generated human image patches over random images sampled from original data to create additional annotations[44]

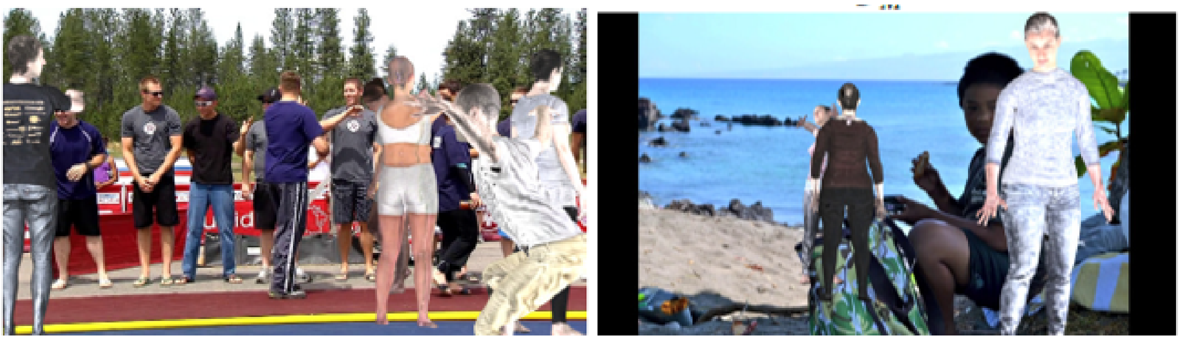


Figure 2.4: Real images of people are augmented by pasting 3D rendered person images over them [18]

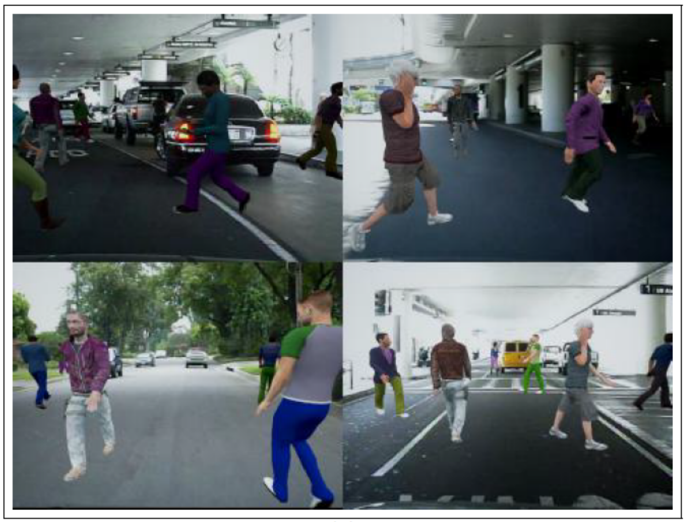


Figure 2.5: Synthetic pedestrians overlayed on real backgrounds in [12]. Agents look realistic with different appearances and brightness similar to the scene.

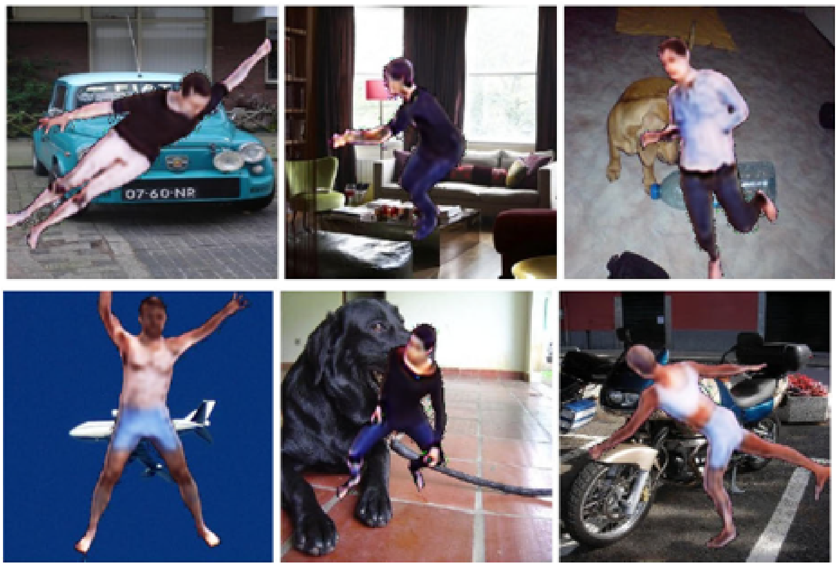


Figure 2.6: Synthetic pedestrians rendered using SMPL mesh and overlayed on real backgrounds in [20]

2.2. GANs for data augmentation

It is also possible to generate more data from existing data by applying various transformations to the original dataset. Some of them include random translations, rotations and flips as well as addition of Gaussian noise. These transformations are quite easy and do not add much information to the data. On the other hand Generative Adversarial Networks(GANs) attempt to learn the internal representations of data and synthesize new samples. [14] describes GANs as a way to “unlock” additional information from a dataset.

In this section we look at examples of Generative Networks as a data augmentation technique and how effective it is in enhancing small datasets.

GAN-based Data Augmentation was used by [14] for liver lesion classification. They report improved classification performance from 78.6% sensitivity and 88.4% specificity using classic augmentations to 85.7% sensitivity and 92.4% specificity using GAN-based Data Augmentation. Figure 2.8 shows some examples of the generator output. In Data Augmentation Using GANs[40], the authors generated totally synthetic data for a binary classification problem (cancer detection). Strikingly, they showed that a decision tree classifier performed better when trained on this totally synthetic dataset than when trained on the original small dataset. As another instance, [7] showed that greatest improvements through GAN augmentation can be seen in the cases where real data is the most limited in the case of brain segmentation task(Figure 2.7).

In[32]GAN-based augmentation is used for detection of Pneumonia and COVID-19 in chest X-ray images. The authors reports moderate improvements in AUC in anomaly detection. Besides medical applications, in [48] CycleGANs[47] were used in the task of emotion classification to improve accuracy by 5−10%. In most of these cases, GAN augmenting approach has been shown to work best in cases of limited data, either through a lack of real data or as a result of class imbalance.

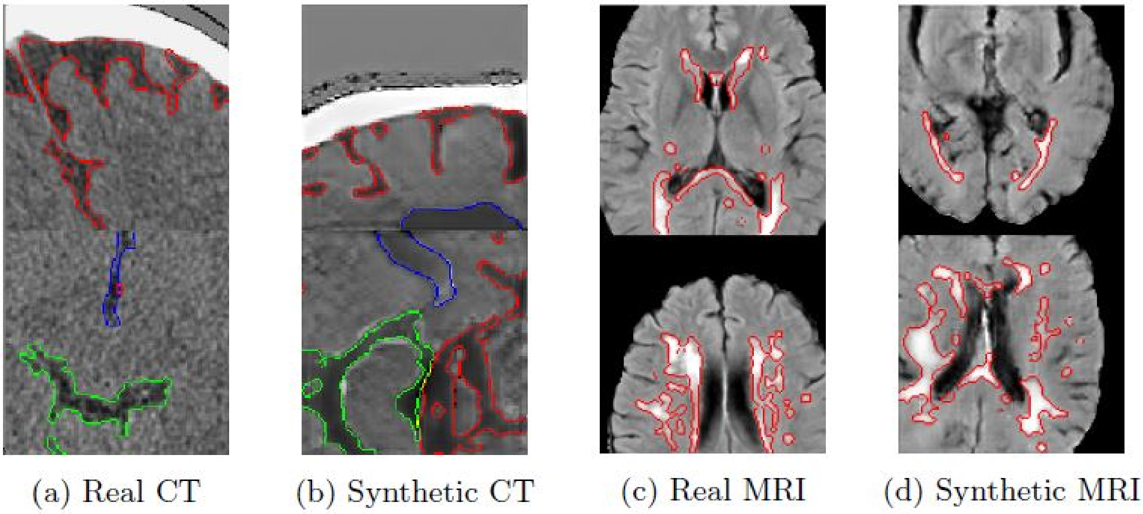

(a) Real CT (b) Synthetic CT (c) Real MRI (d) Synthetic MRI

Figure 2.7: Example images of real and GAN generated images of brain scans for comparison[7]

Most of the augmentations covered above focus on improving downstream image classification, segmentation and few-shot learning applications.

In the case of object detection, GANs are used to augment scarcely available LiDAR point cloud datasets in [38](Figure 2.9). The authors also introduce a new method of annotating these generated images. The YOLO network is trained on the GAN augmented dataset to obtained improved results with respect to non-augmented dataset. In [17] the authors propose a CycleGAN+YOLO combination for data augmentation to train a multi-organ detector for CT images. The approach achieves accurate detection with a significant improvement over YOLO detection alone. [34] investigates GAN augmentation in Structural Adhesive Inspection application. Figure 2.10 shows example results of StyleGAN2 network trained on images of defective structural adhesive beads. The authors claim almost 30% improvement in YOLO detection mAP on the StyleGAN based augumented

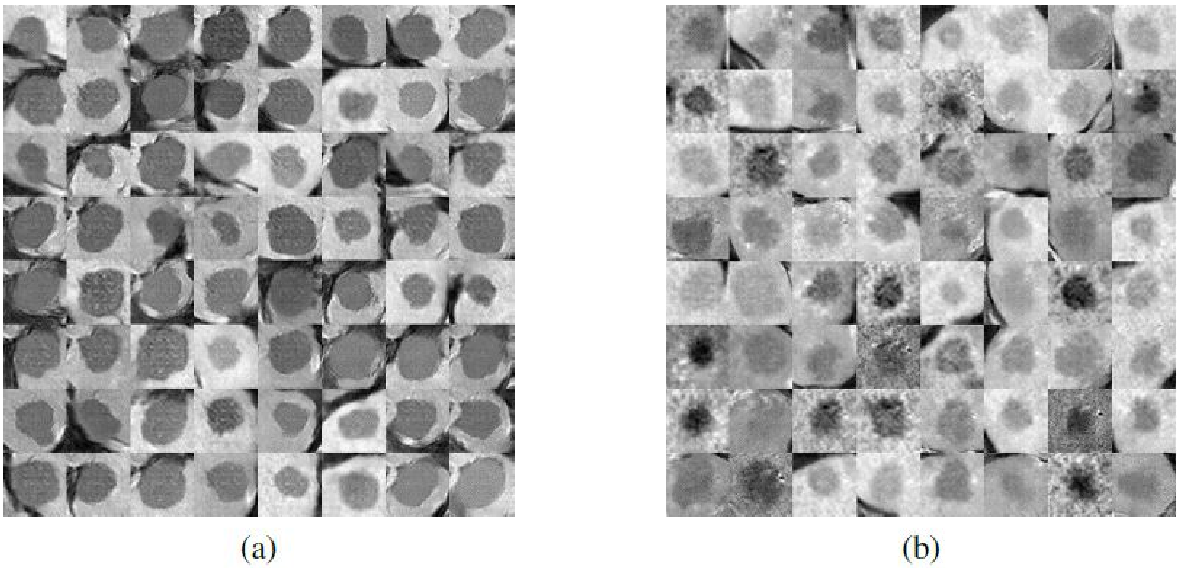


Figure 2.8: Synthetic liver lesion ROIs generated with DCGAN for each category: (a) Cyst examples (b) Metastasis examples [14]

dataset(Figure 2.11). Other applications include image-to-image translation network for generating large-scale trainable data for vehicle detection algorithms[19] and Large scale aerial data sets for environmental perception of autonomous aerial vehicles[31].

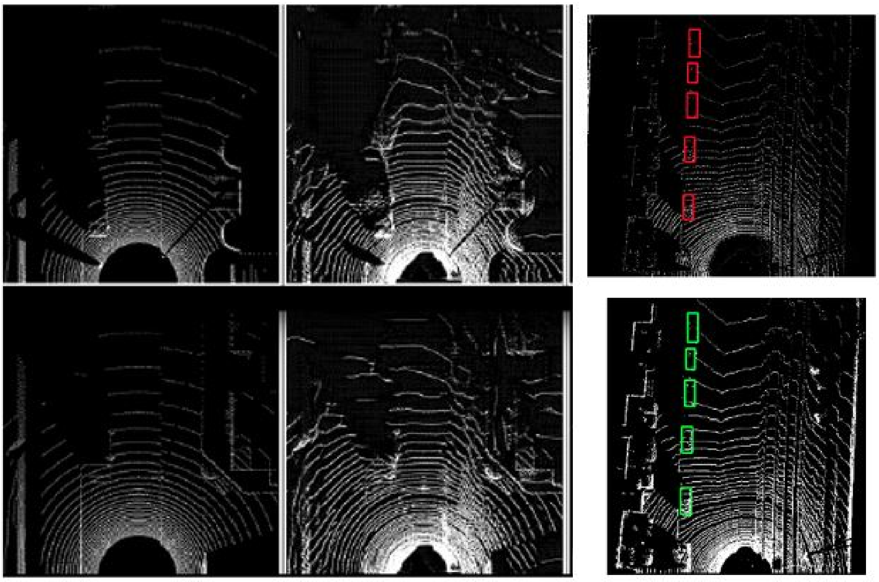


Figure 2.9: First column shows LiDAR scan images from simulated CARLA. Second column shows fake LiDAR scans translated from the domain of simulated images to the real domain using GANs. These scan images have characteristics/artefacts that are typical to real scan images. Last column shows YOLO object detection performed on the fake GAN generated images.[38]

2.3. Human Image Synthesis using conditioning

It seems that the methods discussed in the section have limited editable capacity in the synthesis of full length human images. Although high resolution images are possible, the methods mostly dealt with images having a set layout like faces which are mostly centered. In contrast, the manifold of human images is very complicated which can have infinite variations in size, poses, textures, viewpoints etc,. which becomes very difficult for

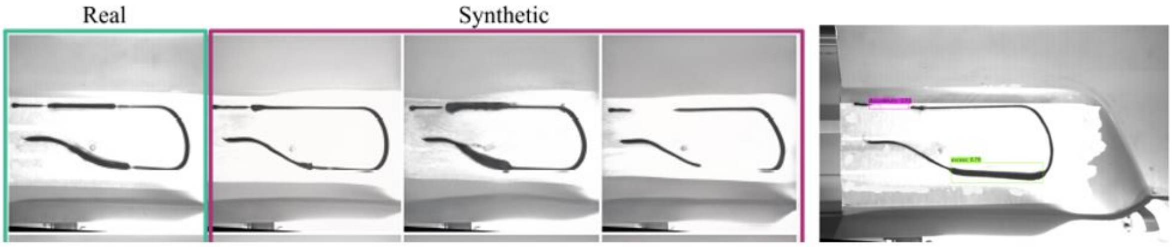
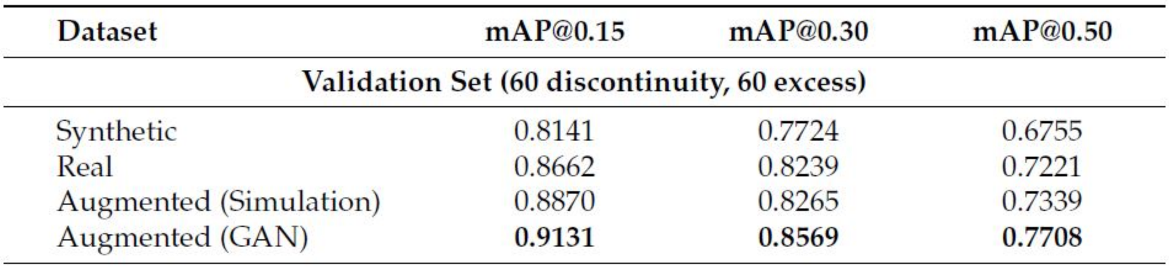


Figure 2.10: Real and synthetic images of defective structural adhesive beads. First column depicts real images from the original dataset. Second, third and fourth columns correspond to synthetic images. Last columns shows defect detection example using YOLO network. [34]



Dataset	mAP@0.15	mAP@0.30	mAP@0.50
Validation Set (60 discontinuity, 60 excess)			
Synthetic	0.8141	0.7724	0.6755
Real	0.8662	0.8239	0.7221
Augmented (Simulation)	0.8870	0.8265	0.7339
Augmented (GAN)	**0.9131**	**0.8569**	**0.7708**

Figure 2.11: Table reported in [34] shows network trained on GAN augmented dataset shows superior performance compared to that trained on purely synthetic or purely real datasets.

GAN networks to learn. Therefore an approach could be to disentangle these variations and target specific important changes that induce maximum diversity in images. This would facilitate targeted image synthesis and decrease the learning task for GANs. In this section we look at some of human image synthesis techniques that are conditioned such that a target image is synthesized based on the given condition.

2.3.1. Domain Transfer

There examples of CycleGAN technique being employed for tasks related to human images. In [22] a mapping from person images to their respective poses is learned(Figure 2.12(a)). In [10], a video to video translation method using pose keypoints as an intermediate representation is discussed (Figure 2.12(b)). Openpose[8]is used to detect pose keypoints from source video. An Image-Image translation technique is used to generate video frames in the target domain conditioned on the detected keypoints. Using unpaired training data, this method of conditioning can be used to generate images of a given person in multiple desired poses unseen in the training data.

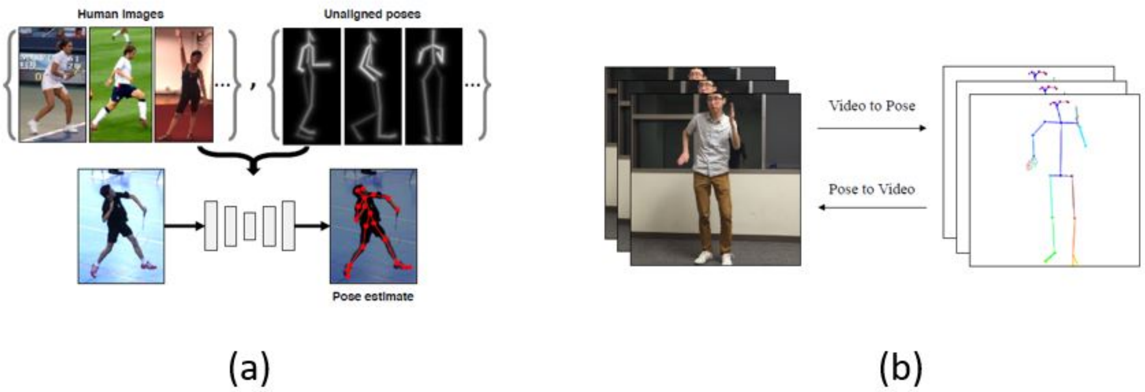


Figure 2.12: CycleGAN applied to translate from the domain of human images to corresponding poses in (a)[22] and (b)[10]

2.3.2. Pose Transfer

In these methods, full scale humans are synthesized conditioned on a given pose. For example, in [2] two inputs-(a)Source image (b) Target pose keypoint are concatenated and fed into the network. The network learns to synthesize a new image of the same person depicting the condition pose while preserving the same background clothing colours, clothing texture and lighting from the source image. The networks achieves this by splitting the network into different modules which perform its own sub-tasks.These sub-tasks are trained jointly using only a single target image as a supervised label. The full architecture is provided in Figure 2.14. What is remarkable in this technique is that even fine details like lights and shadows seen on the source image are preserved in the newly synthesized image as can be seen in Figure 2.13. An adversarial discriminator is used to force the network to learn such realistic details.

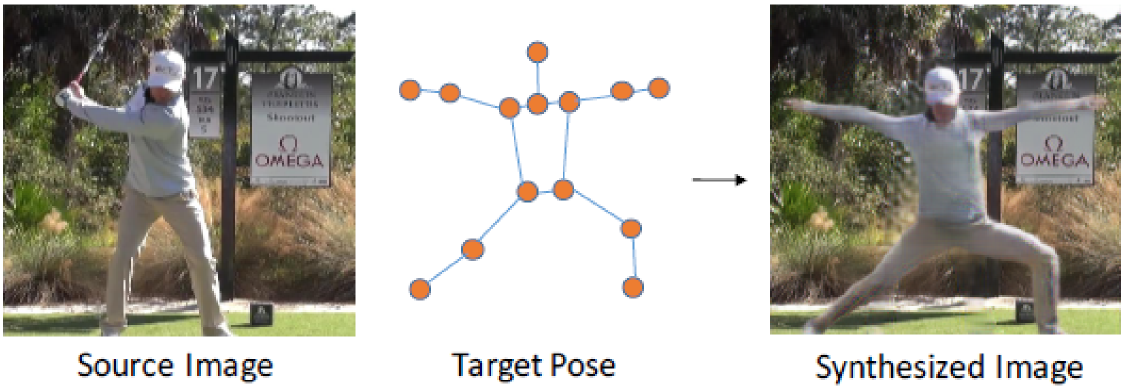


Figure 2.13: Synthesizing a new image of the same person depicting the conditioned pose[2]

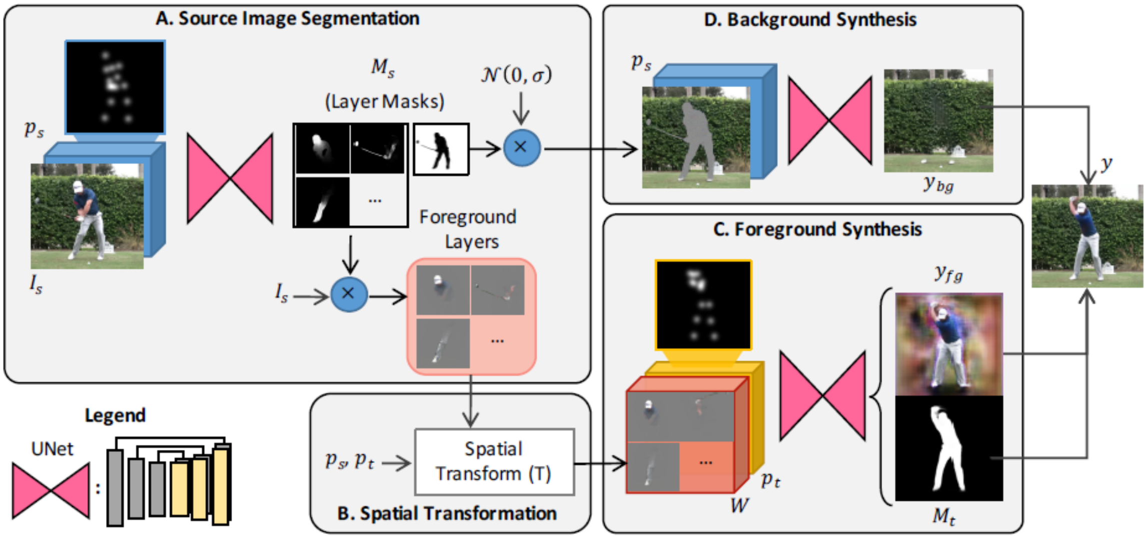


Figure 2.14: Architecture of [2]. First, image segmentation is done by module A on I_s by separating body parts from the background. Module B performs spatial transformation of body parts in I_s. Module C synthesizes the image of the person y_{fg} on the conditioned pose, by fusing the body parts and also outputs a foreground mask M_t in parallel. Module D creates a background image, y_{bg} using hole-filling. Finally, y_{fg} and y_{bg} is combined to produce y.

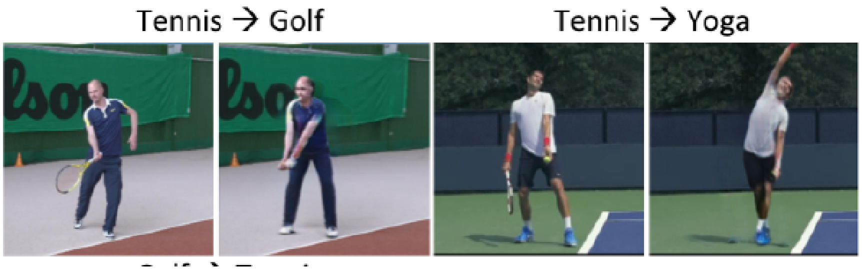


Figure 2.15: The method generalizes satisfactorily even for new poses not seen in the training set. Here, network is trained only on Tennis poses but tested to synthesize humans in Golf/Yoga poses. [2]

As another example, in [28], given a synthetic image silhouette of a projected 3D body model, the network synthesizes new persons with similar pose and shape but in different clothing styles (see Fig. 1). A silhouette of a 3D model of a person is provided as the condition. Figure 2.16 shows sample output images from the network.

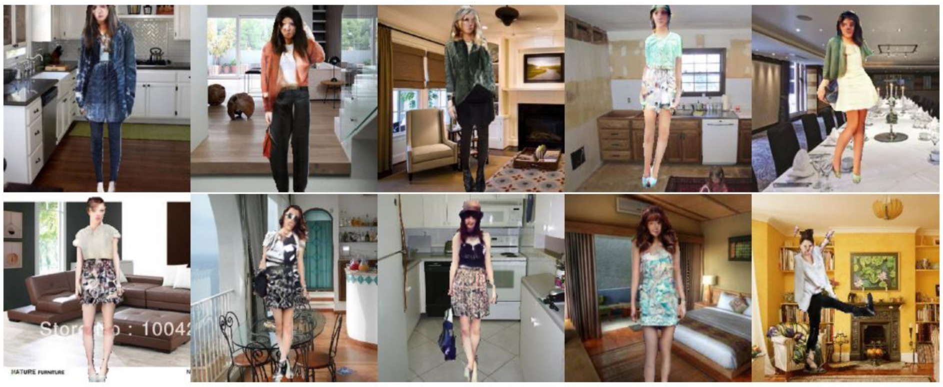


Figure 2.16: Example output from [28] overlapped on indoor backgrounds

- Input: Image of SMPL model
- Output: Multiple human images in different attires condition on input.

2.3.3. Text-Conditioning

In these methods a synthesized image is conditioned on natural language descriptions. In [46], a text description is used as input to synthesize a complete person in the pose and action mentioned. However in this method, the target pose is selected from a bank of standard poses based on the text. The bank of standard poses are basic in nature and do not contain much variability. Figure 2.17 shows some results from this method. One can observe that the input word ”walking forward” has resulted in images of person walking in the same pose. This hinders our original problem of introducing multiple poses. Moreover, these methods require labelled images. It is an exhaustive to generate different labels of colour and pose and therefore proves counter productive to our case.

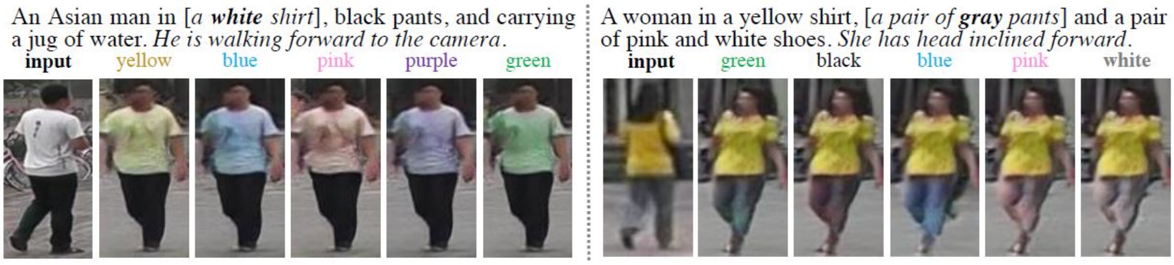

An Asian man in [a white shirt], black pants, and carrying a jug of water. He is walking forward to the camera.
input
yellow
blue
pink
purple
green
A woman in a yellow shirt, [a pair of gray pants] and a pair of pink and white shoes. She has head inclined forward.
input
green
black
blue
pink
white

Figure 2.17: A method for generating human images conditioned on text descriptions[46].

2.3.4. Segmentation based

Another method is to condition images on human parsing segmentation masks to generate new images of people in different clothing/textures. In [43] an image in-painting method is employed to complete missing boxes of pixels from an image. Although it is meant for portrait image editing tasks, the method could be used for generating unique appearances of people by removing clothing information from images. For example, in Figure 2.18 clothing details are removed by boxing it out of the image. The network employs a human parser to obtain body part segmentations from the input. Then, a Generator learns a mapping from random noise to a clothing texture borrowed from the input image domain. The segmentation map is again used to fill in the image with generated texture.

Although these methods provide a method to generate controllable human images in terms of clothing, there is no control on pose. Also it would take a lot of effort and time to create paired datasets of people images and respective body part segmentations. We would also need to provide target segmentations which is also counter-productive to our case.

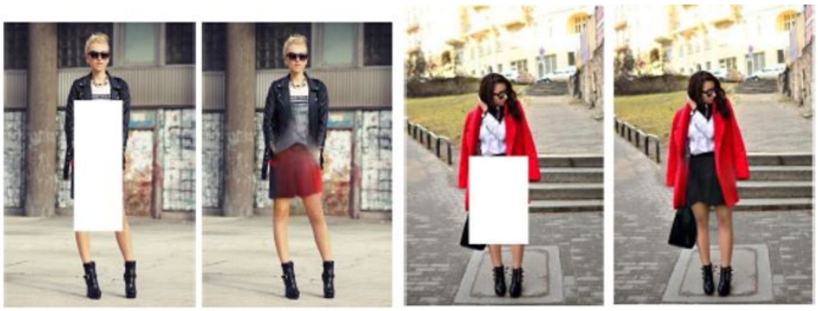


Figure 2.18: GAN network learns to fill blocks of missing pixels to result in new clothing textures[43].

2.3.5. Without conditioning

In [45], a person re-identification dataset is augmented by synthesising all combinations of poses and appearances of people seen in the dataset. However, the network doesn't need an explicit condition vector as an input. The network achieves this by training encoders to learn latent representations of structure(pose, body size, background etc.,) and appearance(clothing, shoes, texture etc.,) separately. The two representations are mixed and fed to Generators which map it back to the image domain borrowing styles from both representations. Figure 2.19 shows example outputs. This allows a user to select the structure and appearance of choice and synthesize N (row) * N (column) new images.

However, in this method no new pose is synthesized. The method ensures that all possible combinations of available poses and appearances are possible. Therefore if the reference data is skewed containing a lot of poses that look similar, the augmented dataset would also contain a similar distribution.

2.3.6. Pose augmentation

In the previous section we saw skeleton like structures called ”keypoints” being used to represent human poses. There are some works in the literature that play around with these pose keypoints with the intention of synthesizing new pose keypoints that are not seen in the original dataset. A couple of those techniques are discussed.

In [24], person re-identification performance is improved by increasing the occurrence of people in diverse poses. First, a Gaussian mixture model(GMM) is used to cluster the ground truth poses into 7 clusters. The 7 cluster centers are considered as unique poses. Each character queried is then conditioned on these 8 poses to synthesize new images of the same character in new poses(Figure 2.20). Such a dataset is then trained on person re-id networks to obtain superior detectors.

In [20], the performance of pose-detection is improved by increasing the occurrences of ”rare” poses in the training set. In the wild human pose datasets have more samples of upright standing poses and less of other types of poses. The authors addresses this by oversampling the minority poses and synthesizing synthetic images in those minority poses and achieve class balance. First, ground truth keypoints from the training set are

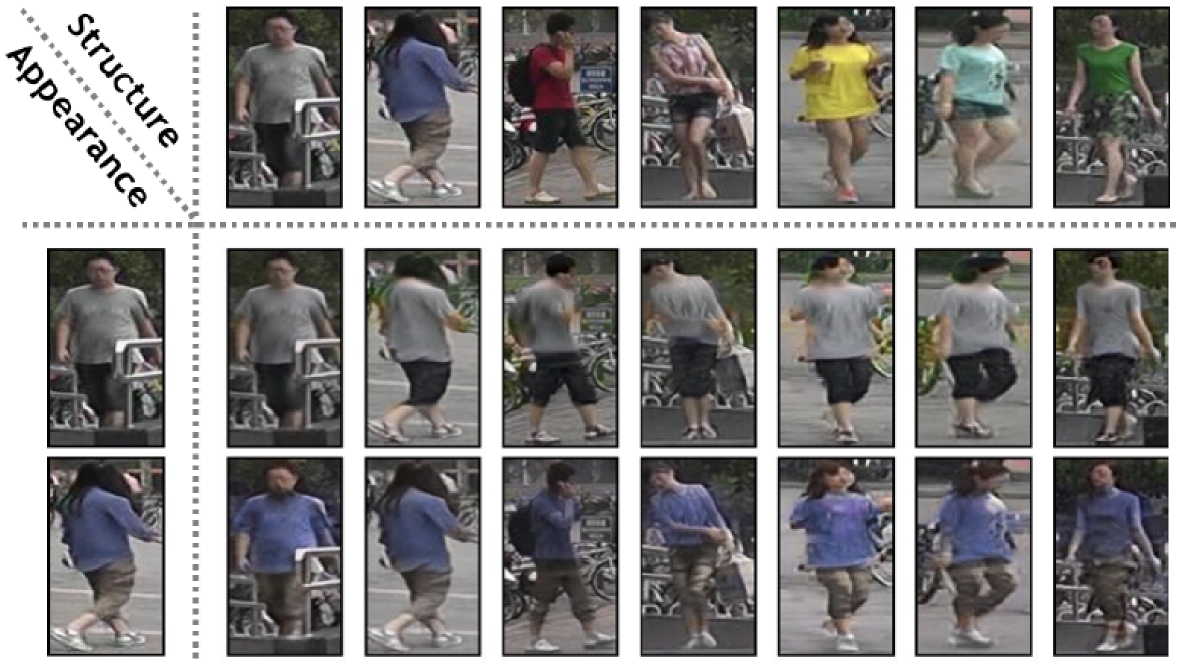


Figure 2.19: Each row and column corresponds to different appearance and structure of people. New images are synthesized by mixing appearance and structure encoding[45].

grouped into clusters using K-Means clustering. Figure 2.21 shows the poses that represent the cluster centers for K=7. The distance between a pose sample and the cluster center(CD), is compared with a pre-defined distance threshold (DT) to determine whether it is a rare sample or not. Figure 2.22 shows the authors's classification of rare and non-rare poses. The filtered poses from this method are then duplicated in the dataset by synthetic people using SMPL mesh[30]. The authors claims an improvement of 13.5 mAP through this method.

These examples show that there is some scope in aiming for better detection performance by enriching pose diversity in training samples. However, these techniques mostly addressed class imbalance by duplicating certain poses in newly synthesized images having new colours and textures(new person same pose). An improvement on this could be to synthesize truly new poses that do not occur in the training set.

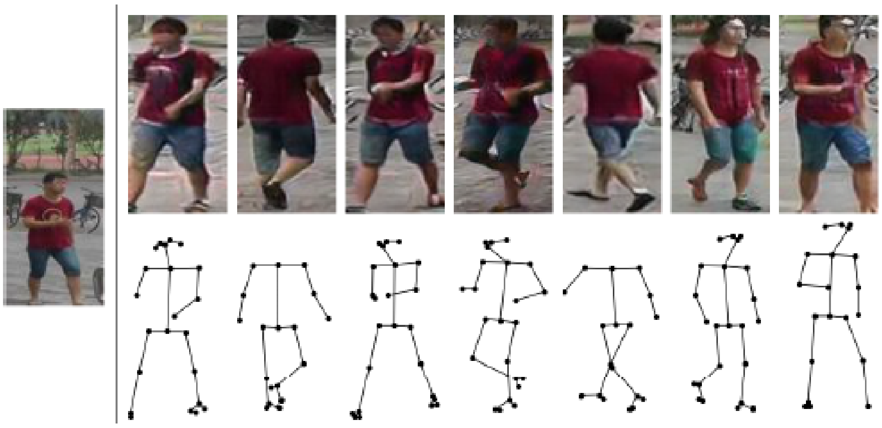


Figure 2.20: The poses represent the resulting 7 cluster centers after GMM clustering on poses extracted from the dataset. Example output when a queried image is conditioned on all 8 poses[24]

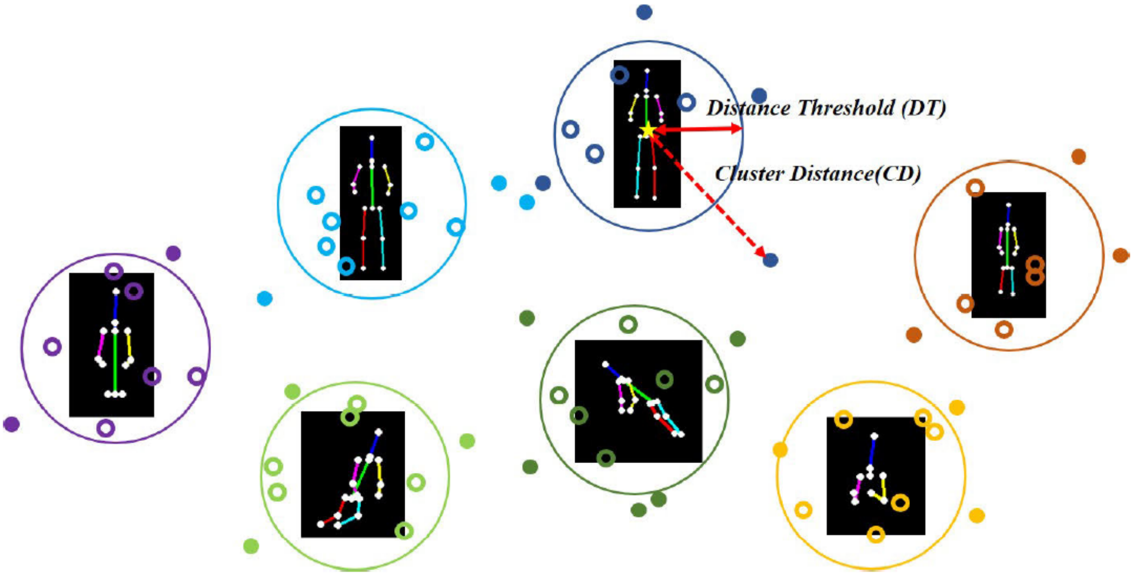


Figure 2.21: Poses represent cluster centers resulting from K-Means clustering. Rarity of a pose is defined as how far a point is w.r.t to the cluster center[20]

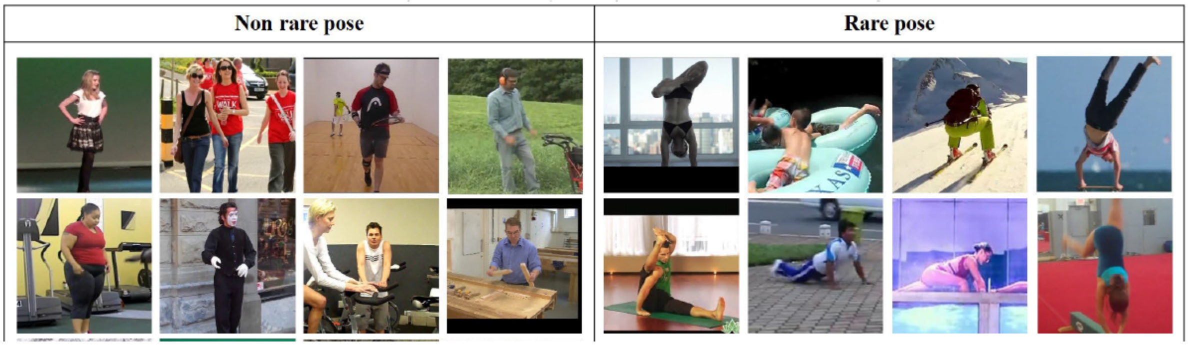


Figure 2.22: Example images of what rare poses and non rare poses look like as defined by[20].None rare poses happen to be mostly upright

2.4. Common Metrics used

In this section we discuss the different types of metrics to evaluate the performance of the methods to be employed.

Comparing Distributions

For high dimensional data, PCA/t-SNE can be applied to obtain a 2-D projection[39] of the high dimensional space. A visual comparison can then be made between the two distributions. Figure 2.23 shows visualization of images from different GAN datasets using t-SNE method. In the example provided by [15], the fake images generated by CycleGAN are distributed uniformly(measure of diversity) and merge relatively well with the distribution of real images(measure of realism/sanity). This is not the case when compared with fake images generated by StyleGAN as the two distributions are linearly separable.

Cosine Similarity

By using the cosine similarity the distance from one vector to another can be computed without being corrupted by the varying length of the vectors. To compare human poses with one another, a metric commonly used in pose-detection networks is cosine similarity. Cosine similarity measures the likeliness between two vectors by measuring the cosine of the angle between them. The metric is ideal because the value is not affected by the magnitude of the vectors(size/scale of pose).Suppose we have vectors A and B then the cosine similarity equation is as follows:

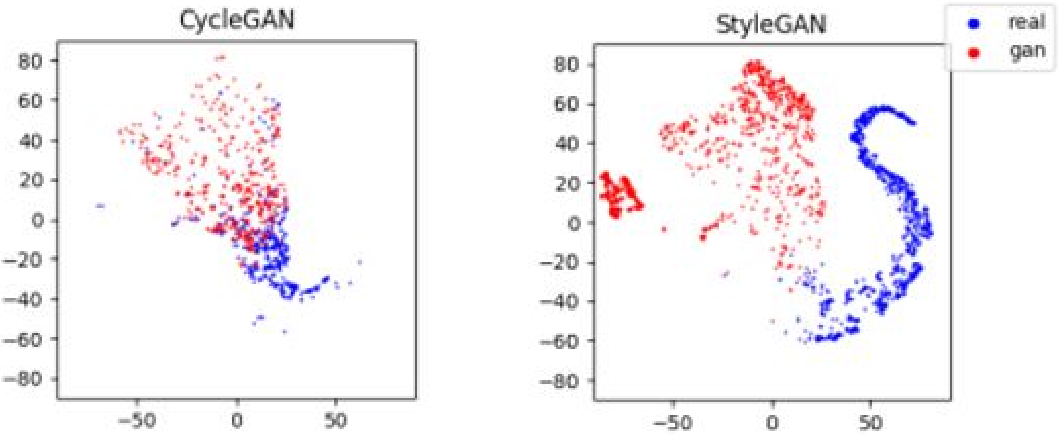


Figure 2.23: t-SNE visualization used to compare perceptual quality of fake samples generated by CycleGAN and StyleGAN with respect to real samples[15]. In CycleGAN there is some overlap in the visualization which means the fake images are pretty close to the real ones. In StyleGAN the distribution is linearly separable which means the fake images are distinguishable from the real ones.

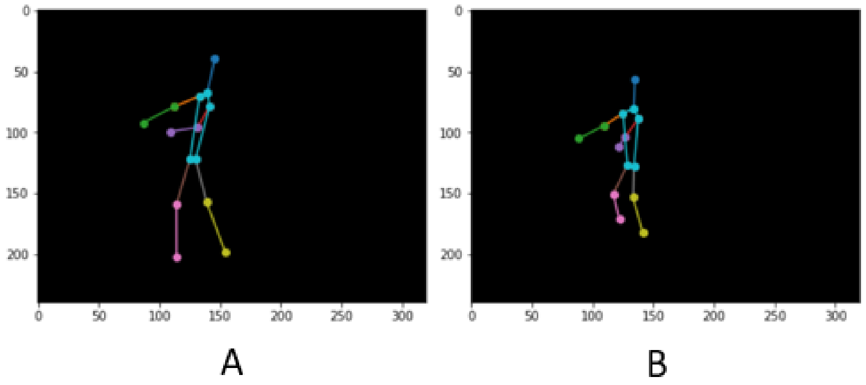


Figure 2.24

$$\text{Cosine Similarity } = S_C(A,B) = \cos(\theta) = \frac{\mathbf{A}\cdot\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} \tag{2.1}$$

2.4.1. Person detection evaluation metrics

The end goal of the this work is to improve person detection performance of a detector like YOLO [6] by training on the enriched dataset. Some common evaluation metrics used for object detection performance are mentioned.

Precision

Precision gives information regarding the proportion of correct predictions(TP) vs all predictions that the model reported as positive(TP + FP).

$$\text{Precision } = \frac{TP}{TP+FP} \tag{2.2}$$

Recall

Recall gives information about how many correct predictions(TP) the model picked up out of all the true occurrences in ground truth(TP + FN).

$$\text{Recall } = \frac{TP}{TP+FN} \tag{2.3}$$

Average Precision(AP)

There is generally a trade off between Precision and Recall depending on the model's sensitivity. Therefore Average Precision is a superior metric because it takes into account both Precision and Recall. It can be considered as the area under the interpolated curve from a Precision vs Recall plot(Figure 2.25).

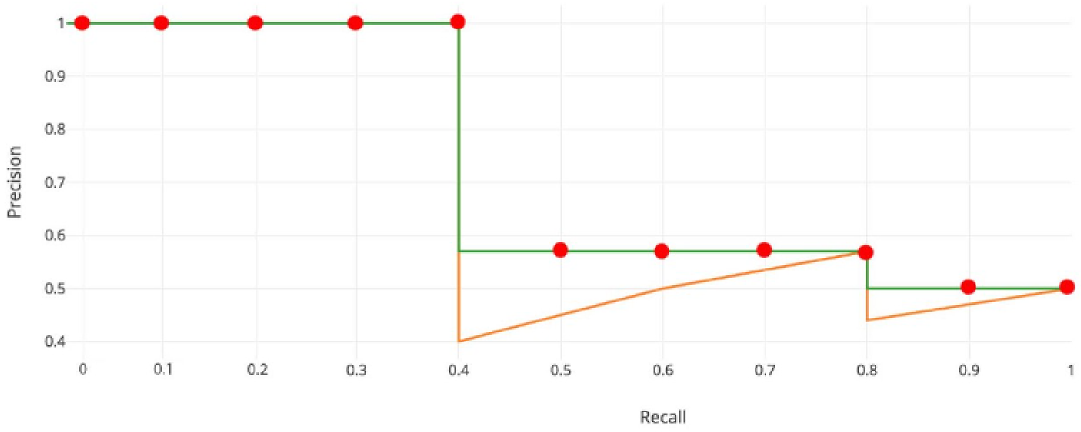


Figure 2.25: Average precision is the area under the interpolated Precision vs Recall curve

2.5. Conclusion and research gaps

The first section introduced the problem statement for this work. There is a necessity for augmenting synthetic datasets that are generated using 3d softwares for their lack of diversity. A possible direction is to use GANs as a method to extract more information from a given dataset and enhance its diversity. In the second section we saw examples of synthetic data being effectively used in training for object detection tasks. A comparison of available popular synthetic human datasets was made. In the third section we looked at some of the works which studied the effectiveness of GANs as a data augmentation technique for popular computer vision tasks like classification, segmentation and object detection. Finally in the last section we looked at targeted human image synthesis techniques using image conditioning followed by some of the commonly used metrics used for our case.Here are the observations and research gaps found from the literature study:

- Networks pre-trained on Synthetic data and fine-tuned on small real data is shown to outperform pure real data training. Training on 100% synthetic data(no fine-tuning) and testing on real data is also shown to perform reasonably well for multiple vision tasks. There are some interesting techniques of increasing training annotations by pasting synthetic/real people patches on real backgrounds. This direction can be explored for augmenting our person detection training set.
- GANs can provide an effective way to fill in gaps in the discrete training data distribution and augment sources of variance which are difficult to augment in other ways. However, variance will not extend to points beyond the extremes of the training data[7].
- Performance of GANs depend on the number of training samples. The quality and diversity of GAN generated images depend on the number of samples and diversity in the training data used. This becomes a chicken and egg problem.

- There is scope for enriching diversity in terms of human poses seen in datasets. Papers that attempted this, approached it as a class balancing problem by oversampling the minority poses[20]. Another, changed appearances of people depicting these poses [24]. A possible direction is to synthesize new, unseen poses from available poses. Full scale humans can then be synthesized in those poses using Pose-Warp GAN[2].
- Photo realism of synthetic data is not as important important as diversity of data[33].

2.5.1. Research Questions

From the observations made above, the following main research question and sub questions can be formulated: *Can a synthetic person detection dataset be enriched with new characters in unseen poses using a GAN?*

Q. 1 How to synthesized new, diverse poses from poses present in the original dataset?

Q. 2 How to measure the diversity of poses in a given dataset?

Q. 3 Does increase in diversity of poses seen in the augmented dataset translate to improved person detection performance?

3 Methodology

In the following chapters, all mentions of the word ”**source**” refers to the dataset of synthetic human-in-the-scene images that is lacking in diversity. The goal of this work is to enrich/augment this limited dataset in terms of the variety of human poses seen in the dataset. Accordingly, all mentions of the word ”**augmented**” refers to the enriched/augmented dataset obtained through the proposed method in this work.

This chapter explains the step by step approach to answering the main research question and its sub-questions arising from the previous chapter. Figure 3.1 shows the entire design pipeline and its evaluation. Section 3.1 (Step 1) explains source data preparation. In Section 3.2 (Step 2), a method of augmenting source poses is detailed. In particular, Section 3.2.1 introduces pose sampling from the source poses distribution using Gaussian sampling. In Section 3.2.2 the sampled poses are augmented using Interpolation as a technique. Section 3.2.3 introduces a metric to compare pose data distributions and proposes Least Neighbours Filtering to obtain a diverse set of human poses. Section 3.3 (Step 3) deals with synthesis of human characters depicting poses obtained from Section 3.2. In particular in Section 3.3.1, Pose Warp GAN[2] takes pairs of source image and conditioned pose and generates a new human character. In Section 3.3.2, the characters are pasted on open source real backgrounds to obtain the augmented dataset. Finally Section 3.4 (Step 4) performs person detection performance evaluation and comparisons between source and augmented datasets. In Section 3.4.1, YOLOV3 object detection network is trained on the source and augmented datasets. It is later fine-tuned and evaluated on real images in Section 3.4.2 and 3.4.3 respectively.

3.1. Step 1: source Data Preparation

The SURREAL synthetic dataset[42] consists of 60000 video files each containing 100 frames. It also comes with accompanying ground truth pose, depth maps, and segmentation masks for the corresponding frame numbers. We choose this dataset as our source dataset as it already consists of considerable diversity. We want to improve on what is already possible using 3D tools. Poses in the dataset are sourced from the CMU MoCap database[9]. CMU MoCap contains more than 2000 sequences of 23 high-level action categories. For this work, only RGB images and corresponding pose keypoint files are required as inputs for further steps. Figure A.5 shows some example images from the SURREAL dataset .

For investigating the effect of source dataset size, 3 datasets consisting of 6000, 12000, 25000 images respectively are extracted from the video files. Videos are picked randomly from the dataset and 4 frames are selected from each video having constant frame number difference. It is convenient that the dataset already contains ground truth pose keypoint files. Figure 3.2 (a) shows an example image overlapped with keypoints from the SURREAL dataset. Pose keypoint files are in the format of [24X2] matrices corresponding to 24 body joints. Each value represents the x-y pixel coordinate of the corresponding body joint. Figure 3.2 (b) shows the order of indices considered in the SURREAL keypoint files. However the input fed to the Pose-Warp GAN in Step 3 takes the pose keypoints file in the format of [14X2] matrices as shown in 3.2 (c). Accordingly, a processing step performs this reordering. Additionally an extra joint is added at the top of the head which is missing in

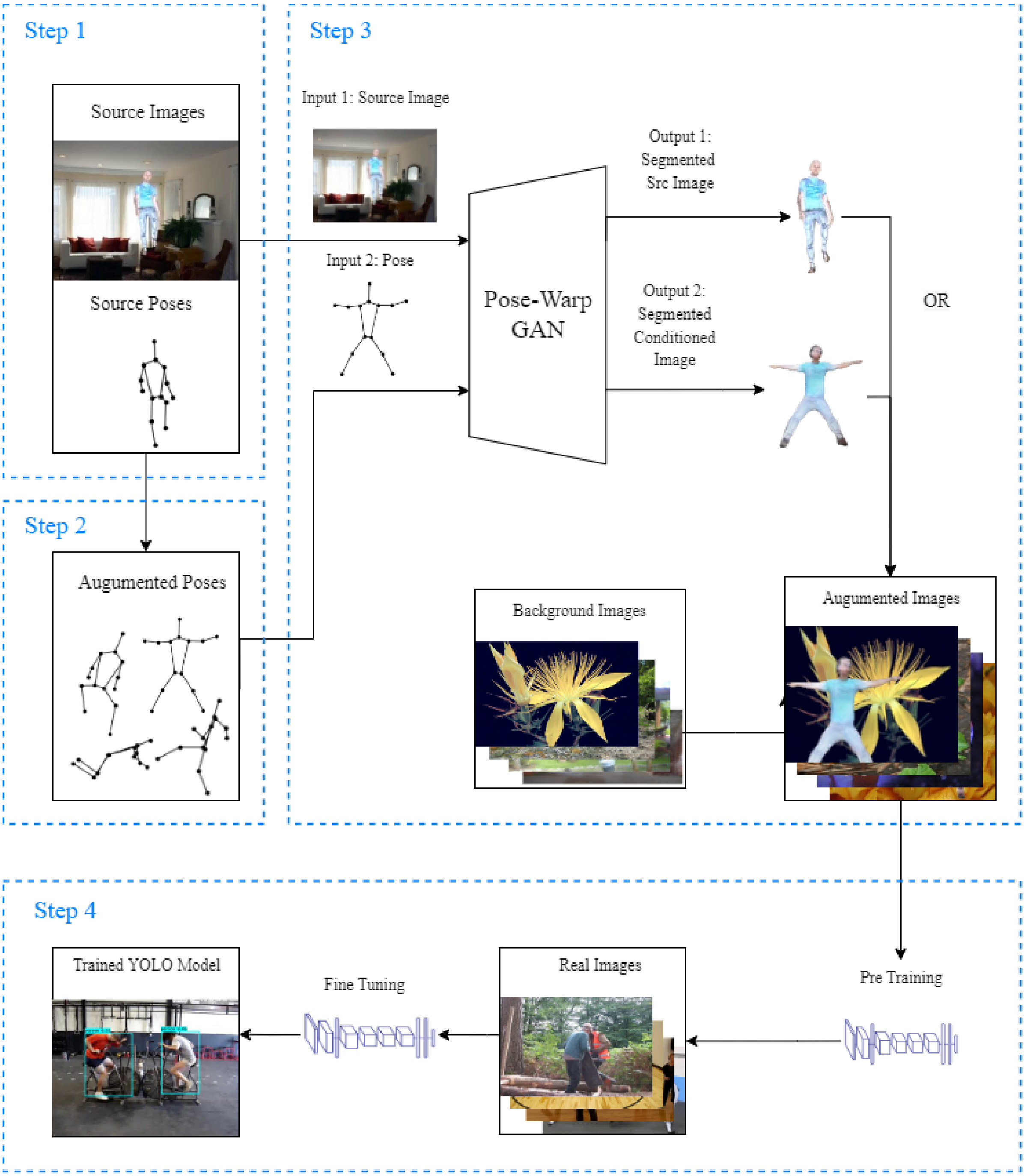


Figure 3.1: The complete pipeline showing all the Steps followed. In Step 1 pose keypoint files are extracted from source images. In Step 2 source poses are augmented and filtered using Least Neighbours method. In Step 3 pairs of Image and required pose are fed into Pose-Warp GAN to generate new human characters. These are pasted on background images sourced from [27] to create the augmented dataset. Step 4 involves training YOLOv3[36] on the augmented dataset and fine-tuning on images from [1] to give a trained person detector.

SURREAL keypoints. This extra joint is extrapolated from the slope of line connecting joints 13-15 in figure 3.2 (b). Resulting overlapped figure is shown in figure 3.2 (d). The head joint is approximated in the following manner:

$$\text{Joint } 0_y = \text{Joint } 0_y + 2.5 * (\text{Joint } 0_y - \text{Joint } 1_y) \tag{3.1}$$

$$\text{Joint } 0_x = \text{Joint } 0_x + 2.5 * (\text{Joint } 0_x - \text{Joint } 1_x) \tag{3.2}$$

In case another dataset is used, Openpose[8] can be used to extract corresponding human pose keypoints from RGB images. Accordingly, this should be followed by an indices reordering step to make it compatible with inputs for Step3.

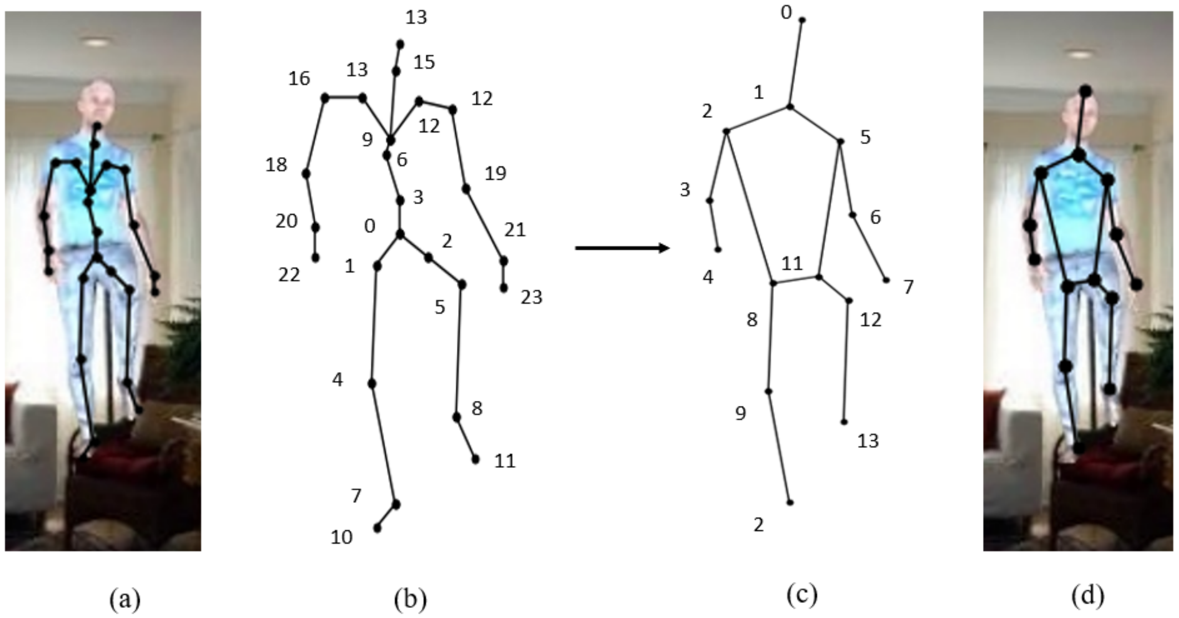


Figure 3.2: (a) and (b) show pose keypoints style and joints ordering in SURREAL[42] format. (c) and (d) shows the format acceptable by Pose-Warp GAN[2]. Head joint 0 position in (d) is approximated by extrapolating line connecting joints 13,15 in (b)

3.2. Step 2: Pose Augmentation

Poses obtained from Step 1 is taken as input for this Step. The [14X2] matrix is flattened to a [28x1] vector. To visualise the pose data, a Principal Component Analysis is performed. Figure 3.3 shows the PCA projection of the source poses on a 2D space by taking the first two components contributing to maximum variance. Although the cumulative explained variance is 55% on a 2D space, the figure reveals the clear issue with the source poses. A majority of the points are concentrated densely in one part of the space while other parts are sparsely populated. There are also pockets of space between point clusters that are empty. An ideal distribution would have the points spread evenly covering the whole space.

3.2.1. Interpolation

This section addresses the following research questions:

Q 1. ”How to synthesized new, diverse poses from poses present in the original dataset?”

An approach is to use the minority oversampling technique SMOTE[11]. In this method, new points are synthesized by interpolating between neighbours of a point sampled from the minority class as shown in figure A.2. The technique requires class labels for each point. Interpolating between neighbours ensures that the synthetically generated point is not an exact replica of an existing data point, while also ensuring that it is not too dissimilar from known observations in the minority class. Although this is a viable option for our case, it has certain disadvantages. There are many ways of clustering the poses point cloud and therefore what pockets constitute ”minority class” is ambiguous. Also, interpolated new points between points that are already close together, as in the case with our point cloud, do not add to diversity. The problem to solve is not just of class imbalance, but also to obtain significantly different poses.

As an alternative, in this work new poses are synthesized by interpolating between pairs of points selected randomly from the poses dataset. Additionally Poses are sampled from a Gaussian Mixtures Model that is

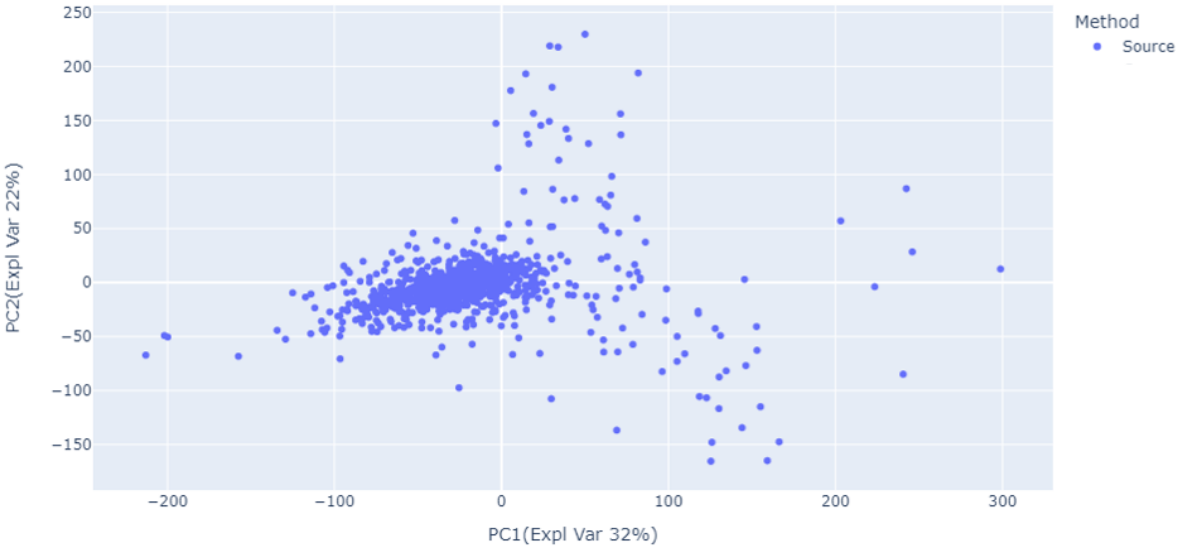


Figure 3.3: PCA projection of source poses on 2D space. The distribution shows a densely populated cluster of points in one region while other region are empty or sparsely populated. This says that majority of poses in the source are similar to each other.

trained on the source Poses. A GMM is a generative, probabilistic model that assumes all the data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters. Under the hood it performs a generalizing k-means clustering to incorporate information about the covariance structure of the data as well as the centers of the latent Gaussians. An advantage of using a GMM is that it already generates poses that are not exact duplicates of the source Poses but rather approximates from the learnt distribution. Also, GMM allows Soft clustering and Captures non-spherical cluster distributions while being computationally efficient. Most importantly, GMM allows sampling upto 10^6 points at a time. This makes a large combinations of interpolations possible.

Accordingly a GMM is used to learn the distribution of source Poses. The optimal value for hyperparameter k(number of clusters), is decided by minimizing the Akaike Information Criterion(AIC).AIC is an estimator of prediction error. It estimates the relative amount of information lost by a given model: the less information a model loses, the higher the quality of that model. This provides a means of correcting for over-fitting by adjusting the model likelihoods. Figure A.1 shows the plot AIC vs Number of clusters for 1000 poses. The minima is at 100 clusters which is set as the value for k.

Next, the learnt model is sampled for 10^6 points separately twice to create two sets P_1 and P_2 . A new set P_I is created by interpolating between random pairs of points from the two sets. Figure 3.4 shows an example of an interpolated pose between two poses. The volume of coverage of the interpolated point cloud P_I should clearly be lesser than P_1 and P_2. To mitigate this, source poses are concatenated with P_I.

3.2.2. Least Neighbours Filtering

This section addresses the following research question:

Q 2. "How to measure the diversity of poses in a given dataset?"

Although the resulting set from the previous section contains new points, it will still contain pockets that are densely populated. This work proposes the Least Neighbours Filtering to create a diverse pose dataset. This technique aims to weed out those poses which are mostly similar to each other while preserving dissimilar poses. This is done by employing the Cosine Similarity metric discussed in Section 2.4.2. The metric outputs a scalar value in the range [0,1]. A value close to 0 means the pair of poses are similar to each other. A value close to 1 means they are dissimilar. Let the number of poses in P_I be denoted by N. Let p_i and p_j denote the i^{th} and j^{th} pose respectively in P_I, where $i \in \{0, N\}$ and $j \in \{0, N\}$. Then the Similarity Score S_{ij} between any two poses p_i and p_j is given by:

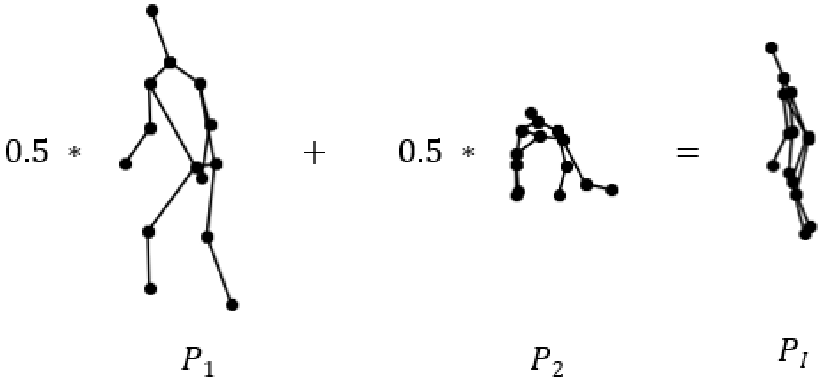


Figure 3.4: Example pair of Poses from two different Gaussian sampled sets P_1 and P_2 interpolated to give an intermediate Pose belonging to set P_I.

$$\mathrm{S}_{ij} = \left|1 - \frac{p_i \cdot p_j}{\|p_i\|\|p_j\|}\right| \tag{3.3}$$

A threshold t decides if the poses are ”neighbours”. The outcome is stored in bool$_{ij}$. This is given by:

$$\mathrm{bool}_{ij} = \begin{cases} 0 & S_{ij} \leq t \\ 1 & S_{ij} > t \end{cases} , where \quad t \in (0,1) \tag{3.4}$$

This value is summed along one of the axis to obtain $neighbours_i$ which gives the count of neighbours for any point p_i in the point cloud P_I. This is given by:

$$\mathrm{neighbours}_i = \sum_{j=0}^{N} \mathrm{bool}_{ij} \tag{3.5}$$

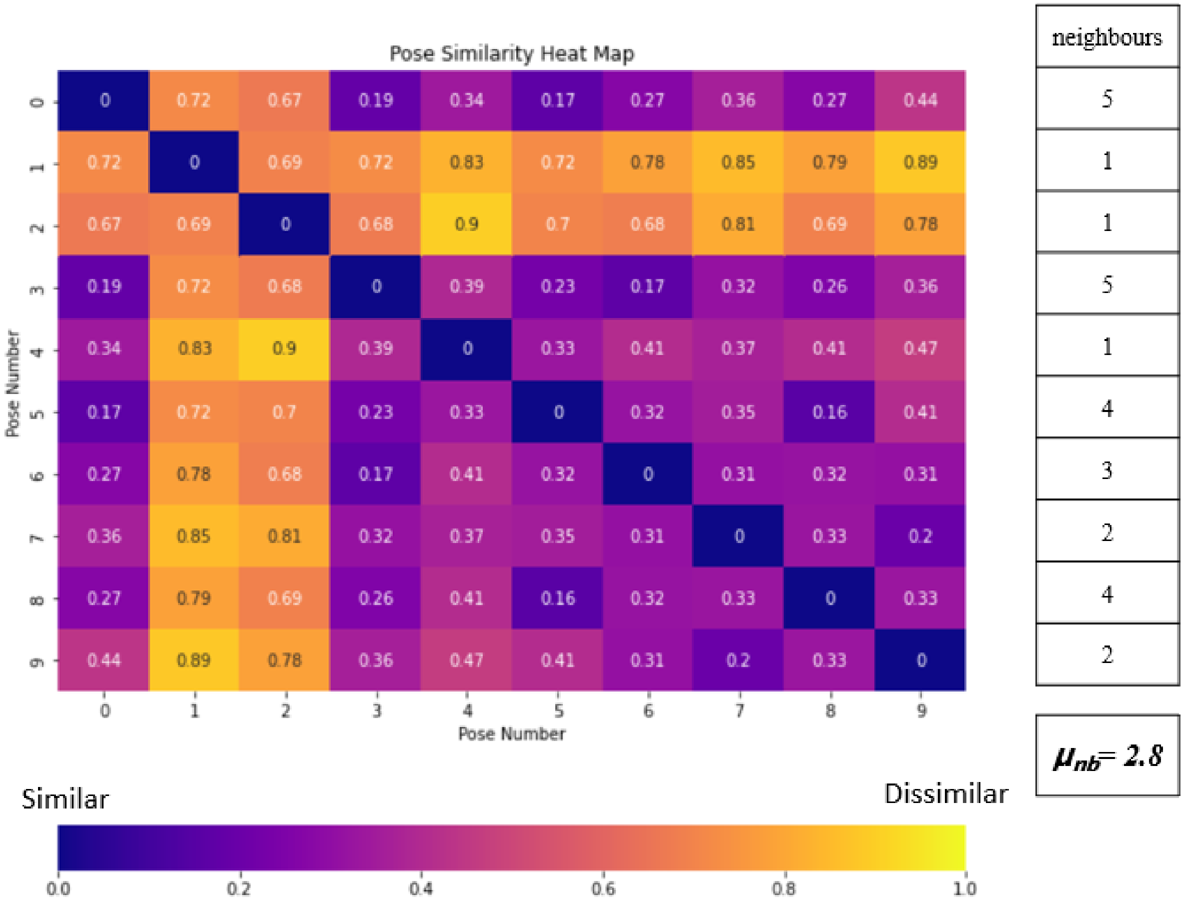
This results in the list $neighbours$ whose elements corresponds to the count of neighbours for every point in the point cloud P_I. $P_I$is then indexed by the sorted indices of list $neighbours$. From this, the first n points are picked which gives a subset P$_{IF}$. This resulting set should consist of points that have the least neighbours on average comparatively. This is given by:

$$\mathrm{P}_{IF} = \mathrm{P}_I^{(n)} = \mathrm{P}_I[\mathrm{argsort}(neighbours)][:n] \tag{3.6}$$

To illustrate with an example refer to figure 3.5. The figure shows a heat map of matrix of Similarity Scores between the first 10 poses from P_I. For example, the Similarity Score S_{01} between poses p_0 and p_1 is given by the value 0.72 at the the intersection of o^{th} row and 1^{st} column. All the diagonal elements are zero as it compares similarities of every pose with itself. The matrix is symmetric. The list on the right shows the count of neighbours for each point in the point cloud considered. For example, Pose Number 0 (p_0) has 2 neighbours. This is calculated by counting the number of values in the 0^{th} row of the matrix that exceed threshold t=0.3. The mean of list $neighbours$, μ_{nb} gives the mean count of neighbours for the point cloud considered. In this example, on average a point has 2.8 neighbours. The mathematical expression for μ_{nb} is given by:

$$\mu_{nb} = \frac{1}{N} \sum_{i=0}^{N} \sum_{j=0}^{N} \left[\left|1 - \frac{p_i \cdot p_j}{\|p_i\|\|p_j\|}\right| \leq t \right] \tag{3.7}$$

The metric μ_{nb} provides a method of comparing diversity in Pose data distributions. Consider two sets having the same number of points. Then, the set having the lower μ_{nb} consists of points that have less neighbours on average. If a distribution has to accommodate the same number of points at a lower μ_{nb}, then the distribution needs to be more spread out. Therefore, it can be inferred that the distribution having the least μ_{nb} has the most diverse samples. It is to be noted that μ_{nb} depends on the number of points in the dataset and the



	0	1	2	3	4	5	6	7	8	9	neighbours
0	0	0.72	0.67	0.19	0.34	0.17	0.27	0.36	0.27	0.44	5
1	0.72	0	0.69	0.72	0.83	0.72	0.78	0.85	0.79	0.89	1
2	0.67	0.69	0	0.68	0.9	0.7	0.68	0.81	0.69	0.78	1
3	0.19	0.72	0.68	0	0.39	0.23	0.17	0.32	0.26	0.36	5
4	0.34	0.83	0.9	0.39	0	0.33	0.41	0.37	0.41	0.47	1
5	0.17	0.72	0.7	0.23	0.33	0	0.32	0.35	0.16	0.41	4
6	0.27	0.78	0.68	0.17	0.41	0.32	0	0.31	0.32	0.31	3
7	0.36	0.85	0.81	0.32	0.37	0.35	0.31	0	0.33	0.2	2
8	0.27	0.79	0.69	0.26	0.41	0.16	0.32	0.33	0	0.33	4
9	0.44	0.89	0.78	0.36	0.47	0.41	0.31	0.2	0.33	0	2

$\mu_{nb}= 2.8$

Figure 3.5: Heat map of Pose Similarity Scores for first 10 poses from P_I. Pose Similarity Score between any two poses is given by checking for the value along matching row, column number. All diagonal elements are 0 as it compares Pose Similarity with itself. Column on the right shows the count of neighbours for each pose. For example Pose 0 and Pose 5 have 5 neighbours each. In this example, a pose is considered as a neighbour if their Pose Similarity Score falls below threshold $t = 0.3$. μ_{nb} gives the mean count of neighbours. Here, on average a pose has 2.8 neighbours.

threshold cutoff t considered.The reasoning is substantiated further visually in the results Section 4.1.

Figure 3.6 shows all pose pairs making up the first 5 elements of 0^{th} row from the matrix in figure 3.5. The Pose on the left(Pose 0) is maintained the same for all sub-figures. Pose 0 is upright, standing which represents majority of poses seen in source dataset. Figures 3.5(a),(b) shows comparisons with poses like kneeling and sitting which return Similarity Scores closer to 1. Figures 3.5(c),(d) shows comparisons with upright poses having variations in gait and body orientation. These return proportional Similarity Scores closer to 0. The difference in scale of the poses bears no influence on their Similarity Scores.

Figure 3.7(a),(b) shows the distance between these same pairs on 2D and 3D PCA projected spaces respectively. In Figure 3.7(a) it can be observed that all distances between point pairs are proportional to their respective Similarity Scores except the pair P0-P4. This is because the 2D projection accounts for only 55% of the total variance in the dataset. This is verified by the 3D projection in figure 3.7(b), where all lengths are proportional to their respective Similarity Scores.

To conclude, filtering using Equation 3.6 results in the dataset P_{IF}. This augmented dataset is more diverse and uniform in comparison to the source dataset P.

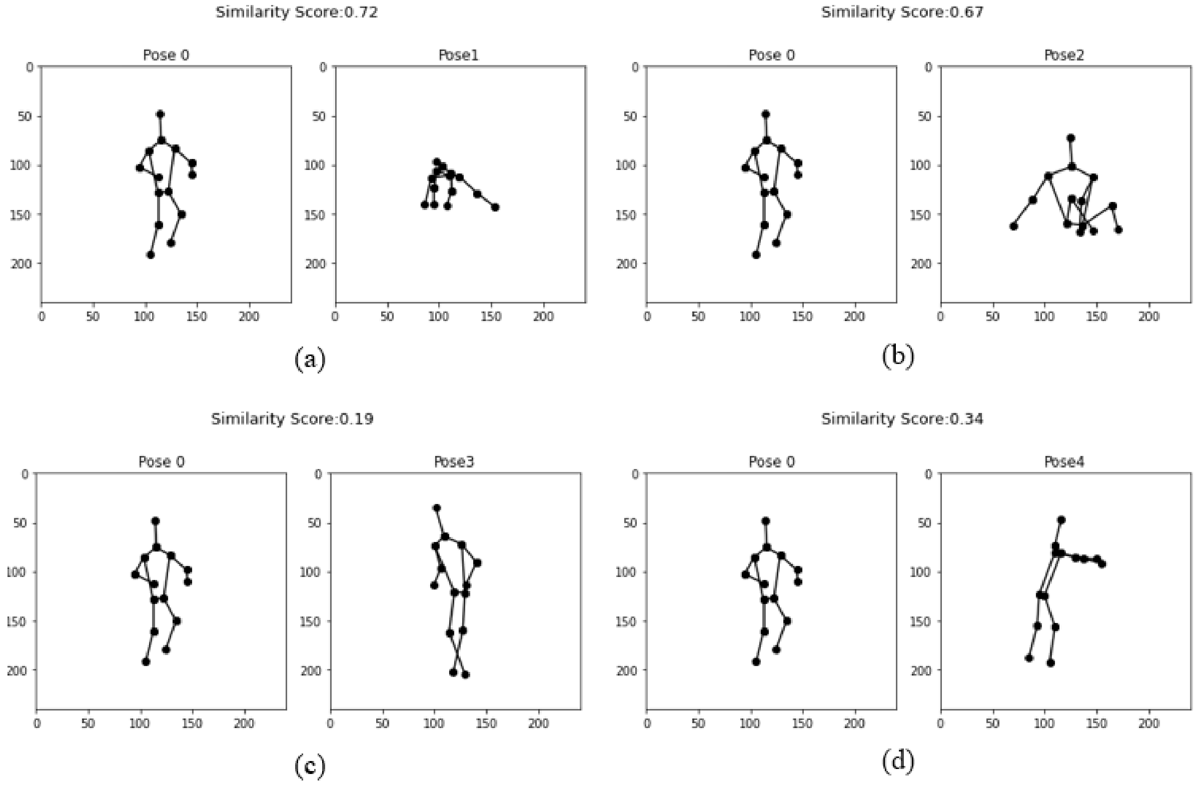


Figure 3.6: Example pairs of poses from P_{IF}. Pose on the left is maintained same for all pairs for comparison sake. Upright poses similar to Pose 0, like in (c) (d) return Similarity Scores closer to 0. Poses significantly different such as squatting and kneeling (a) (b), return Similarity Scores closer to 1.

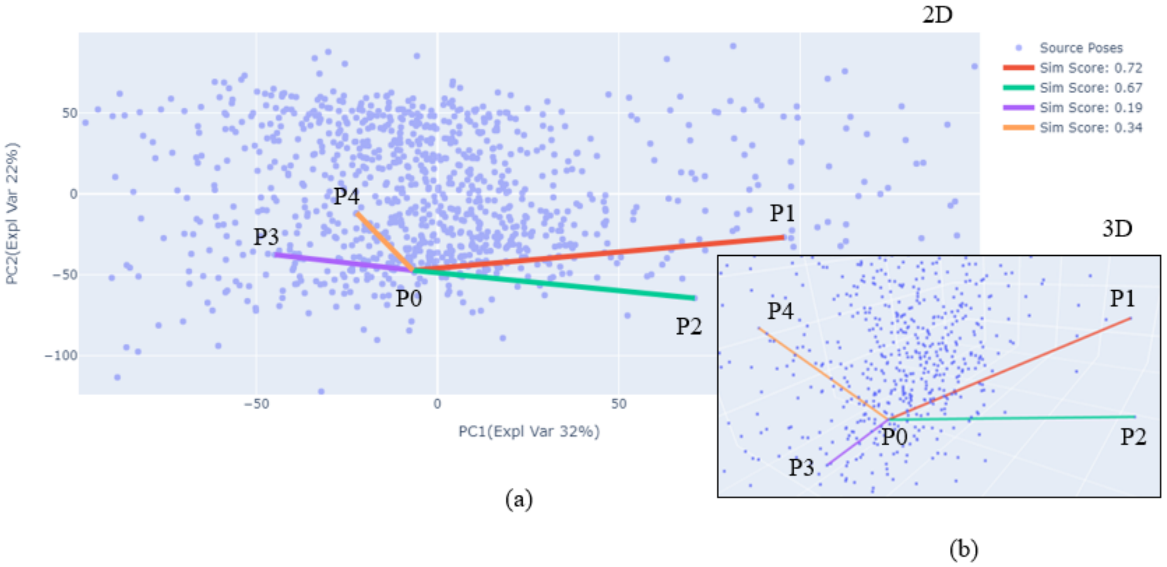


Figure 3.7: Point pairs from figure 3.6 highlighted on PCA projected 2D and 3D spaces. In the 2D projected space (a), distances between all pairs are proportional to their Similarity Scores except pair P0-P4. This is because the cumulative explained variance on 2D is just 55% of the actual. On the 3D projected space(b) all distances are proportional.

3.3. Step 3: Synthesizing New Images

In this section Pose-Warp GAN discussed in Section 2.3.2 is used to generate human characters in new poses.

3.3.1. Image conditioning

Refer Step 3 in Figure 3.1 for the full pipeline. Initially the network is trained on our source dataset instead of using the trained weights provided by the authors of the paper directly. This is done because the model released by the authors was trained on a dataset of limited action classes. The source images and poses are split into two parts to create ground truth labels for training. The network takes pairs of RGB image, and conditioning pose as Inputs which are picked at random. The task of the Generator network is to synthesize a new image showing the input character in the conditioned pose. The Discriminator network compares the output of the generator with the ground truth image. The training loop adjusts the weights of both the Discriminator and Generator for two losses— 1) L_1 loss that corrects for high level structure (pose), 2) $L_{VGG+GAN}$ loss that corrects for finer details and realism. The model is trained for 50000 iterations at a learning rate $1e^{-5}$ for 48 hrs using 1 P100 GPU.

Once trained, only the Generator is used during test time. The augmented Poses dataset P_{IF} from the previous section serves as the input in this step. Refer Step3 in figure 3.1. Poses are picked at random from P_{IF} and paired with random source images. The network gives two outputs— 1) Segmented human character from the source image, 2)Segmented human character synthesized by the generator depicting the conditioned pose. Output 2 is the important one as this constitutes our augmented characters. An example output is provided in figure 3.8 which shows the desired intention of this work clearly. A character from the source is conditioned on 4 different poses taken from the augmented Poses set P_{IF}. The network outputs 4 characters that takes appearance cues from the source image(cloth type, color), and overall structure(pose, scale, orientation) from the conditioned pose. Such control allows the synthesis of an unlimited number of human characters in the desired pose.

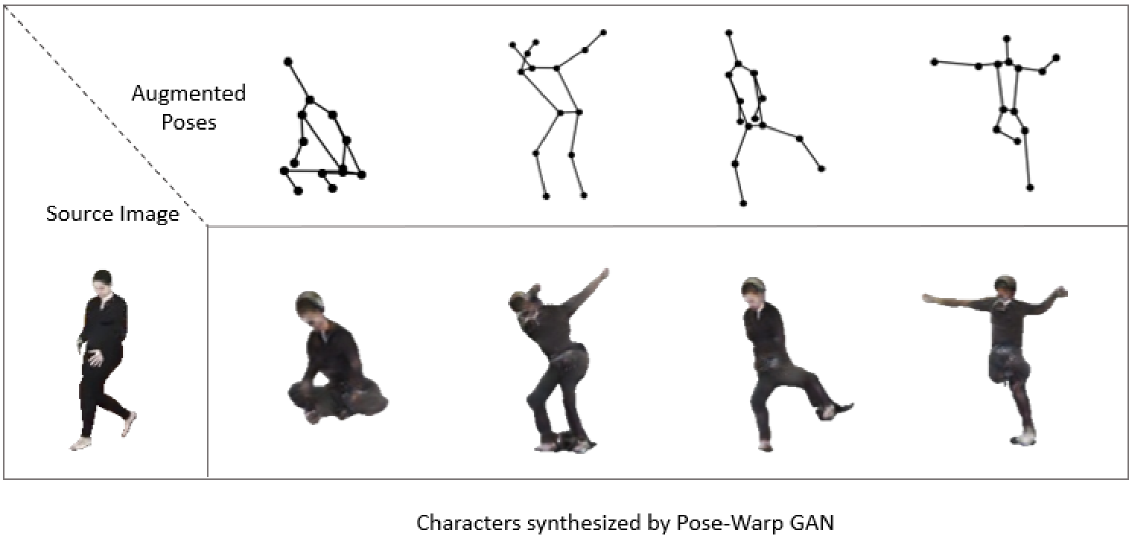


Figure 3.8: Example outputs from Pose-Warp GAN in Step3. A character in source image is augmented by conditioning on different poses to generate multiple characters in desired poses.

3.3.2. Background addition

The characters are then pasted alternatively on images sourced from Google Open Images [27]. From the repository, only those images that do not contain any person annotations are downloaded so as to not corrupt our training dataset. Originally, the background in the source images were sourced from LSUN bedrooms dataset by the authors. When visually inspected it was observed that they lacked in diversity. Pasting the characters on backgrounds that include better variations sets up the trained model to perform better. A similar approach has been followed in [20]. Figure 3.10 shows some example images from the augmented image dataset.

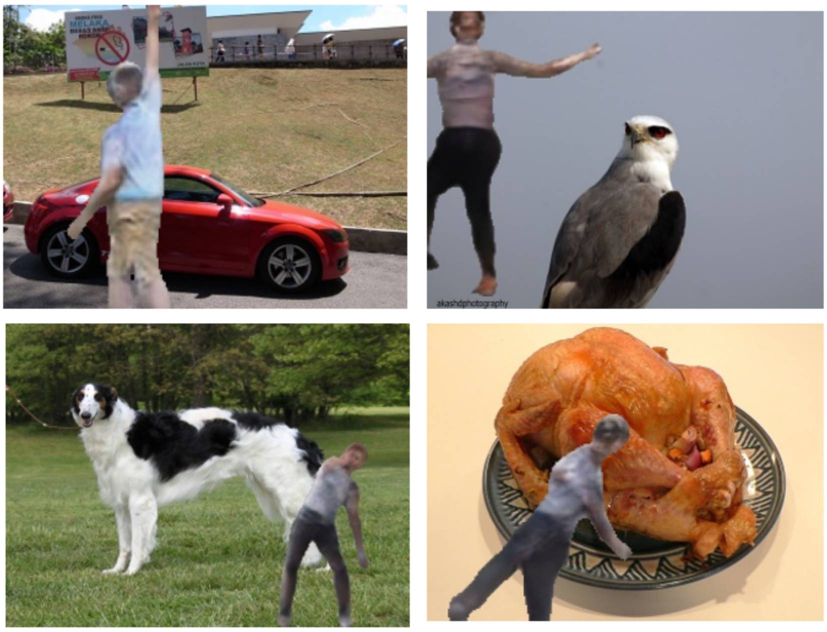


Figure 3.9: Example images from the augmented dataset.

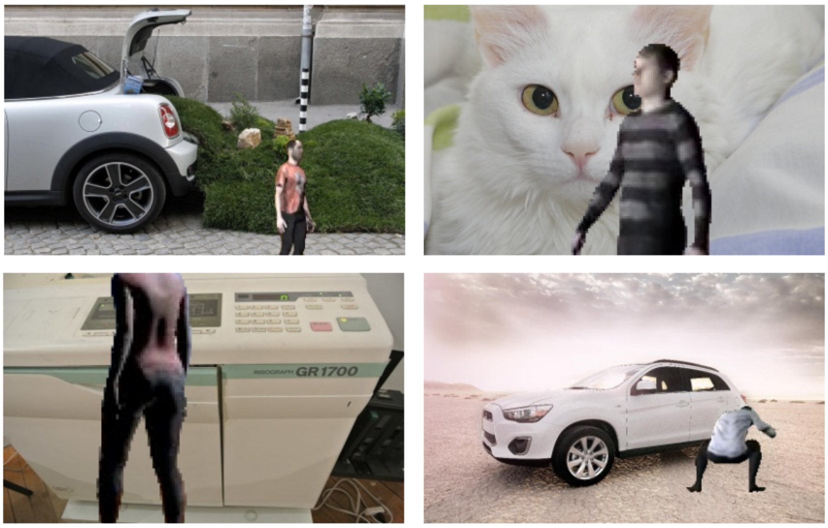


Figure 3.10: Example images from the source data after background addition.

3.4. Step 4: Training and Evaluation

This section involves evaluation of the methods proposed in this work. It aims to answer the main research question:

Q 3. Does increase in diversity of poses seen in the augmented dataset translate to improved person detection performance?

3.4.1. Training

The end goal of this thesis is to improve person detection performance. YOLOv3[36] is used as the person detection network for training and evaluation in this work. Changes are made to the Config file to adapt the network to single class object detection for our case. Ground truth bounding boxes for the source characters are already available in the SURREAL dataset. Ground truth bounding boxes for the augmented images are obtain by locating the extreme joint locations along both the x and y axes. Subsequently, YOLOv3 is trained on both the source and augmented datasets.

3.4.2. Fine tuning

It is to be noted that the human characters present in the source images that was downloaded from the SURREAL dataset were generated synthetically using 3D software tools. As a consequence, the GAN generated images also seem similarly ”synthetic” as they borrow appearance characteristics from the source image. Visibly, there is a clear domain shift between real images of people seen in public datasets compared to human characters in our dataset. A network trained on such images shows poor performance when tested on real images, which was verified. To handle this domain shift, the weights are fine-tuned by training the network once again on a small number of images from MPII dataset[1].

3.4.3. Evaluation

Finally the trained YOLOv3 model is tested on held out samples from the MPII human pose dataset[1]. This dataset is chosen because it contains images of people displaying a wide variety of action classes. The dataset is usually used for training pose detection networks. The comprehensive dataset includes 800 human activities. Figure 3.11 for a plot of different categories.

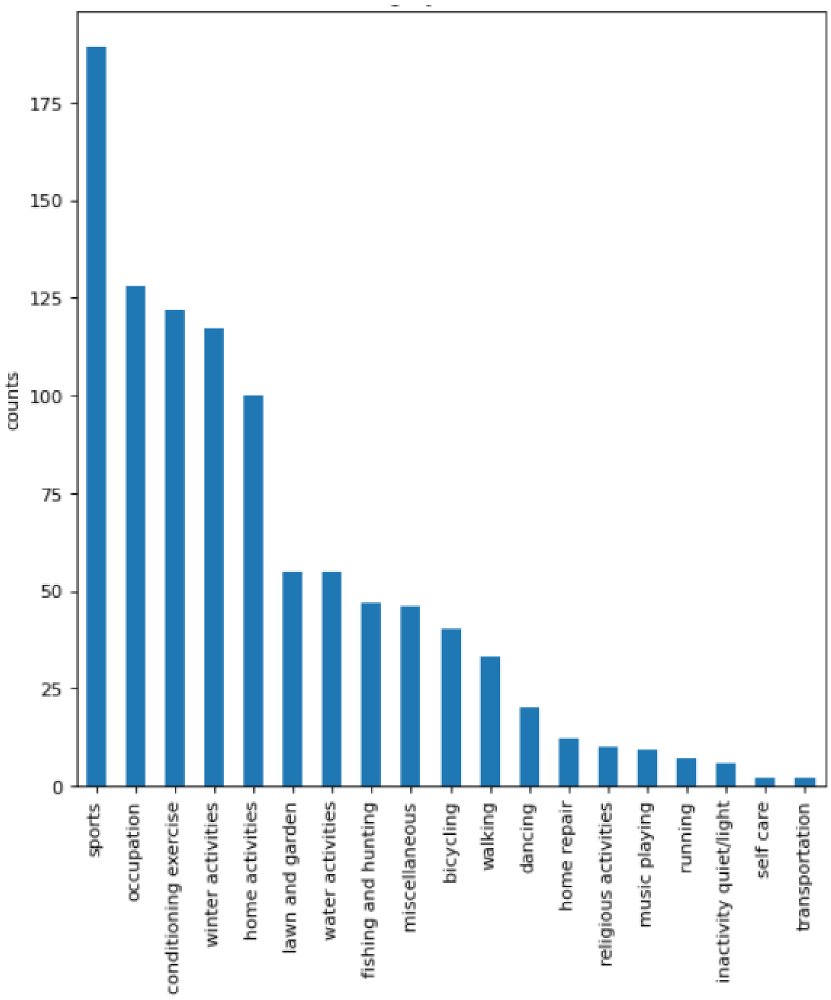


Figure 3.11: Action classes in MPII human pose dataset

The diversity and complexity of poses in this dataset makes it suitable to employ it for our case. Using such a test set allows for a fair evaluation of our method.

4
Results

This chapter presents the results of the methods explained in the methodology in chapter 3. Section 4.1 provides an analysis on different methods of pose augmentation by visualizing PCA projections of the resulting distributions and comparing respective μ_{nb} values. Section 4.2.1 details the Person detection training and testing setup and hyperparameters used. Section 4.2.1 presents the results of an evaluation of the different methods explained in Section 4.1. Section 4.2.3 investigates the role of source image dataset size, towards affecting the contribution of the data augmentation technique proposed by this work.

4.1. Pose Augmentation

Poses extracted from the source images are augmented by the technique explained in Section 3.2. In this Section, an ablation study is performed to asses the individual contributions of each component. Additionally the pose filtering technique proposed by [20] is reproduced for comparison. Below, we list the different combinations:

A: These are source poses. They are the ground truth poses of the dataset that we intend to augment. If they aren't available already, they are extracted from the images using pose detectors like Openpose[8]. This forms the baseline.

B: This dataset is created by sampling from a GMM that is trained on the source poses as explained in Section 3.2.1.

C: This dataset is created by interpolating between poses from the source without Gaussian sampling.

D: This is created by sampling a large number of poses from a trained GMM and filtered using cluster distance method discussed in[20] to be left with "rare poses". A pose is considered rare by the authors if its distance from its cluster center exceeds a certain threshold in the euclidean space. The method is covered in Section 2.3.6.

E: This is created by sampling from GMM, followed by interpolation like in Section 3.2.1 and filtering using the cluster distance method.

F: This is created by sampling from GMM and filtered using the Least Neighbours method as explained in Section 3.2.2.

G: This is created by sampling from GMM, interpolating and filtering using the Least Neighbours method as explained in Section 3.2.2. This is the complete method with all components included.

H: These are ground truth poses from the test set used for evaluating our trained YOLO models in the next section. Test images are sourced from MPII Human Pose dataset[1].

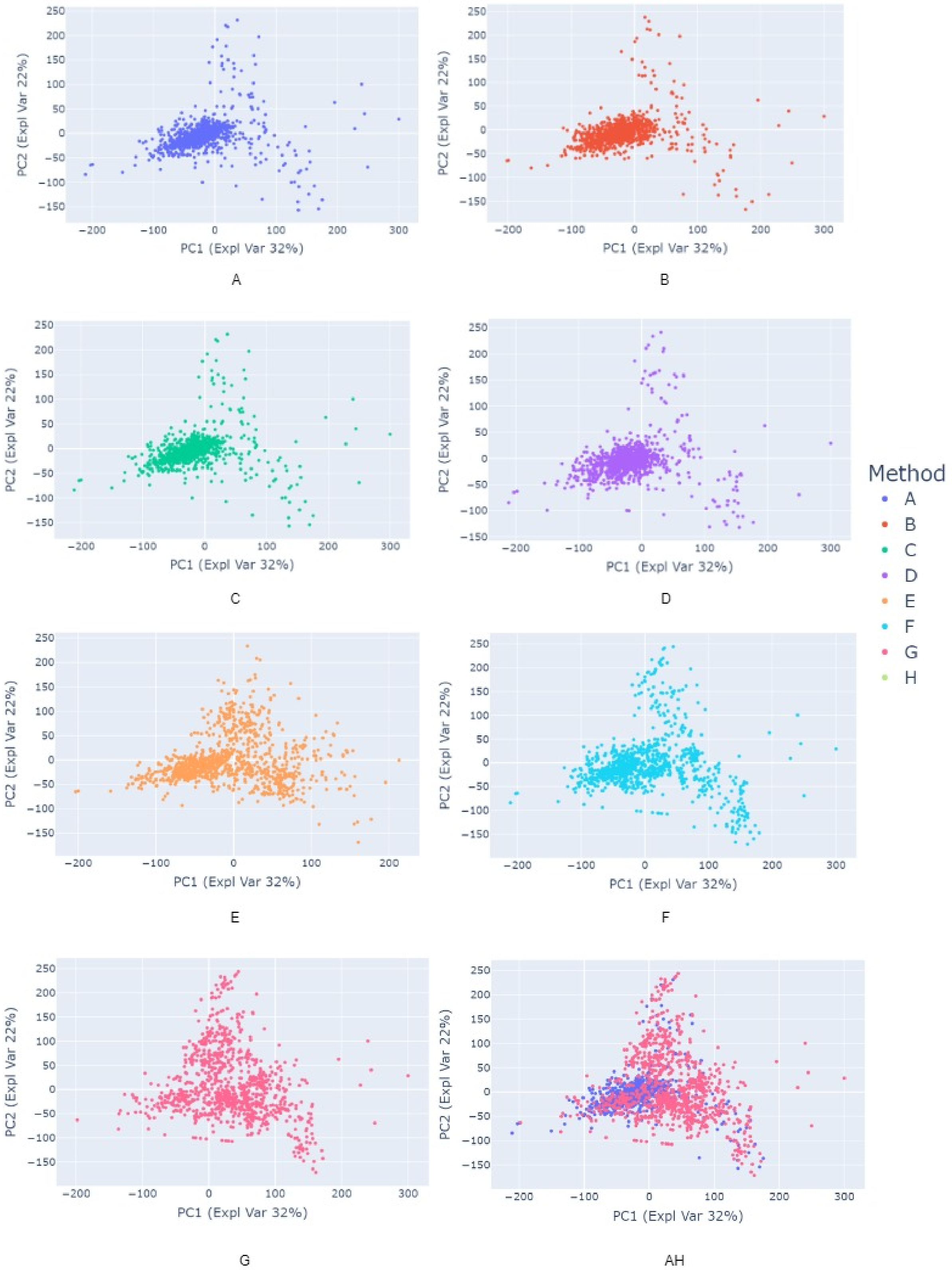
Dataset label	Method	$\mu_{nbt=0.3)}$
A	source	288
B	Gaussian Sampling	267
C	Interpolation	493
D ([20])	Gaussian Sampling + Filtering (CD)	197
E	Gaussian Sampling + Interpolation+ Filtering (CD)	101
F	Gaussian Sampling + Filtering (LN)	40
G (Ours)	Gaussian Sampling + Interpolation +Filtering (LN)	15
H	MPII test set	3

Table 4.1: An ablation study to determine contributions of individual components. Dataset G (Ours) having the least μ_{nb} is the most diverse.

Table 4.1 shows the results of pose augmentation. Column μ_{nb} shows Mean Neighbours for each of the methods. Figure 4.1 shows PCA projections on 2D space for respective methods. Each point represents a pose. Figure 4.2 shows heat maps of similarity matrices for respective methods in the format explained in Section 3.3.2. Each line represents a similarity comparison of one pose with all other poses in the set. Take the example of source dataset A. On average, a pose in A has 288 neighbours. Figure 4.1 (A) shows the respective distribution. Points are mostly concentrated in one big pocket. Points in other regions are sparse and contain empty pockets in between. This shows that most poses in the dataset are of one kind. Those poses that are different do not have equal representation. Figure 4.2 (A) shows the heat map for A. It contains dark lines in majority with streaks of lighter lines. This shows most points have a large number of neighbours and only a few proportion of points are unique. Following the same sequence of analysis for method B, we can see that it shows the same characteristics as A. It has around the same μ_{nb} value. Accordingly, points distribution (figure 4.1 (b)) and heat map (figure 4.2 (B)) is similar to A visually. This is expected given that B is sampled from a GMM trained on A. Points obtained by interpolation in method B is denser in comparison. Although 2D projection of C (figure 4.1 (C)) looks similar to A and B, shade of heat map (figure 4.2 (c)) matches with the increase in μ_{nb}. A possible explanation could be that, points that are already close to each other in A when interpolated result in an extra neighbour. Because C is not filtered it has resulted in a denser but not uniform distribution. Method D and E add the filtering through cluster distance step to methods B and C. This additional step decreases μ_{nb} in both. PCA projection of D shows a distribution mostly similar to A,B with slight increase in area of the densely populated cluster (figure 4.1 (D)). This has resulted in a decrease in μ_{nb} . Accordingly corresponding heat map shows a lighter shade (figure 4.2 (D)).In E, points are distributed more evenly with increased coverage due to new points being generated from interpolation (figure 4.1 (E)). Accordingly the heat map (figure 4.2 (E)) has increased number of lighter shaded lines. Method F and G replaces the filtering technique used in D and E with Least Neighbours filtering method proposed by this thesis. The value of μ_{nb} for F and G is lowered in both cases compared to D and E. Respective PCA projections (figure 4.1 (F), figure 4.1 (G)) and heat maps (figure 4.2 (F), figure 4.2 (G)) make the improvements more evident. This improvement can be attributed to the way points are filtered in E and F. In the D and E, poses are filtered based on their distance to their respective mean pose. This can result in poses that are away from the mean but similar to each other. On the other hand, in E and F, filtering takes into account similarity of points with each other. Figure 4.1 (AG) shows projections of A (source) and the best method G (complete version) overlapped. Figure 4.3 shows histogram plots for A and G overlapped. Most of the points in A have more than 300 neighbours each. On the other hand, most points in G have neighbours less than 100. This shows that G is significantly more diverse than A.

To summarize, an ablation study was performed comparing contributions of individual components. The effect of adding extra components are largely additive in nature. The exception is interpolation without filtering which further increased μ_{nb} compared to baseline A. The PCA projections of distributions and heat maps match the change in μ_{nb} values. The model having the least μ_{nb} was found to be the method G which includes all components. It is an improvement upon the method used in [20].

Following this, Step 3 explained in the previous section is performed to obtain augmented images. Six different sets of augmented images are obtained by using six different sets of Poses (B, C, D, E, F, G).



A

B

C

D

E

F

G

AH

Figure 4.1: Pose keypoints from different datasets projected on 2D space using PCA. Points in Dataset G shows the best coverage and is most uniform

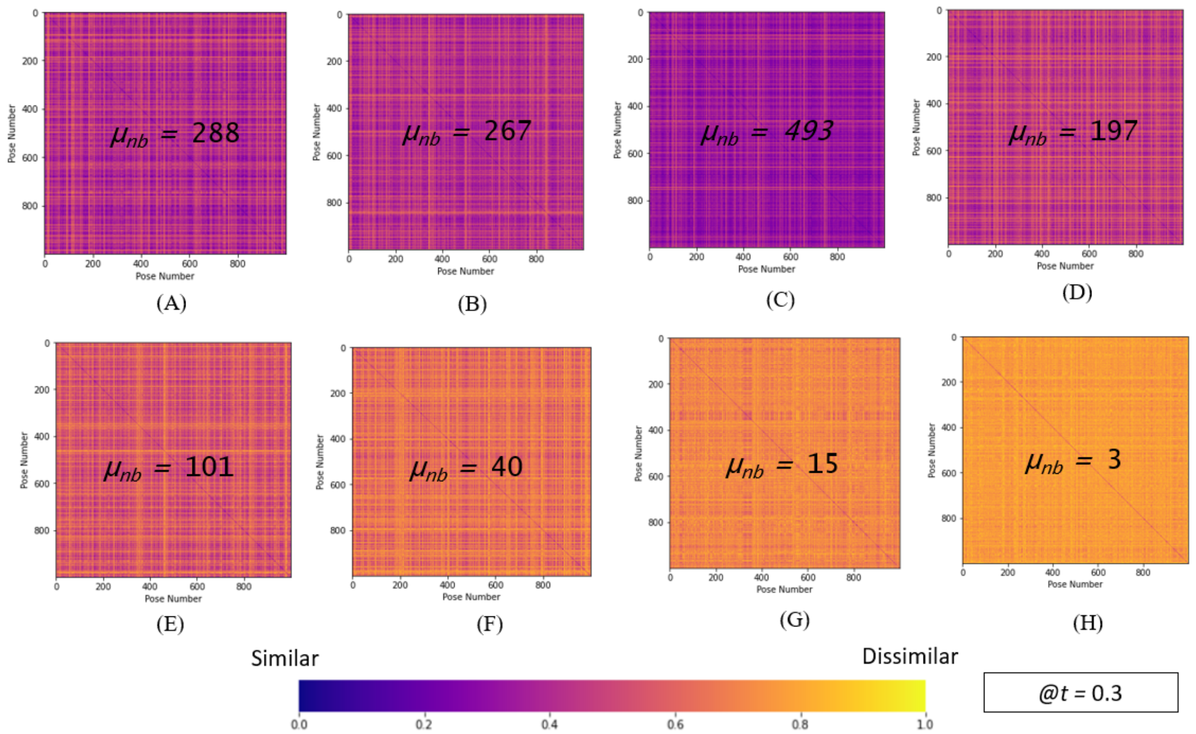


Figure 4.2: Heat map of Pose Similarity Matrices for different methods for threshold t=0.3. Each line represents a pose from the dataset. A lighter line means the corresponding pose has less neighbours. A darker line means the corresponding pose had more neighbours. The dataset having the least Mean Neighbours is the lightest shaded matrix and also the most diverse.

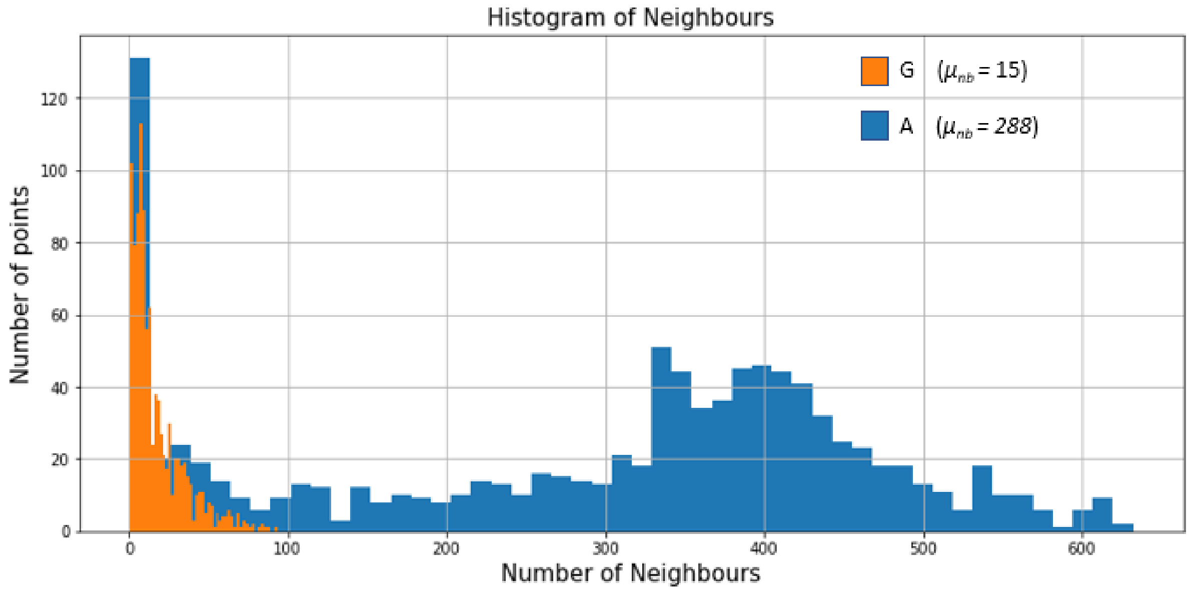


Figure 4.3: Histograms of source and augmented pose datasets. source data consists of majority points having greater number of neighbours. Augmented data has lesser points with less number of neighbours. This shows augmented data is more diverse compared to source

4.2. Person detection evaluation

This step involves validation of the proposed data augmentation technique using a person detection network and comparing metrics with baseline.

4.2.1. Training set up

Training

YOLOv3[6] is used as our person detection network. The images for training are obtained from the methods explained in the previous section. The training strategy recommended by[35] is followed. The number of training images is 12000 for all methods. Weights until bottom last 3 layers are frozen for first 10 epochs. Only last 3 layers are trained initially at learning rate $1e^{-3}$. After 10 epochs all weights are unfrozen and trained with a learning rate $1e^{-4}$. Further learning rates are set by a scheduler. Weights from ImageNet trained DarkNet-53[36] model are used for initialization. Data augmentation like rotations, flipping, random scaling and cropping is performed online. An early stopping mechanism is built into the training loop. Training time is 15hrs using 1 P100 GPU (Google Colab Pro+ Environment).

Fine tuning

Trained weights are fine tuned on 250 images from MPII human pose dataset[1] for adjusting to domain shift. Learning rate for first 5 epochs is $1e^{-4}$ with partially frozen weights and $1e^{-5}$ with all weights unfrozen. Training time is 30 min.

4.2.2. Ranking methods

Dataset Label	Method	Precision (%)	Recall (%)	F1 (%)	AP (%)
A	source	68.4	55.0	61.3	46.5
B	Gaussian Sampling	68.0	56.1	61.5	46.1
C	Interpolation	67.6	52.1	58.8	45.1
D[20]	Gaussian Sampling + Filtering (CD)	73.4	55.1	63.5	47.2
E	Gaussian Sampling + Interpolation+ Filtering (CD)	73.1	56.3	63.6	47.2
F	Gaussian Sampling + Filtering (LN)	77.3	59.1	67.0	48.6
G (Ours)	Gaussian Sampling + Interpolation +Filtering (LN)	68.0	60.2	63.9	51.3

Table 4.2: Results tabulated for YOLOv3 trained on different methods and tested on MPII dataset[1]

Table 4.2 presents the results on four metrics—1)Precision, 2)Recall, 3)F1 score, 4)Average Precision . Precision gives an idea of accuracy of predictions. Recall gives an idea of proportion of True Positives picked. F1 score is the best of both Precision and Recall at an optimal confidence threshold. Average Precision (AP) measures the area under the Precision vs Recall curve plotted for different confidence threshold. In terms of priority, AP comes first because it is calculated globally and is therefore more robust. It is also the most used metric in academia for object detection. This is followed by the F1 score.

The person detector trained on dataset G (complete method), shows the best performance overall at AP=51.3%. AP is increased by 4.8 percentage points compared to the baseline A. AP for B is around the same as A. This is expected because poses in the dataset were sampled from a GMM trained on source poses. A dip in AP is shown for C, the interpolated set. This matches with the observation made in the previous section regarding the increase in μ_{nb}. Datasets D,E which use filtering technique proposed by [20] shows an improvement upon baseline A. F is the second best performing model showing AP=48.6%.Figure 4.4 shows the Precision vs Recall curve for the same at an IOU=0.5.

Some interesting observations can be drawn by plotting all the metrics discussed against μ_{nb}, shown by Figure 4.7. In general there is a strong negative correlation between μ_{nb} and Recall. Recall increases steadily with decrease in μ_{nb}. This is understandable because decrease in μ_{nb} means points are spread more uniformly along the space.This happens when distribution covers a larger area, ending up with more diverse samples. As a result, a network trained on such a dataset picks up defections of people in difficult poses, that weren't otherwise picked. This directly improves the Recall of the model. Recall influences AP and therefore we see a constant improvement in AP. There is however a small abberation seen in dataset B to this trend. However this may be attributed to the random seed hyperparameter of the GMM that controls the generation of random samples from the fitted distribution. The GMM outputs a different set of samples for the same distribution depending on the seed. This would affect the results in a small way on either side. Therefore, this does not question the correlation observed. Precision also increases with decrease in μ_{nb} except in G. A combination of interpolation

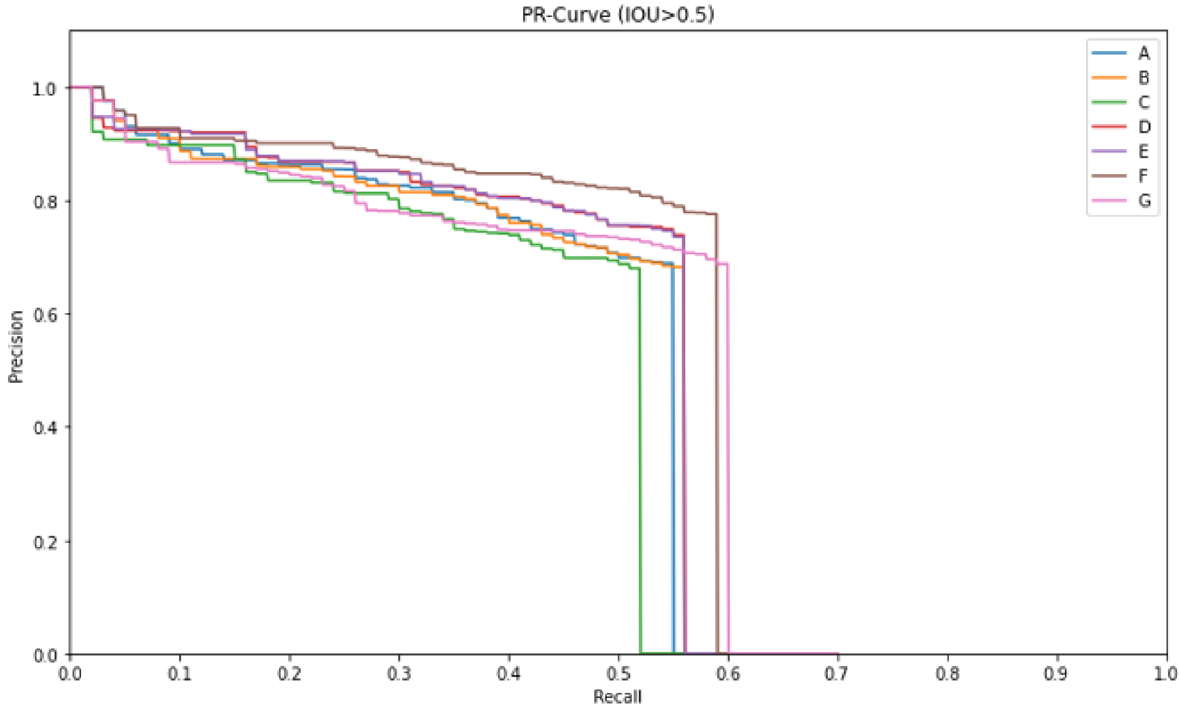


Figure 4.4: Precision vs Recall curve for differnt methods.

and filtering with Least neighbours seems to decrease precision. However AP increases nevertheless, since it is more sensitive to Recall than Precision.

To summarize, the evaluation showed that YOLO object detection network trained on dataset G improved AP by 4.8 percentage points compared to the baseline A. G represents the dataset created by including all components of the data augmentation method proposed by this work. It was also shown that decrease in μ_{nb} translates to an increase in AP. This means that μ_{nb} can be used as a metric to assess diversity in pose datasets.

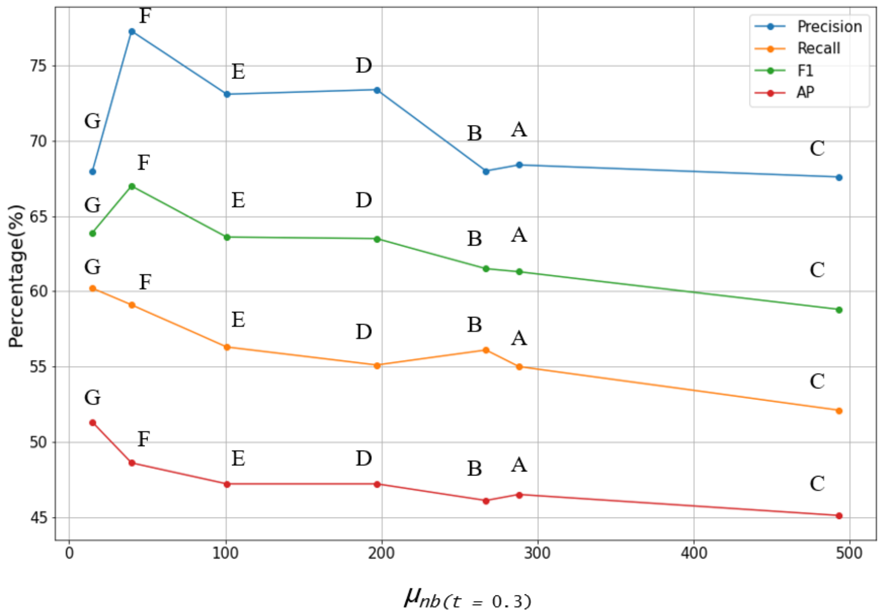


Figure 4.5: Plot of μ_{nb} vs object detection metrics for all datasets. It shows a negative correlation between μ_{nb} and AP. This shows increase in pose diversity translates to improved person detection performance.

4.2.3. Effect of dataset size

In this Section we investigate the influence of source dataset size on the effectiveness of the proposed augmentation technique. The previous section saw experiments with 12000 source images. Two more source datasets are created having 6000, 25000 images respectively. Pose Augmentation is performed on each of these datasets using method G.Number of images is maintained the same for both source and augmented images for a fair comparison. The number of images used for fine-tuning is also maintained the same for all datasets.

Table 4.3 summarizes the test results.Augmentation results in an increase of AP for small and medium sized source datasets (6000,12000 images) considered. For the smallest source dataset, augmentation improved AP by 8.4 percentage points. For medium sized source dataset (12000 images), augmentation improved AP by 4.8 percentage points. For the large source dataset (25000 images), augmentation has in-fact decreased the performance slightly by 1 percentage point. Figure 4.6 further emphasizes the case of diminishing returns. Precision, Recall, F1 and AP values are plotted for all the datasets considered. For example, ”Src 6k” refers to source dataset having 6000 images. It can be observed from the AP plot that slope decreases gradually until it tapers off at Src 25k. It infact decreases slightly at Aug 25k. Let us first look at only source images. source images also show a diminishing increase in performance with increase in dataset size. This is already expected. This matches with the observations made in [33], discussed earlier in Section 2.1. The authors observed that impact of pre-training networks with synthetic data was more pronounced for low data regimes. However, we are more concerned about the impact of additional data augmentation using our technique. Observing the slopes between Src and Aug for different dataset sizes shows that here too, the difference is highest for the smallest source dataset. The same behaviour is observed by [7] (Table 3) for GAN augmented images, discussed earlier in Section 2.1. Using GAN generated image patches, the authors observed an improvement in brain CT scan segmentation performance by 3.3 percentage points for small dataset vs only 1.2 percentage points improvement for large dataset. This should be expected because we technically did not add any ”new” information to the source datasets. The pose data augmentation part added variety through interpolations. Synthesis using Pose-Warp GAN added extra characters in already seen clothes and colours. This resulted in more permutations and combinations inside the manifold of all possibilities. However larger datasets might already have enough diversity. Therefore, the regularizing effect of this kind of data enrichment should have a theoretical limit. It should also be noted that in all the experiments, the entire set of source images was not carried over to the augmented dataset. In order to maintain the same number of total images between the two datasets, part of source images (that are less diverse) were removed during the filtering step. This information loss could also influence the true contribution of the proposed data augmentation.

Dataset	Training Annotations	Fine-tuning Annotations	Precision (%)	Recall (%)	F1 (%)	AP (%)
source	6000	250	53.9	38.0	44.6	28.7
augmented (G)	6000	250	58.8	47.2	52.4	37.1
source	12000	250	66.4	55.0	61.3	46.5
augmented (G)	12000	250	68.0	60.2	63.9	51.3
source	25000	250	73.3	66.9	70.0	53.8
augmented (G)	25000	250	69.4	66.0	67.0	52.8

Table 4.3: Results of YOLOv3 trained on different sized source and augmented datasets and tested on MPII dataset[1].

Additionally we perform perform two more tests. To analyse the impact of pre-training using synthetic data, YOLOv3 was trained on only real images. The number of training images is kept the same as what was used during fine-tuning for previous methods. Next, we create another augmented dataset whose images contain multiple human characters. This is done because in all previous tests, datasets contained single annotation per image. Also, the entire set of source images is added to the augmented set. Therefore the total annotations in this case is 50000 with equal proportion of source and augmented characters. Results for the two experiments are presented in Table 4.4. Training only on 250 real images results in very poor performance. This shows that synthetic pre-training provides a good initialization for further fine-tuning on real images. Synthetic data is therefore a viable option for pre-training networks and is consistent with observations made in literature study in Section 2.1. Next we look at the performance of the augmented dataset. This dataset reports the best results compared to all previous experiments. This can be attributed to increased number of total annotations. Also,

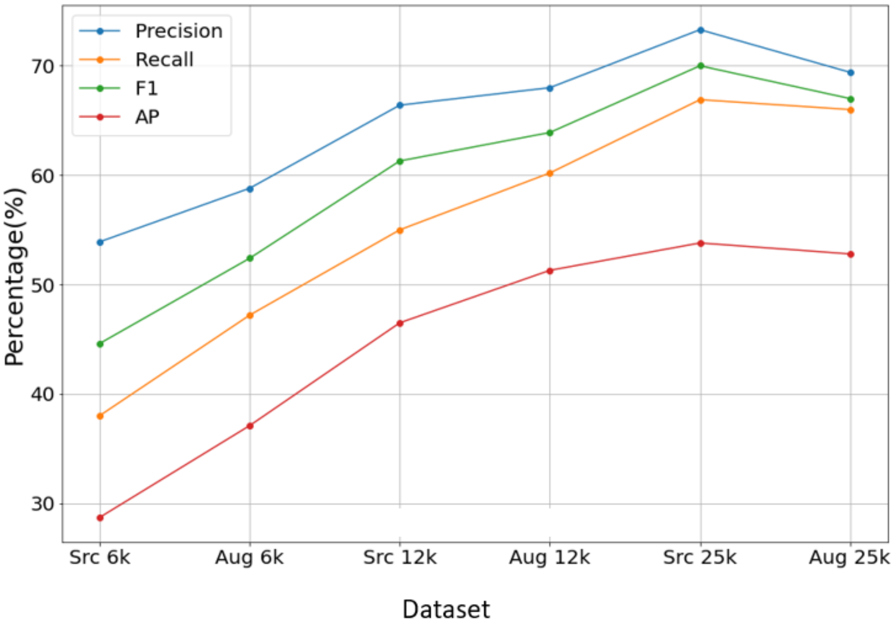


Figure 4.6: All metrics plotted for YOLOv3 trained on different sized source and augmented datasets. Plot shows diminishing returns of improvement due to augmentation. Precision drops for largest sized dataset. Highest improvement observed from lowest sized dataset (6000 images).

pasting multiple characters on the same image creates occlusions and overlapping body parts, which makes for a more difficult dataset. Such scenarios occur routinely in real world images. Refer figure A.9 for example images from the dataset. Figure A.10 shows detections picked up by the network when trained on images with multiple characters, that were missed previously.

Dataset	Training Annotations	Fine-tuning Annotations	Precision (%)	Recall (%)	F1 (%)	AP (%)
Real Images	250	NA	10.10	6.20	7.7	2.2
Aug Multi Annot (G)	50000	250	70.8	69.8	70.3	59.8

Table 4.4: Results for YOLOv3 trained dataset of only real images and a dataset with images having mutiple characters per image.

4.3. Discussion

In this Section, we provide further commentary on the results obtained and postulate on some of the behaviours observed.

The technique of interpolating between poses to obtain new poses has both pros and cons. Interpolation resulted in new poses with more variations that were unseen in the source data. Characters in these new poses were synthesized by the GAN successfully which enriched the data. Figure A.3 shows some examples. This increased the Recall of the model when trained as was observed in the previous section from Figure 4.7. Figure A.4 shows some detections of people in ”difficult” poses that were otherwise not picked. However, interpolation also sometimes produces poses that seem unrealistic and impossible because there is no control. Joints can end up bent either too much, or the direction reversed. Figure A.6 shows some example of such poses resulting from interpolation. Pose-Warp GAN sometimes already produces outputs that are missing limbs and incoherent. When it is asked to synthesize characters conditioned on such implausible poses, the outputs are even poorer. This adds noise to the dataset as the network is forced to learn these blotchy pixels to be detections. This can result in increased False Positives and decreased Precision. Figure A.8 shows some examples of False positives. The network incorrectly classifies rectangle shaped silhouettes as persons including reflections. This trade of between Precision and Recall is more pronounced for larger datasets as observed in figure 4.6. This could be because the larger dataset already has a considerable amount of pose variations. Also, It is to be noted

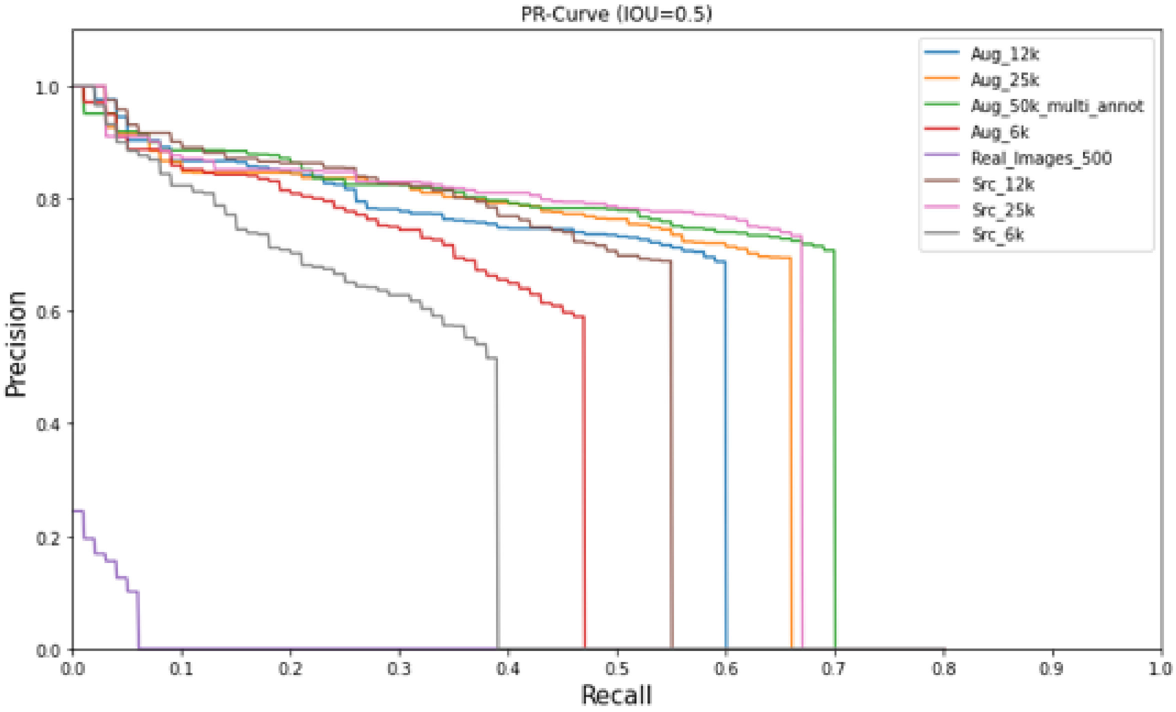


Figure 4.7: Precision vs Recall curve for different sized datasets.

that although interpolation fills holes in missing regions, it does not extend the boundaries of the distribution. This can only be done by adding new data.

Ultimately, to expect good performance, the training distribution must be representative of the test distribution. To unlock the true potential of augmentation, one must ensure that the test set is diverse enough. This can be verified from figure 4.8. The figure shows the 2D PCA projections of poses from source, augmented set and the test set overlapped. Points in the test set cover a larger area and is distributed more uniformly. Accordingly, this set returned the lowest Mean Neighbours (μ_{nb}=3) out of all the distributions seen. Figure 4.2 (H) shows the corresponding heat map. Therefore, it can be reasoned that increase in diversity of poses resulted in an increased person detection performance.

In the experiments conducted in the previous section, the size of the augmented data was forced to be the same as the source data. In the real world this is not necessary as augmented images were cheap and required no extra labels. There is no limit to the number of images that could be synthesized this manner. Therefore it would be interesting to understand the limit of performance improvement by testing on different sized augmented sets.

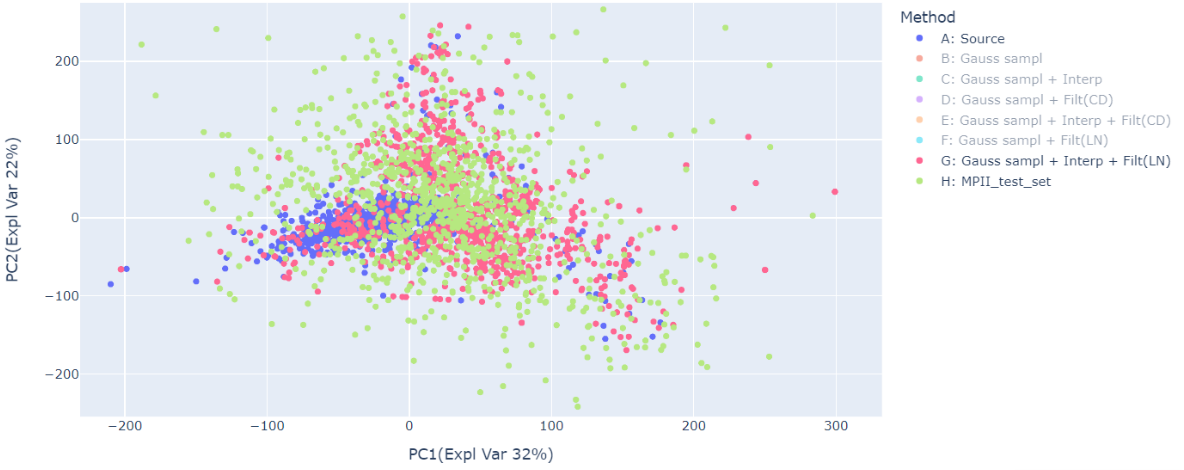


Figure 4.8: PCA projection on 2D space of Pose keypoints from A, G and H overlapped. H represents poses from the test images used during our evaluation. Poses from this dataset are the most diverse compared to all others. This means there is still scope for improvement in diversity for our augmented dataset.)

5 Conclusion

This goal of this research was to improve the performance of person detection networks when trained on images that are synthetic images. The problem working with synthetic images that are generated using 3D softwares is that,it requires manual effort in infusing diversity and ensuring class balance. For example, in order to include more variations of poses and colours, one needs to explicitly provide the animations and clothing skins to the rendering pipeline. As an alternative, we proposed to augment synthetic datasets with new human characters in unseen poses using a pose conditional GAN. In this section, we summarize the results from the previous section and reiterate the answers to the research questions listed at the end of Section 2.

5.0.1. Pose Augmentation

Q. 1 How to synthesized new, diverse poses from poses present in the original dataset?

Q. 2 How to measure the diversity of poses in a given dataset?

Regarding Q.1, Section 3.1 and 3.2 explained the proposed method of pose augmentation. The steps involve sampling poses from a GMM, interpolation between poses and filtering using Least Neighbours Filtering technique. This resulted in a dataset of poses that was more diverse and uniform compared to the poses in the Source(original) dataset. This was verified quantitatively by visualizing the respective distributions using 2D PCA projections(figure 4.1). The distribution of points in the augmented dataset was more spread out and uniform.Further, an ablation study was performed in Section 4.1 to assess the individual contributions of the different components in our proposed method of augmentation. The effect of adding extra components were found to be largely additive in nature. This was again verified by observing their respective PCA projections(figure 4.1).

Regarding Q. 2, Section 3.2 introduced a metric μ_{nb} that gives the mean number of neighbours for of a given pose in the dataset. It is a measure of diversity in pose distributions. For similar sized datasets, if poses in the augmented dataset have lesser number of neighbours compared to the Source, then the augmented set needs to be more spread out. This is verified in the results, in Section 4.1 by comparing μ_{nb} values of different methods and their respective distributions (refer Table 4.2). The correlation was further emphasized through heat maps and a histogram plot in Section 4.1(figures 4.2, 4.3). Finally, The model having the least μ_{nb} was found to be the method G, the proposed technique which includes all components. This method is an improvement upon the method used in [20].

5.0.2. Person detection Evaluation

Q. 3 Does increase in diversity of poses seen in the augmented dataset translate to improved person detection performance?

Regarding Q. 3 Experiments were conducted in Section 4.2.4 for different methods of pose augmentation. Results showed a clear negative correlation between respective μ_{nb} values and AP. For dataset size of 12000 images, the person detector trained on dataset G(complete method), showed the best performance overall at AP=51.3% . AP increased by 4.8 percentage points for G compared to the baseline A. It is also more than 4.1

percentage points compared to D and hence and improvement upon [20].

Experiments in Section 4.2.3 and 4.2.4 investigated the effect of our data augmentation method for different sized datasets. Given that we had forced our Augmented dataset to have same size as the Source dataset, a diminishing increase in performance was observed with increase in dataset size. The highest improvement in performance(8.4 percentage points) was observed for the dataset with least number of images. This trend is consistent with what was observed in the literature[7][34]. The best reported AP out of all experiments was 59%, for the Augmented dataset with multiple annotations per image. This dataset had occluded people which mimicked scenarios occurring in real world images. Also, it is important to note that for all the experiments, traditional data augmentation (rotations, scaling, flipping, cropping) was built into the image loader during training. Therefore our data augmentation is an improvement over and above traditional data augmentation methods.

5.1. Recommendations and Future scope

- The interpolation technique used in this work to augment poses did not set limits to joint rotations, resulting in some unrealistic poses. Synthesis of human characters with Pose-Warp GAN conditioned on such poses gave incoherent pixel patches. This added noise to the dataset. An improvement could be to set joint limits. A mask with appropriate joint limits can be multiplied with the interpolated pose to obtain more ”human-like” poses.
- Alternatively, using a superior pose conditioned GAN that can generalize better to different poses, can also result in less noise by synthesizing better images.
- Pose-Warp GAN ensures that the characters in the input image and the output image have the same clothes. If data augmentation is the goal this is not necessary. The network could be modified to pick a random color from a user provided color palette. This can increase color variations seen in the dataset.
- It would be interesting to apply the proposed data augmentation for pose detection application as it directly deals with augmenting poses. This could have a much bigger impact compared to object detection.

References

[1] Mykhaylo Andriluka et al. “2d human pose estimation: New benchmark and state of the art analysis”. In: *Proceedings of the IEEE Conference on computer Vision and Pattern Recognition.* 2014, pp. 3686–3693.

[2] Guha Balakrishnan, Amy Zhao, and Adrian V Dalca. “Fredo, Durand, and John Guttag”. In: *Synthesizing images of humans in, unseen poses. In, CVPR* 2 (2018).

[3] Andrew Beers et al. “High-resolution medical image synthesis using progressively grown generative adversarial networks”. In: *arXiv preprint arXiv:1805.03144* (2018).

[4] Karan Bhanot et al. “The problem of fairness in synthetic healthcare data”. In: *Entropy* 23.9 (2021), p. 1165.

[5] Jordan J Bird et al. “Fruit quality and defect image classification with conditional GAN data augmentation”. In: *Scientia Horticulturae* 293 (2022), p. 110684.

[6] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. “Yolov4: Optimal speed and accuracy of object detection”. In: *arXiv preprint arXiv:2004.10934* (2020).

[7] Christopher Bowles et al. “Gan augmentation: Augmenting training data using generative adversarial networks”. In: *arXiv preprint arXiv:1810.10863* (2018).

[8] Zhe Cao et al. “Realtime multi-person 2d pose estimation using part affinity fields”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017, pp. 7291–7299.

[9] *Carnegie Mellon University - CMU Graphics Lab - motion capture library.* `http://mocap.cs.cmu.edu/`. October 17, 2022.

[10] Caroline Chan et al. “Everybody dance now”. In: *Proceedings of the IEEE/CVF international conference on computer vision.* 2019, pp. 5933–5942.

[11] Nitesh V Chawla et al. “SMOTE: synthetic minority over-sampling technique”. In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.

[12] Ernest Cheung et al. “Mixedpeds: pedestrian detection in unannotated videos using synthetically generated human-agents for training”. In: *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 32. 1. 2018.

[13] Matteo Fabbri et al. “Motsynth: How can synthetic data help pedestrian detection and tracking?” In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2021, pp. 10849–10859.

[14] Maayan Frid-Adar et al. “GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification”. In: *Neurocomputing* 321 (2018), pp. 321–331.

[15] Michael Goebel et al. “Detection, attribution and localization of gan generated images”. In: *Electronic Imaging* 2021.4 (2021), pp. 276–1.

[16] Ian Goodfellow et al. “Generative adversarial networks”. In: *Communications of the ACM* 63.11 (2020), pp. 139–144.

[17] Maryam Hammami, Denis Friboulet, and Razmig Kéchichian. “Cycle GAN-based data augmentation for multi-organ detection in CT images via Yolo”. In: *2020 IEEE International Conference on Image Processing (ICIP).* IEEE. 2020, pp. 390–393.

[18] David T Hoffmann et al. “Learning to train with synthetic humans”. In: *German conference on pattern recognition.* Springer. 2019, pp. 609–623.

[19] Sheng-Wei Huang et al. “Auggan: Cross domain adaptation with gan-based data augmentation”. In: *Proceedings of the European Conference on Computer Vision (ECCV).* 2018, pp. 718–731.

[20] Jihye Hwang, John Yang, and Nojun Kwak. “Exploring Rare Pose in Human Pose Estimation”. In: *IEEE Access* 8 (2020), pp. 194964–194977.

[21] Phillip Isola et al. “Image-to-image translation with conditional adversarial networks”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017, pp. 1125–1134.

[22] Tomas Jakab et al. “Self-supervised learning of interpretable keypoints from unlabelled videos”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2020, pp. 8787–8797.

[23] Wei Jiang and Na Ying. “Improve object detection by data enhancement based on generative adversarial nets”. In: *arXiv preprint arXiv:1903.01716* (2019).

[24] Arnab Karmakar and Deepak Mishra. “Pose invariant person re-identification using robust pose-transformation gan”. In: *arXiv preprint arXiv:2105.00930* (2021).

[25] Tero Karras et al. “Analyzing and improving the image quality of stylegan”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition.* 2020, pp. 8110–8119.

[26] Tero Karras et al. “Progressive growing of gans for improved quality, stability, and variation”. In: *arXiv preprint arXiv:1710.10196* (2017).

[27] Alina Kuznetsova et al. “The open images dataset v4”. In: *International Journal of Computer Vision* 128.7 (2020), pp. 1956–1981.

[28] Christoph Lassner, Gerard Pons-Moll, and Peter V Gehler. “A generative model of people in clothing”. In: *Proceedings of the IEEE International Conference on Computer Vision.* 2017, pp. 853–862.

[29] Weixing Liu, Jun Liu, and Bin Luo. “Can Synthetic Data Improve Object Detection Results for Remote Sensing Images?” In: *arXiv preprint arXiv:2006.05015* (2020).

[30] Matthew Loper et al. “SMPL: A skinned multi-person linear model”. In: *ACM transactions on graphics (TOG)* 34.6 (2015), pp. 1–16.

[31] Stefan Milz, Tobias Rudiger, and Sebastian Suss. “Aerial ganeration: Towards realistic data augmentation using conditional gans”. In: *Proceedings of the European Conference on Computer Vision (ECCV) Workshops.* 2018, pp. 0–0.

[32] Saman Motamed, Patrik Rogalla, and Farzad Khalvati. “Data augmentation using Generative Adversarial Networks (GANs) for GAN-based detection of Pneumonia and COVID-19 in chest X-ray images”. In: *Informatics in Medicine Unlocked* 27 (2021), p. 100779.

[33] Farzan Erlik Nowruzi et al. “How much real data do we actually need: Analyzing object detection performance using synthetic and real data”. In: *arXiv preprint arXiv:1907.07061* (2019).

[34] Ricardo Silva Peres et al. “Generative adversarial networks for data augmentation in structural adhesive inspection”. In: *Applied Sciences* 11.7 (2021), p. 3086.

[35] *qqwweee/keras-yolo3: A Keras implementation of YOLOv3 (Tensorflow backend).* `https://github.com/qqwweee/keras-yolo3`. October 17, 2022.

[36] Joseph Redmon and Ali Farhadi. “Yolov3: An incremental improvement”. In: *arXiv preprint arXiv:1804.02767* (2018).

[37] Zhihang Ren, Stella X Yu, and David Whitney. “Controllable medical image generation via generative adversarial networks”. In: *Electronic Imaging* 2021.11 (2021), pp. 112–1.

[38] Ahmad El Sallab et al. “LiDAR Sensor modeling and Data augmentation with GANs for Autonomous driving”. In: *arXiv preprint arXiv:1905.07290* (2019).

[39] Umesh C Sharma et al. “Modified GAN Augmentation Algorithms for the MRI-Classification of Myocardial Scar Tissue in Ischemic Cardiomyopathy”. In: *Frontiers in cardiovascular medicine* (2021), p. 1097.

[40] Fabio Henrique Kiyoiti dos Santos Tanaka and Claus Aranha. “Data augmentation using GANs”. In: *arXiv preprint arXiv:1904.09135* (2019).

[41] Jonathan Tremblay et al. “Training deep networks with synthetic data: Bridging the reality gap by domain randomization”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops.* 2018, pp. 969–977.

[42] Gul Varol et al. “Learning from synthetic humans”. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2017, pp. 109–117.

[43] Xian Wu et al. “Deep portrait image completion and extrapolation”. In: *IEEE Transactions on Image Processing* 29 (2019), pp. 2344–2355.

[44] Jie Yu et al. “Improving person detection using synthetic training data”. In: *2010 IEEE International Conference on Image Processing.* IEEE. 2010, pp. 3477–3480.

[45] Zhedong Zheng et al. “Joint discriminative and generative learning for person re-identification”. In: *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 2138–2147.

[46] Xingran Zhou et al. “Text guided person image synthesis”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 3663–3672.

[47] Jun-Yan Zhu et al. “Unpaired image-to-image translation using cycle-consistent adversarial networks”. In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2223–2232.

[48] Xinyue Zhu et al. “Emotion classification with data augmentation using generative adversarial networks”. In: *Pacific-Asia conference on knowledge discovery and data mining*. Springer. 2018, pp. 349–360.

A

Appendix

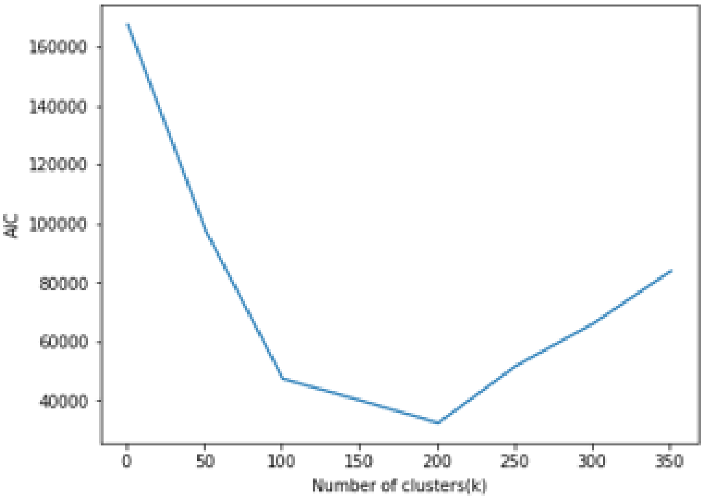


Figure A.1: AIC estimates the relative amount of information lost by a given model. At some point there is over-fitting when the number of clusters are too high. Number of clusters corresponding to lowest loss is selected.

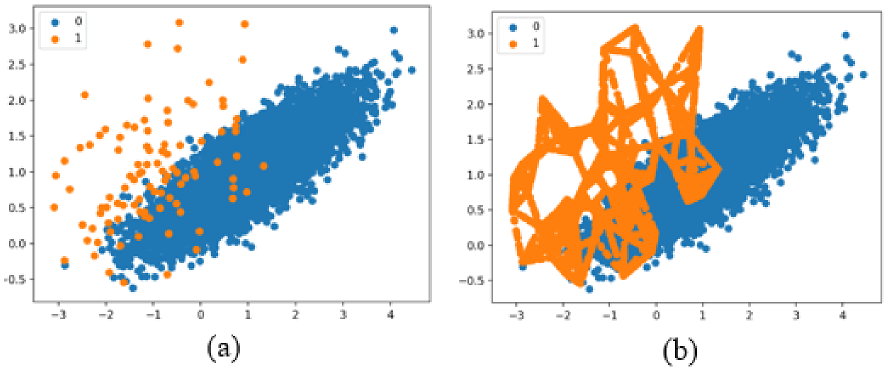


Figure A.2: Oversampling technique in [11]. (a) shows minority class 1 points thinly populated. (b) shows new points in class 1 after interpolating between neighbouring points.

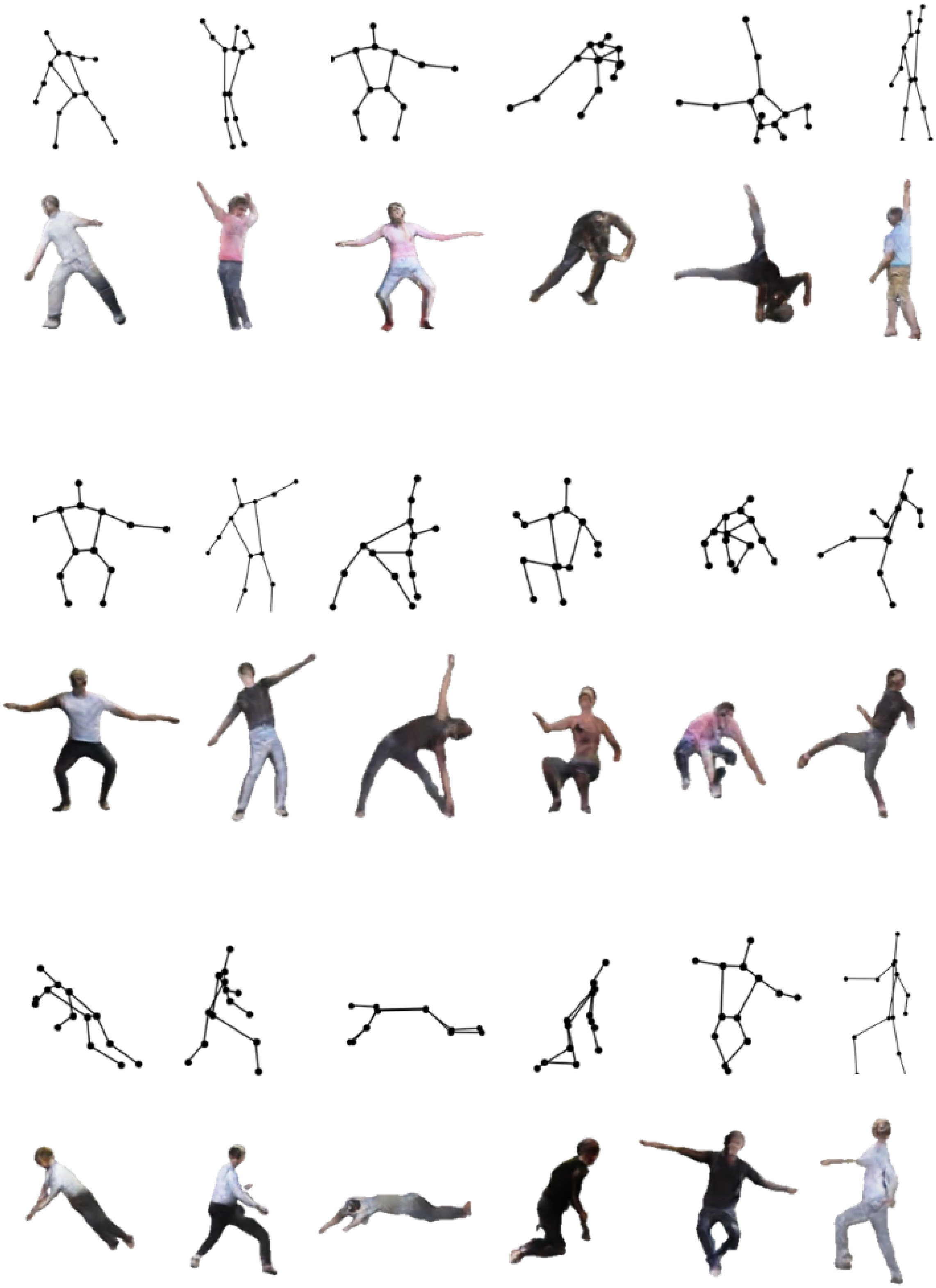


Figure A.3: Examples of good new poses and good outputs from Pose-Warp GAN.

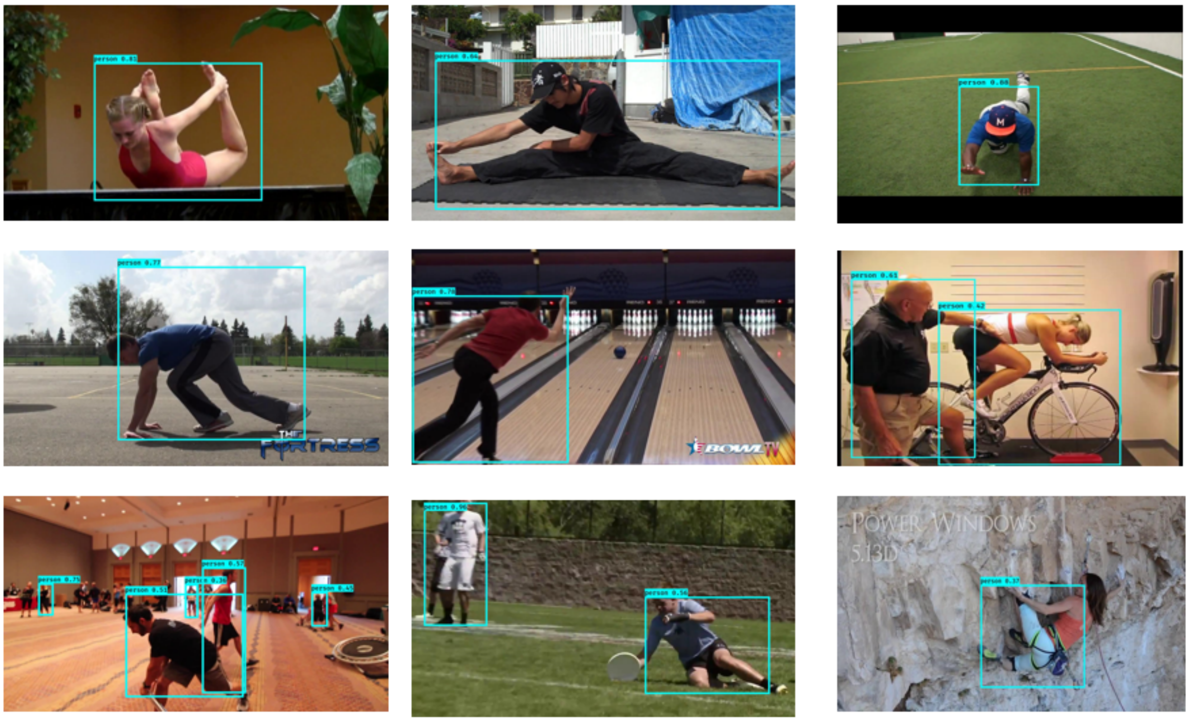


Figure A.4: Difficult detections picked up by YOLOv3 trained on Augmented dataset.

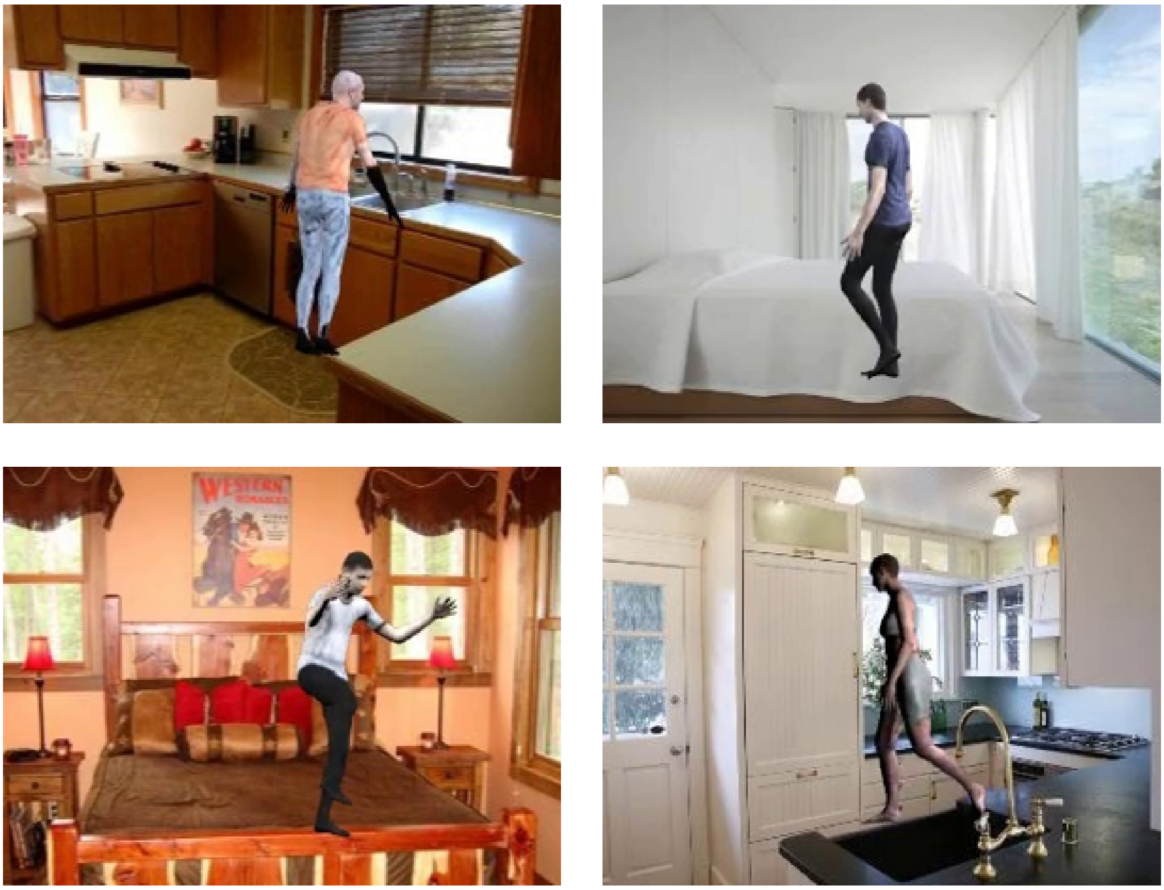


Figure A.5: Example images from SURREAL dataset[42]

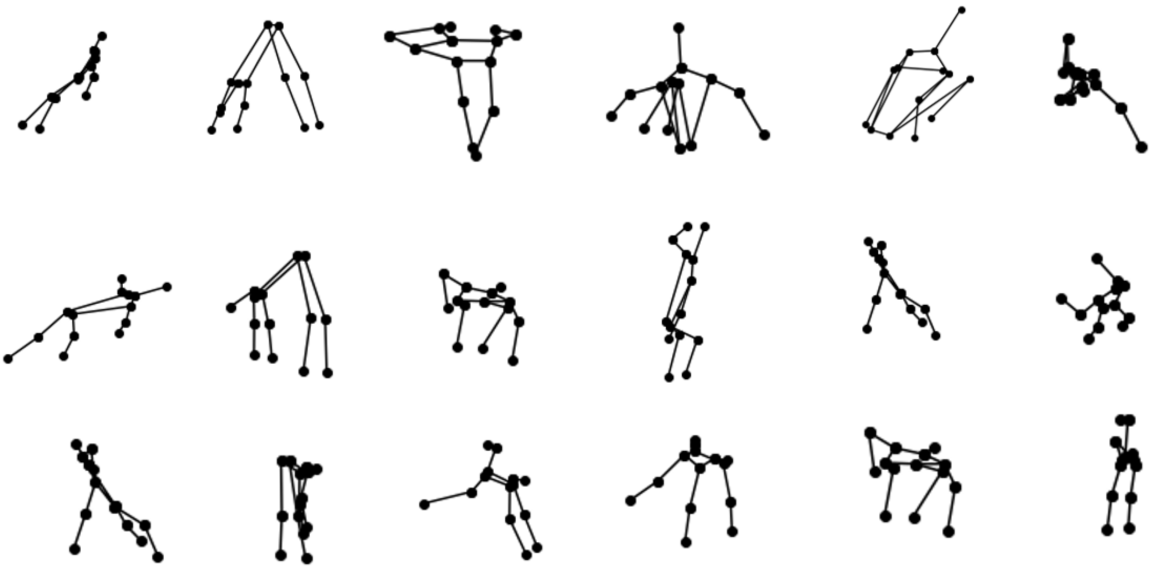


Figure A.6: Examples of unrealistic poses due to interpolation.

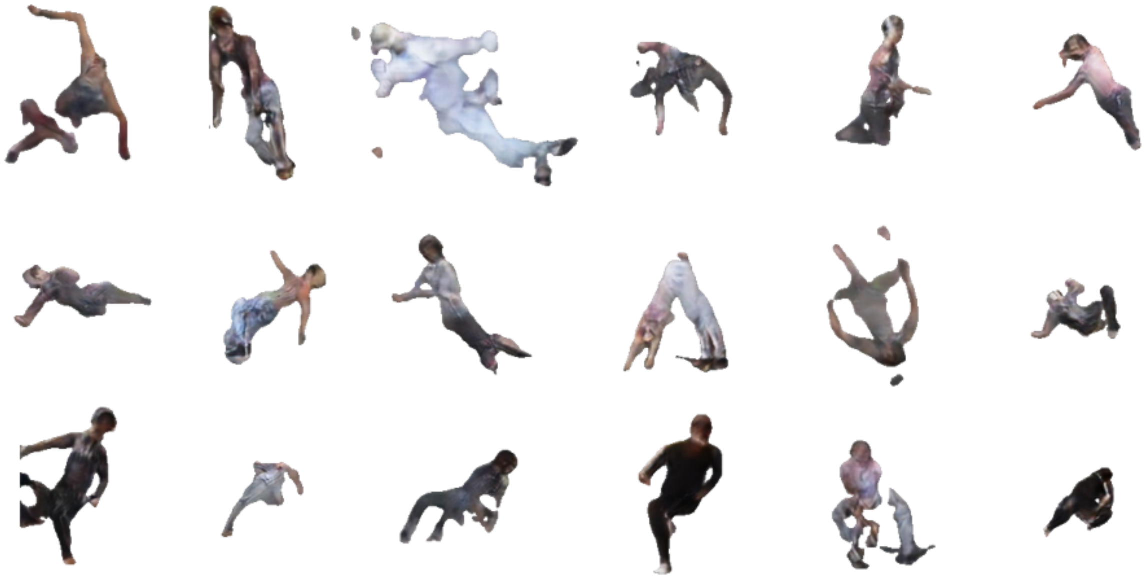


Figure A.7: Example of bad outputs from Pose-Warp GAN

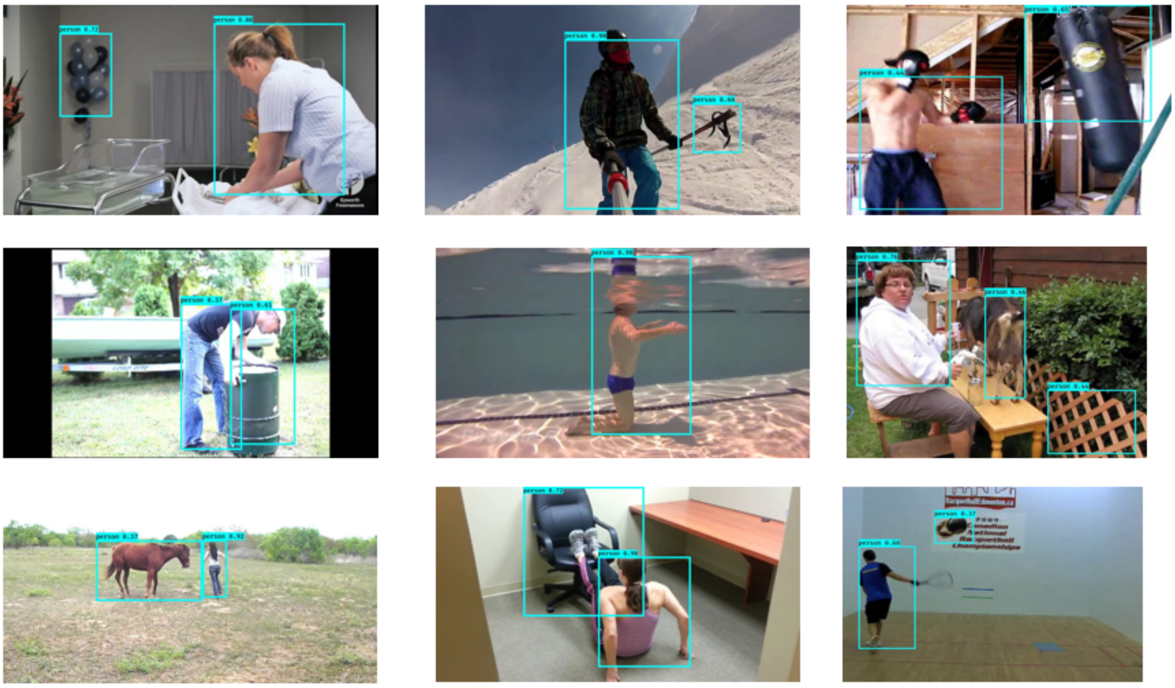


Figure A.8: Examples of False positives picked up by YOLOv3 trained on noisy Augmented dataset

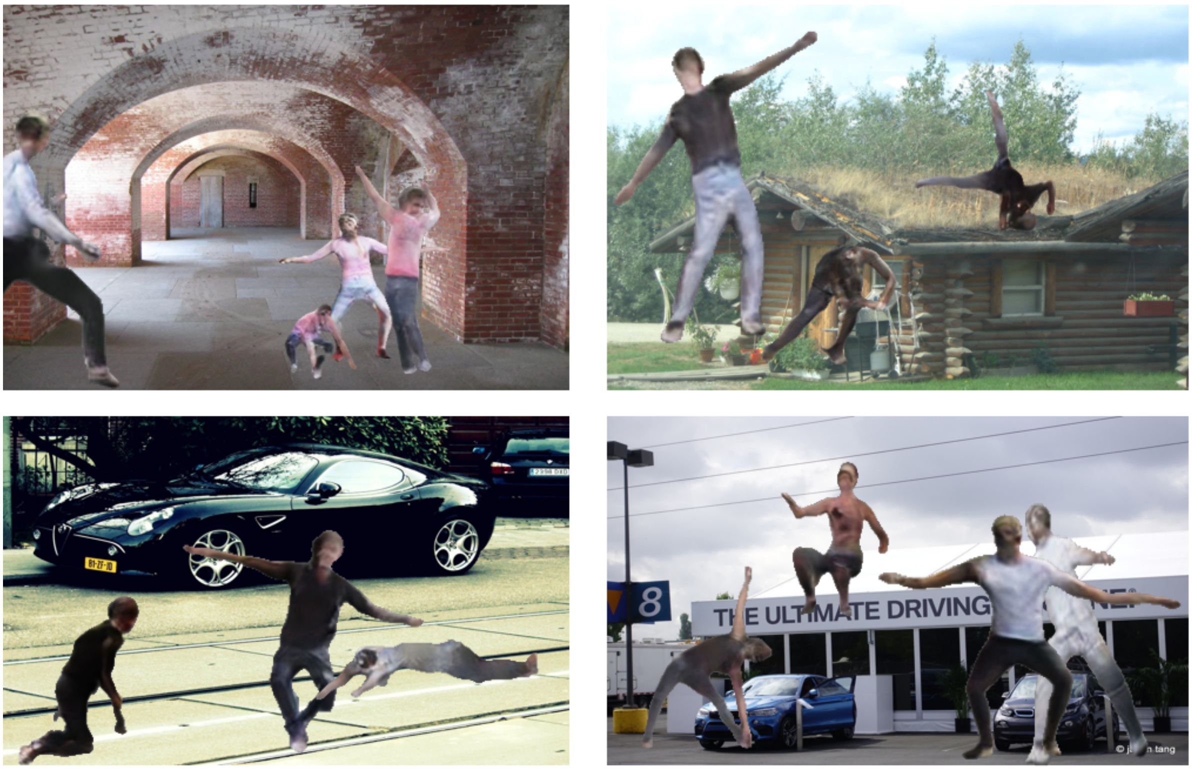


Figure A.9: Augmented Images containing multiple characters resulting in desirable occlusions

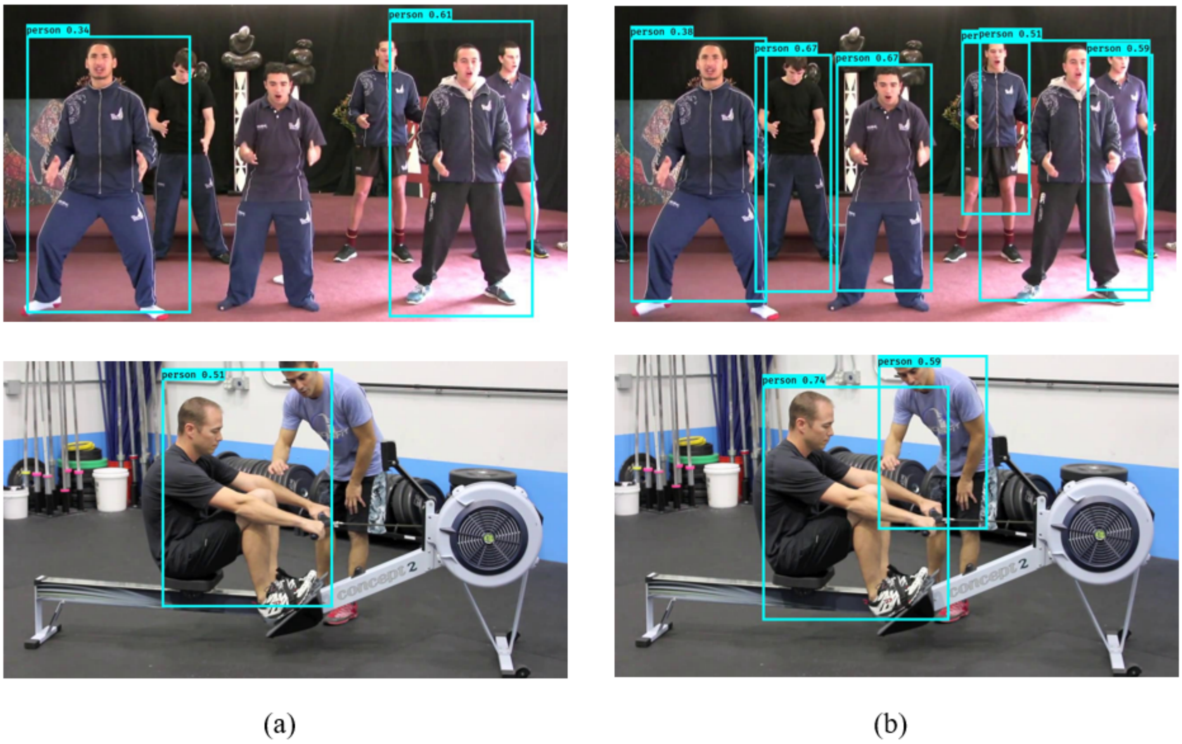


Figure A.10: (a)Detections picked when YOLOv3 trained on dataset with single character per image. (b) Detections picked when YOLOv3 trained on dataset with multiple characters per image.