Advanced Computer Vision Coursework

Anonymous author: chpf93

1 Human Feature Analysis

1.1 Human Patch Extraction (*100 words*)

I extract and save the frames using ffmpeg. Following this, I apply a pre-trained MaskR-CNN [1] model to identify the objects in each frame. I use an improved version of MaskR-CNN [2] with a Vision Transformer [3] backbone instead of ResNet [4] from PyTorch [5]. I discard all classes except the person class and remove all instances that have a confidence score below 0.965. I use a pre-trained model due to the unlabelled dataset making training infeasible.

Using the predicted bounding box from MaskR-CNN, I crop and save over 55000 human patches (of which there may be multiple per frame) for later processing.

Figure 1: Hand-picked examples of human patches from the Game domain videos.

Figure 2: Hand-picked examples of human patches from the Movie domain videos.

1.2 Classification (*99 words*)

I use OpenPose [6] to extract the pose to a JSON file (per patch). I analyse the pose data and classify according to the visibility of groups of joints, these groups are: head, torso and arms, hips, legs, and feet.

For full body, I calculate the angle between the knees and hip joints. If either leg angle is less than 50° then they are classified as sitting.

To evaluate, I label 150 random patches from each domain, the results are:

	Game Domain	Movie Domain
Accuracy	74.7%	78.4%

Table 1: Pose classification accuracy

Classification is successful but issues such as camera distance, the angle people are facing, and joint grouping choices, impact accuracy.

1.3 Training Data Selection (*100 words*)

I discard all human patches that are facing away from the camera and those that are in the 'Other' category. These samples are not as useful for learning human features based on my experimentation. The 'Other' category contains incomplete poses and largely occluded humans.

I group consecutive human patches and sample 5% of images from each group to reduce bias based on screen time. All images smaller than 64x64 pixels are discarded. Resizing them will degrade their quality substantially.

I resize the remainder to 128x128 pixels and sample a training set of 1200 images while keeping 250 images for testing.

1.4 Performance (*100 words*)

The bounding boxes from MaskR-CNN [1] could be problematic. Firstly, the threshold value for the confidence of the patch presents a trade-off between true positives and false positives. Additional resources, with labelled data, would enable optimisation of the threshold value. Secondly, it is possible that multiple humans appear in a single bounding box causing inaccuracy during pose extraction. The background could be separated out, but this may impact the style model's ability to differentiate between background and foreground information by itself.

Additionally, finetuning the MaskR-CNN model on labelled data specific to the datasets could improve the accuracy with more computational resources.

2 Real-world Application

2.1 Image Model Deployment (*250 words*)

I implement CycleGAN [7] and train it with 1200 images per domain resized to 128x128. I modify the generator ResNet [4] to reduce the checkboarding effect by substituting the transposed convolutions [8] with upsampling layers and regular convolutions [9]. I also experiment with PixelShuffle [10] but find the upsampling layers work better.

I train for 100 epochs to balance mode collapse [11] with style transfer between the two domains X and Y within my resource constraints. CycleGAN works by training two generators, $G : X \rightarrow Y$ and $F : Y \rightarrow X$, and two 70x70 patch discriminators [12], D_X, D_Y, using L1 and MSE loss while enforcing a cycle-consistency loss by minimising:

Figure 3: Example of Checkboarding Artifacts. Left: Upsample. Right: PixelShuffle.

$$L_{GEN}(G, D, x) = \|D(G(x)) - 1\|_2^2 \tag{1}$$
$$L_{DIS}(G, D, x) = \|D(G(x)) - 0\|_2^2 + \|D(x) - 1\|_2^2 \tag{2}$$
$$L_{CYC}(G, F, y) = \|G(F(y)) - y\|_1 \tag{3}$$
$$L_{IDT}(G, y) = \|G(y) - y\|_1 \tag{4}$$
$$\begin{aligned} L(G, F, D_Y, D_X, x, y) = {} & L_{GEN}(G, D_Y, x) + L_{GEN}(F, D_X, y) + L_{DIS}(G, D_Y, x) + L_{DIS}(F, D_X, y) \\ & + \lambda L_{CYC}(G, y) + \lambda L_{CYC}(F, x) + \gamma L_{IDT}(G, y) + \gamma L_{IDT}(F, x) \end{aligned} \tag{5}$$

I use the Frechet Inception Distance (FID) [13] and Kernel Inception Distance (KID) [14] to compare distributions of images using feature vectors from a pre-trained Inception [15] model to evaluate. I transfer the style of the other domain onto the test datasets (e.g. Game→Movie) and use FID and KID with the matching domains (e.g. Game→Movie and Movie) [16]:

Metric	Game and Movie→Game	Movie and Game→Movie
FID [13] (0 is best)	196.0	203.8
KID [14] (0 is best)	0.047	0.065

Table 2: FID and KID scores

I cycle the style transferred data back to the original domain (e.g. Game→Movie→Game) and use MaskR-CNN [1] to generate segmentation maps for all classes from COCO [17] for both datasets. These segmentation maps are compared to measure per-pixel accuracy by adapting the FCN score [18]:

Metric	Game and Game→Movie→Game	Movie and Movie→Game→Movie
Per-Pixel Accuracy [13]	0.177	0.315

Table 3: Per-pixel segmentation map accuracy, close to 1 is best

One issue with the outputs is failure in regions of dark colours likely caused by insufficient dataset diversity despite image augmentation of the training data. Issues also occur with blur (most likely due to the input size) and oversaturation of foreground colours while retaining (and worsening) washed out background colours:

Figure 4: CycleGAN success cases. Top row: Movie→Game. Bottom row: Game→Movie.

Figure 5: CycleGAN failure cases. Top row: Movie→Game. Bottom row: Game→Movie.

2.2 Temporal Enhancement (*244 words*)

I extract each frame from the test video and apply the generator to transfer the test video to the game domain style. Evidence of style transfer exists but as highlighted previously, there remains clear issues with worsened, washed-out colouring and artifacts caused by the L1 loss encouraging blurring when the edge is difficult to define [12]. Checkerboarding is noticeably reduced in areas without edges. Some parts of the video are good, the close up of the man in the tent has been brightened yet facial features still remain clearly defined.

I utilise the temporal information of the videos by extending CycleGAN [7] to RecycleGAN [19]. Using two U-Net [20] predictor networks, $P_X : (X, X) \to X$ and $P_Y : (Y, Y) \to Y$, that take two frames and attempt to predict the next, it further constrains the generators and discriminators by minimising the loss of the prediction using generated and real samples.

I train for 100 epochs using 256x256 images in the form of triplets representing 3 consecutive human patches. Using similar techniques to question 1, with some local improvements such as sampling higher resolution images, I extract 600 triplets for each domain (1800 total images). The results are substantially better. Artifacts at the edges of objects are reduced (although they still persist) and the washed-out colour effect from CycleGAN is lessened. Failure in dark colour regions persists but is decreased. Again, this is likely due to the dataset diversity despite image augmentation. I evaluate using the same metrics as CycleGAN with substantial improvement and show 12 frame comparisons:

Metric	Game and Movie→Game	Movie and Game→Movie
FID [13] (0 is best)	179.0	172.3
KID [14] (0 is best)	0.048	0.041

Table 4: FID and KID scores

Metric	Game and Game→Movie→Game	Movie and Movie→Game→Movie
Per-Pixel Accuracy [13]	0.710	0.547

Table 5: Per-pixel segmentation map accuracy, close to 1 is best

Figure 6: Frame 1 Comparison. Left: Original. Middle: CycleGAN. Right: RecycleGAN.

Figure 7: Frame 2 Comparison. Left: Original. Middle: CycleGAN. Right: RecycleGAN.

Figure 8: Frame 3 Comparison. Left: Original. Middle: CycleGAN. Right: RecycleGAN.

Figure 9: Frame 4 Comparison. Left: Original. Middle: CycleGAN. Right: RecycleGAN.

Figure 10: Frame 5 Comparison. Left: Original. Middle: CycleGAN. Right: RecycleGAN.

Figure 11: Frame 6 Comparison. Left: Original. Middle: CycleGAN. Right: RecycleGAN.

Figure 12: Frame 7 Comparison. Left: Original. Middle: CycleGAN. Right: RecycleGAN.

Figure 13: Frame 8 Comparison. Left: Original. Middle: CycleGAN. Right: RecycleGAN.

Figure 14: Frame 9 Comparison. Left: Original. Middle: CycleGAN. Right: RecycleGAN.

Figure 15: Frame 10 Comparison. Left: Original. Middle: CycleGAN. Right: RecycleGAN.

Figure 16: Frame 11 Comparison. Left: Original. Middle: CycleGAN. Right: RecycleGAN.

Figure 17: Frame 12 Comparison. Left: Original. Middle: CycleGAN. Right: RecycleGAN.

References

[1] Kaiming He et al. "Mask R-CNN". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2961–2969.

[2] Yanghao Li et al. "Benchmarking detection transfer learning with vision transformers". In: *arXiv preprint arXiv:2111.11429* (2021).

[3] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[4] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[5] Adam Paszke et al. "PyTorch: An imperative style, high-performance deep learning library". In: *Advances in neural information processing systems* 32 (2019).

[6] 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.

[7] 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.

[8] Matthew D Zeiler et al. "Deconvolutional networks". In: *2010 IEEE Computer Society Conference on computer vision and pattern recognition*. IEEE. 2010, pp. 2528–2535.

[9] Augustus Odena, Vincent Dumoulin, and Chris Olah. "Deconvolution and Checkerboard Artifacts". In: *Distill* (2016). DOI: 10.23915/distill.00003. URL: http://distill.pub/2016/deconv-checkerboard.

[10] Wenzhe Shi et al. "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1874–1883.

[11] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[12] 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.

[13] Martin Heusel et al. "Gans trained by a two time-scale update rule converge to a local nash equilibrium". In: *Advances in neural information processing systems* 30 (2017).

[14] Mikołaj Bińkowski et al. "Demystifying mmd gans". In: *arXiv preprint arXiv:1801.01401* (2018).

[15] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[16] Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. "On aliased resizing and surprising subtleties in gan evaluation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 11410–11420.

[17] Tsung-Yi Lin et al. "Microsoft COCO: Common objects in context". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.

[18] Marius Cordts et al. "The Cityscapes Dataset for Semantic Urban Scene Understanding". In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[19] Aayush Bansal et al. "Recycle-gan: Unsupervised video retargeting". In: *Proceedings of the European conference on computer vision (ECCV)*. 2018, pp. 119–135.

[20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.