Advanced Computer Vision Coursework

Word count: 1499

Introduction

This work aims to take advantage of the ever-converging movie and game industries to explore using generative computer vision to make games more “movie-like”. This project tailors a state-of-the-art unpaired image-to-image generative network for converting full frames and faces between game and movie domains.

Preprocessing

For each domain, 1000 identically distributed frames were trialed for extraction. Since all videos contain anomalous scenes e.g., game menus, black screens, and fully-occluded scenes, as shown in Figure 1, only frames with at least one face detected by the Dlib face detector [1] were extracted. Furthermore, these detected faces were cropped and extracted to form the face dataset, yielding 443 game frames with 483 faces, and 307 movie frames with 342 faces.

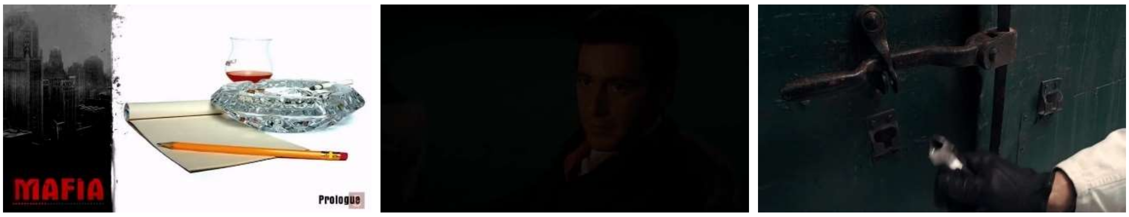

Figure 1. Left-to-right: Game menu (*Mafia*); black screen (*The Godfather*); occluded scene (*The Irishman*)

This dataset was split into train, validation, and testing sets. The validation and testing sets genuinely test whether the model overfit the data by taking advantage of video’s ordered structure. As the frames are unshuffled prior to being split, the first 70% of the video forms the training dataset, with the next 20% and 10% assigned to validation and testing. Since the frames remained ordered, while these sets may contain the same characters, they will not be from the same scene and therefore will be under lighting conditions, be posed differently, or dressed differently.

Network Architecture

To handle the complex task of style transfer over the whole scene, an attention-guided GAN was applied. This architecture type was chosen so that the attention mechanism would learn the important areas of the scene to be transformed and outperform other CycleGAN techniques by focussing on these areas.

The original architecture iterates on the original CycleGAN by introducing a fourth output channel, an attention mask of the image [2]. The image output is formed by combining the original image with the generated image, with areas of high attention drawing from the generated image and vice versa [2].

We reduce the input image size to 128x128 to improve training speed while retaining global scene information. The images are normalised between -1 and 1, therefore any outputs first go through the Tanh activation function, whereas the original used unnormalized data and {0,1}-Sigmoid functions [2].

It also differs from the original by removing a loss function, opting for three generator loss functions: validation loss, identity loss, and reconstruction loss. Validation loss is the L2 loss between the domain discriminator's prediction of an image synthesised to that domain and a tensor of ones, its purpose is to ensure that the model learns to produce images deemed to be in the new domain. Identity loss is the L1 loss of the synthesised image and the original, causing the model to learn to produce outputs like the original. The final reconstruction loss takes the L1 loss of the original image and the synthesised image after being put through both generators. This reconstruction loss function ensures the cyclical nature of the CycleGAN and encourages the model to learn that an image converted back to the original domain should be the same as the original image. The Attention Loss described in the paper [2] was omitted as, through experimentation, it had the opposite effect than its designed purpose, causing the attention masks to quickly saturate to 1. Omitting this function and retaining the original CycleGAN generator loss functions trained both the attention mask, without saturating it, and the image synthesiser.

The discriminator loss functions remain unchanged, both using L2 loss to predict whether a generated image is a real or fake. Furthermore, the discriminator was not overtrained, as shown in Figure 2, the generators and discriminators were well balanced.

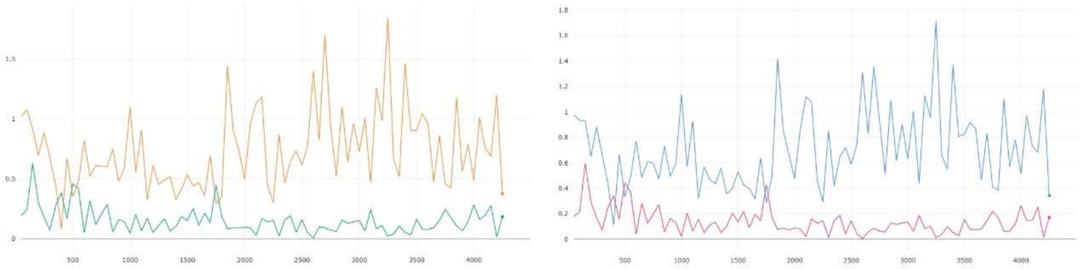

Figure 2. Left-to-right: Game-to-movie generator loss (orange) vs discriminator loss (green); Movie-to-game generator loss (blue) vs discriminator loss (pink).

The learning rate of the generator was set to 0.002 and 0.0002 for the discriminator. After a series of experiments, it was determined that these values resulted in the most balanced generator/discriminator loss. The size of the batches was set to one so that instance normalisation could be implemented, and the gradients could be updated after each frame – allowing the model to build on its knowledge from previous frames in the epoch extracted from the same scene.

Frame-to-Frame Performance

The model performed poorly on this full-scene dataset. It performed best on simple scenes, with a single character taking up the foreground, examples are shown in Figure 3 and Figure 4.

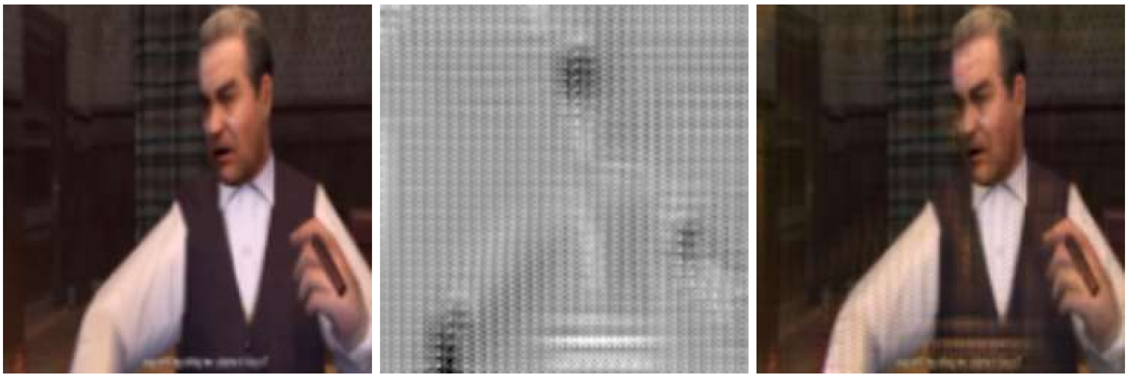

Figure 3. Simple game scene. Left-to-right: Original; attention mask; synthesised

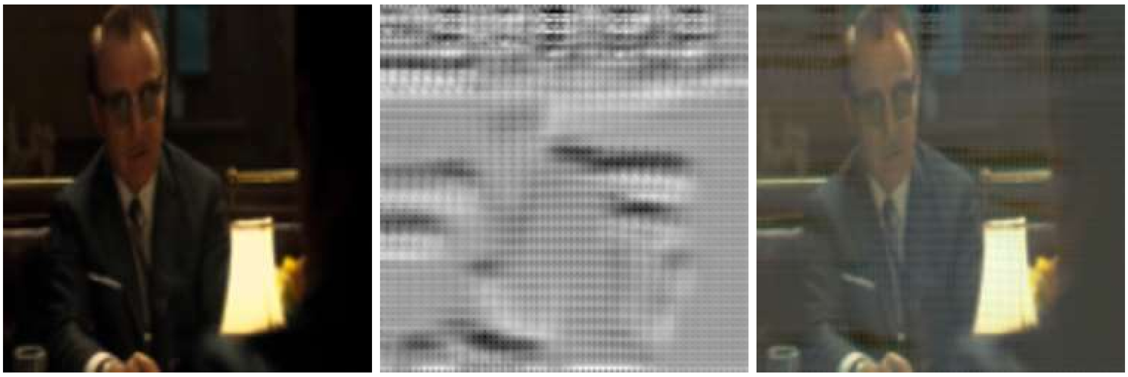

Figure 4. Simple movie scene. Left-to-right: Original; attention mask; synthesised

In both examples noise is introduced from the mask being constructed poorly. While an outline of the subject is somewhat visible, perhaps indicating that with more training it would improve, the mask is still letting through too much incorrect information. This is unfortunately an unavoidable problem of the dataset. Since the subject of each frame is so visually different and posed differently, it is hard for the attention network to suitably learn to find the subject. This is not a problem on the simpler problems it has been previously applied to e.g., horse-to-zebra where both animals are posed consistently and differ very slightly in appearance. Despite this, the generally expected result of the game images darkening while the movie images lighten is present here. This is expected as most scenes from the movies have more saturated and darker lighting.

In more complicated scenes, the same problem is present, with the mask indicating the presence of the subjects but failing to adequately filter out the background. This is shown in Figure 5 and Figure 6.

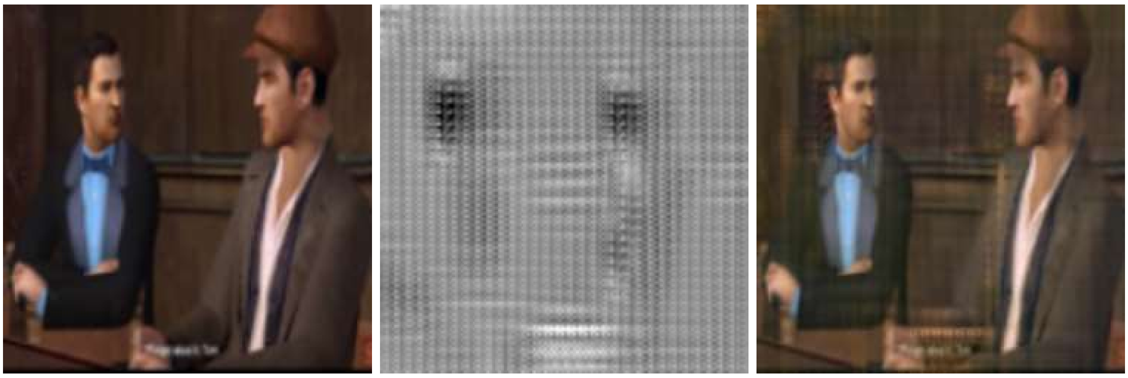

Figure 5. Multi-subject game scene.

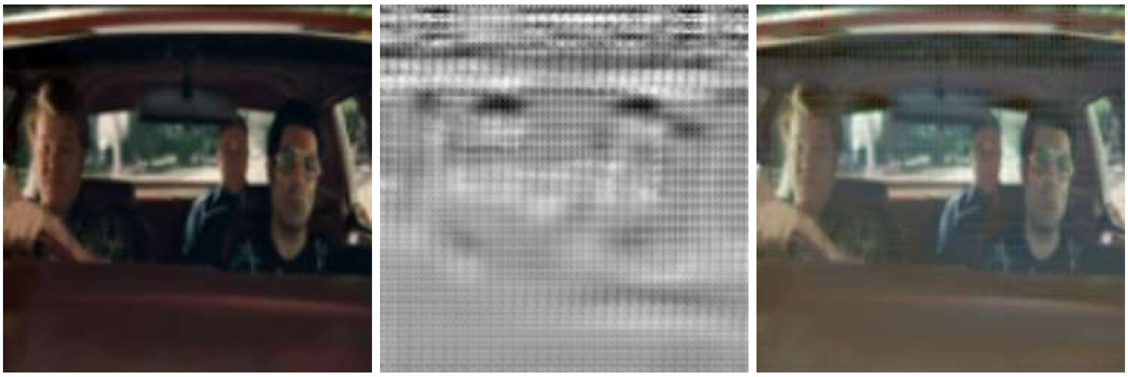

Figure 6. Multi-subject movie scene.

However, for some scenes no meaningful mask was obtained. Examples shown in Figure 7 and Figure 8.

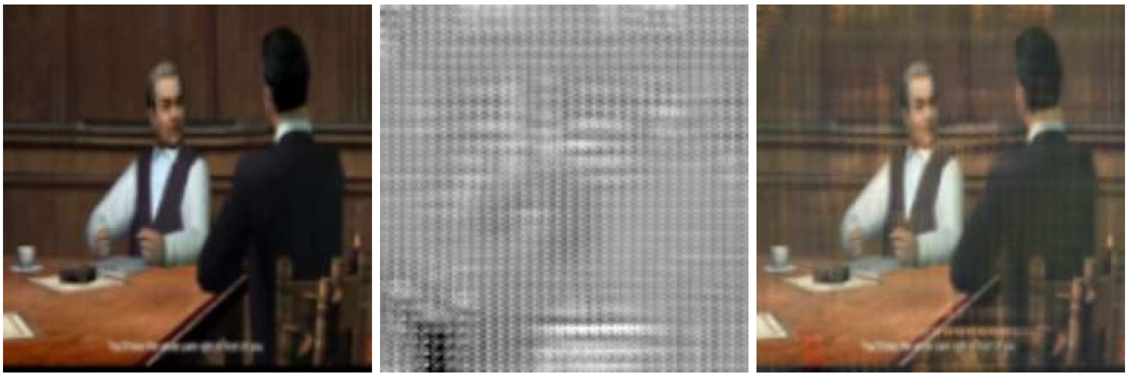

Figure 7. Failed game scene.

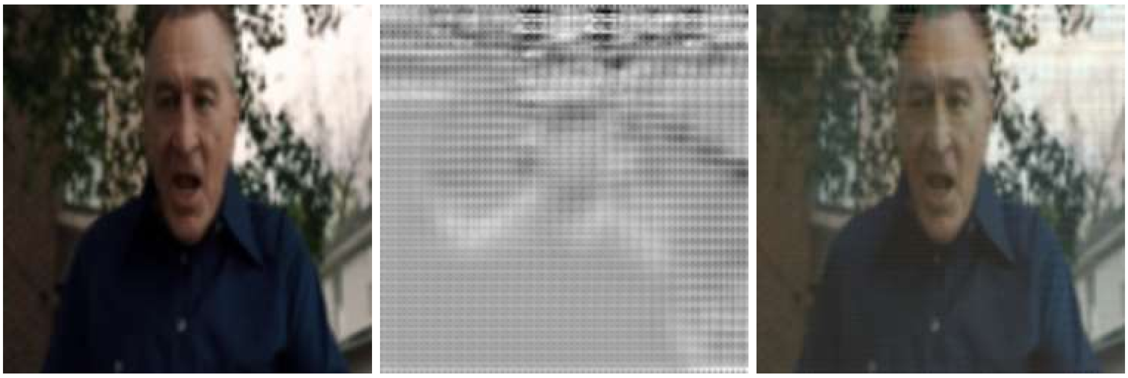

Figure 8. Failed movie scene.

Face-to-Face Performance

As expected, the model performed better when the dataset was limited to faces. This is because when seeing faces the model can learn the important facial features. This allowed the model to generate much more precise attention masks in the same time span, leading to more accurate results. The best results are showcased in Figure 9 and Figure 10.

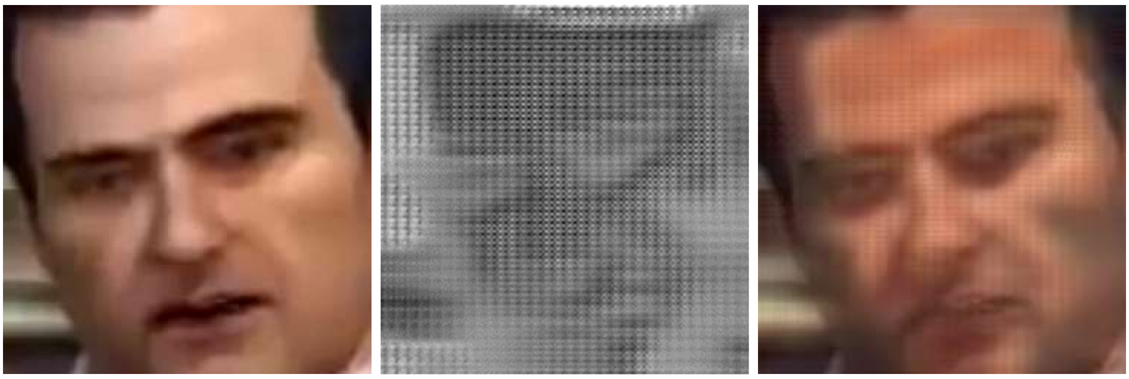

Figure 9. Successful game scene.

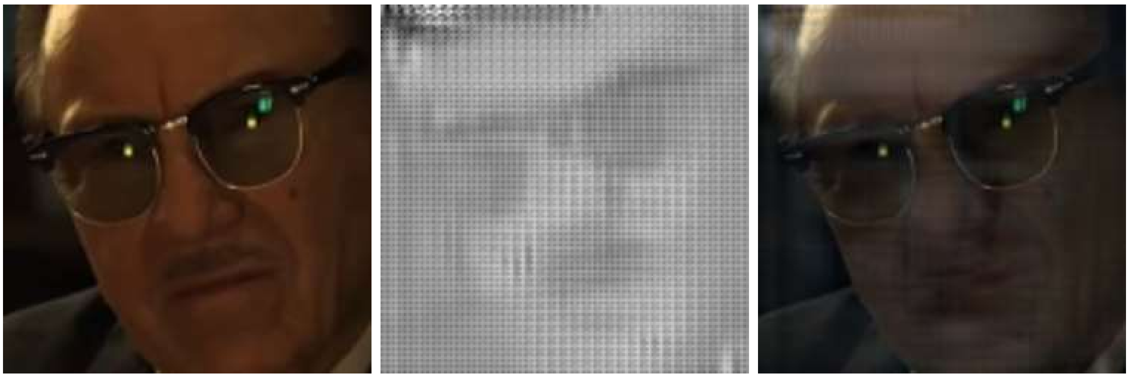

Figure 10. Successful movie scene.

In these examples, not only is the expected saturation/desaturation present, but due to the precise attention mask, the model seeks to avoid the eye, hair, and lip colour, most strongly affecting the colour of the skin. This allows for a very strong synthesis of the movie-to-game face, which thanks to the mask, produces a glasses-agnostic transformation. The game-to-movie masks are generally weaker, causing a patchy appearance to the recolouring of the face. A severe example of this is shown in Figure 11.

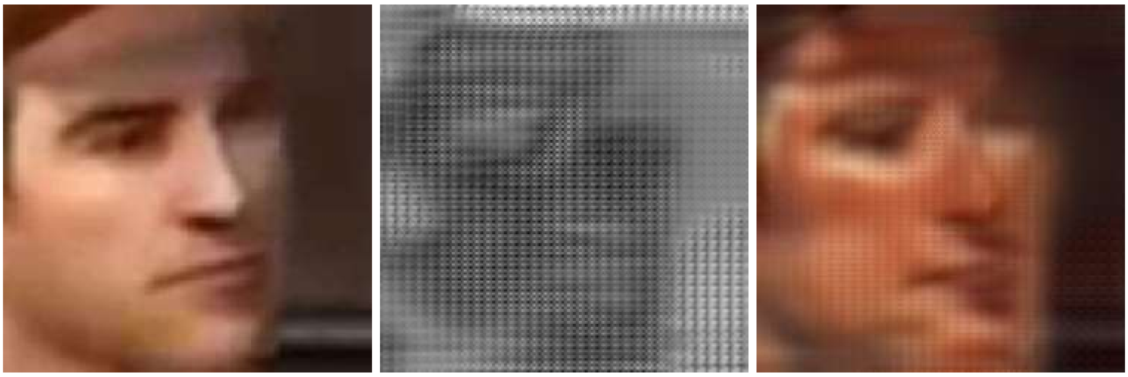

Figure 11. Severe example of game-to-movie "patchy" face colour.

Data Augmentation

Data augmentation was introduced in attempt to artificially expand the face dataset. Random crop and random horizontal flip augmentations were selected as it would change the pose of the faces and how much of the face is shown to the network, without modifying the colour or posing unnaturally.

These augmentations allowed the mask to train faster and more precisely, solving the patchy colouring problem and producing more accurate colourings.

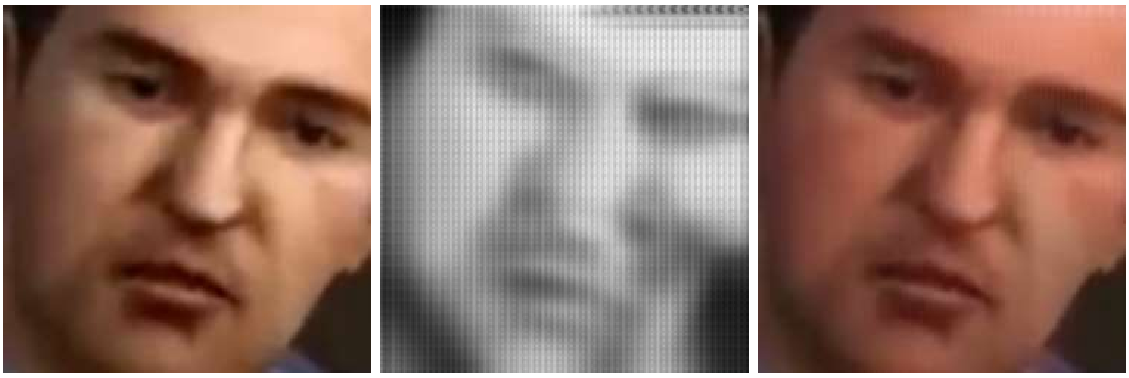

Figure 12. Successful Patchy-Colouring Mitigation

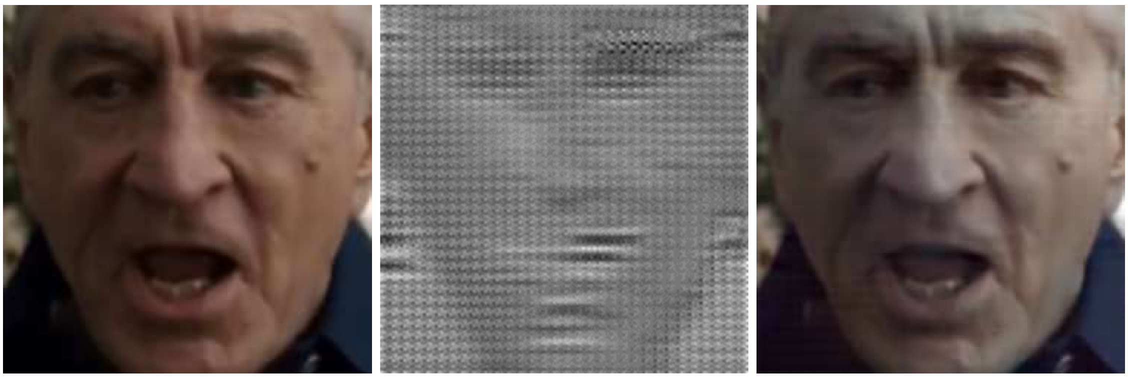

Figure 13. Successful Movie Scene

One weakness of this new model is that it can be too severe when recolouring, causing red or ghostly-white faces as shown in Figure 14 and Figure 15.

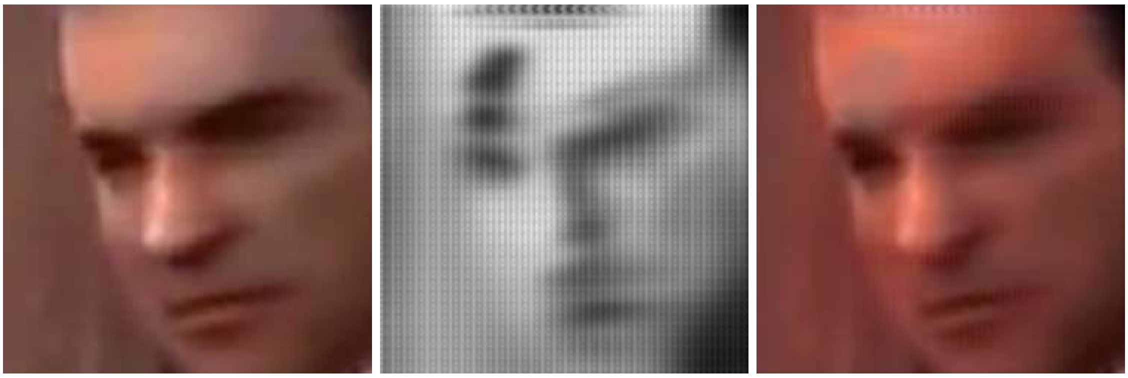

Figure 14. Red-faced game scene

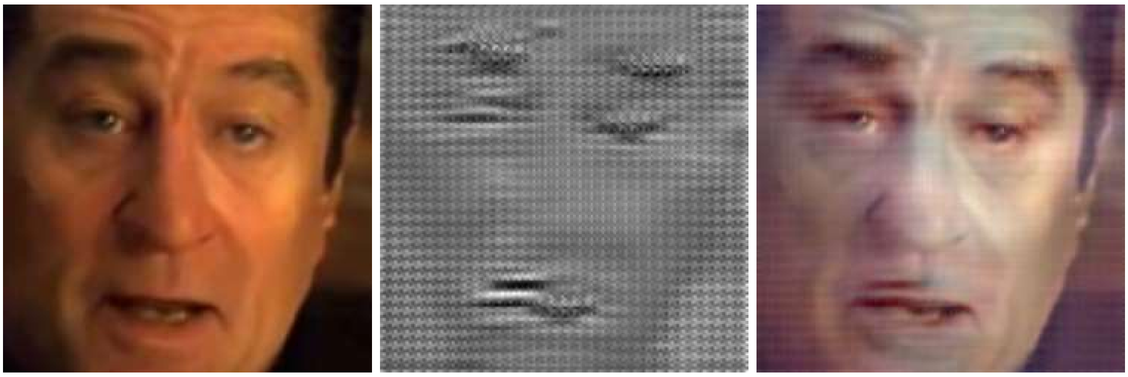

Figure 15. White-faced movie example

Dataset Patterns

In addition to the colour palettes discussed, the game scenes are synthesised by computer graphics, whereas the movie scenes are videos of real people. As a result, the movie characters' faces are far more detailed, with smoother edges, than the low-polygon shapes used to create characters in the game. As a result, the games' faces are blockier than the movie faces, but this effect has not been synthesised by this work, only the colouring.

Video synthesis

Applying the augmented faces model to a series of frames yields a video. Few changes to the original video are made, however the model successfully pays attention to the face, darkening it consistently with the other examples. In addition, artifacts are introduced to the scene, presumably due to not being exposed to any prior examples of the background of a scene.

The video suffers with stability and temporal consistency, with the whole scene flashing when there is no face, causing the attention mask to become confused. In addition, when the face is shown, its colour also fluctuates as the character walks.

To improve temporal consistency, recycle loss can be implemented. This works by first predicting the next frame in the video based on the previous frames [3]. The recycle loss function, Equation 1, measures how accurate the prediction was, hence producing more accurate predictions of a natural next frame, preserving consistency [3].

Equation 1. Recycle Loss

$$L_r(G_x, G_y, P_y) = \sum_t \left|\left|x_{t+1} - G_x\left(P_y\left(G_y(x_{1:t})\right)\right)\right|\right|^2$$

i.e., the difference between the true frame and the predicted frame

This system would learn the how to naturally form a video alongside how to map between the domains. However, when unpredictable actions occur, the model will suffer. Since this loss function is only applied during training, in a real-time application, the overhead will be performed prior, introducing little additional latency.

Rendering Pipeline Extension

If the model had access to the whole rendering pipeline, individual textures could be synthesised in advance, allowing for fast rendering of the game models during gameplay. The audio could be synthesised by similar CycleGANs trained on audio spectrograms [4]. There are various 3D-shape reconstruction techniques [5] which could be applied to generate 3D models of objects from movies. By comparing lighting with an OLAT dataset [6], the lighting of the movie can be gleaned and reproduced.

References

[1] Dlib, "Image Processing," [Online]. Available: http://dlib.net/imaging.html#get_frontal_face_detector. [Accessed 31 May 2021].

[2] H. Tang, X. Dan, S. Nicu and Y. Yan, "Attention-guided generative adversarial networks for unsupervised image-to-image translation," *2019 International Joint Conference on Neural Networks (IJCNN),* pp. 1-8, 2019.

[3] A. Basnal, S. Ma, D. Ramanan and Y. Sheikh, "Recycle-gan: Unsupervised video retargeting," *Proceedings of the European conference on computer vision (ECCV),* pp. 119-134, 2018.

[4] M. Pasini, “Voice Translation and Audio Style Transfer with GANs,” 6 November 2019. [Online]. Available: https://towardsdatascience.com/voice-translation-and-audio-style-transfer-with-gans-b63d58f61854. [Accessed 31 May 2021].

[5] N. Wang, et al. “Pixel2mesh: Generating 3d mesh models from single rgb images,” *Proceedings of the European Conference on Computer Vision (ECCV),* pp. 52-67, 2018.

[6] C. Guo, et al. “The Relightables: Volumetric Performance Capture of Humans with Realistic Relighting,” *ACM Transactions on Graphics,* vol. 38, no. 6, pp. 1-19, 2019.

Disclaimer

The model was trained for 50 epochs (~15 mins) on a Tesla P100-PCIE-16GB GPU.