Introduction:

Style transfer is a well researched field in Computer Vision. In this paper, Contrastive Unpaired Translation [4] is applied to footage from the Mafia video game and various live-action films, evaluating its effectiveness through observation. Two versions are implemented: a network that translates frame-to-frame, and one that translates face-to-face. Finally, a sequence of frames are processed, and the network's temporal consistency is evaluated.

Method:

The test system uses a Ryzen 7 3700X CPU and a GTX 1080 GPU. Models were trained for 75 epochs, unless stated otherwise, and inputs are normalized.

1.1 To gather the dataset, 1000 frames were selected such that they were temporally spaced evenly apart. With consideration to overfitting, the frames were split such that the first three-quarters of the videos are used for training, and the final quarter is used for verification. The three movies are treated as a single movie, and the dataset is gathered similarly. Overall, it is hard to mitigate overfitting due to the limited size of the dataset.

2.1 This report tests the CUT model, composed of a ResNet-based generator and a PatchGAN discriminator. Inputs are resized to 256x144 RGB images to reduce the training time. A batch size of 1 is used, and instance normalization [1] is applied. A learning rate of 0.0002 and beta-values of 0.5, 0.999 are used for the Adam optimizer. Dropout layers were applied to the residual blocks of the generator, to mitigate overfitting.
Two loss functions are tested: the LSGAN least-squares loss and a non-saturating MinMax loss. Fig.1 shows that the unsaturated loss results in sharper images, but the colouring is more different from the original. The LSGAN loss has red artifacting on the left sides. On the other hand, the LSGAN loss provides much more stable training.

Fig.1: Comparison of GAN loss functions. Top is ground truth, middle is LSGAN loss, bottom is Nonsaturating Minimax Loss

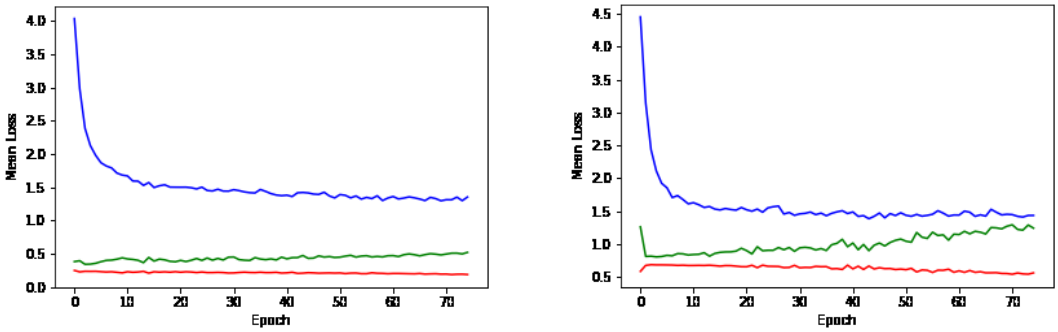

Fig.2: Comparison of GAN loss graphs. Left is LSGAN loss, right is Nonsaturating Minimax Loss

[4] also use two λ-parameters to adjust the weighting of the GAN loss versus the contrastive loss, when calculating the generator loss. Biasing the weights towards the contrastive loss produces the sharpest results. However, the style doesn't change noticeably. For example, in the middle column of fig.3, the outline of the gun has less of a stair-stepped effect in the 2nd row as opposed to the 4th row. Increasing the GAN weights resulted in more colour distortion, and seems less sharp.


Fig.3: Comparison of λ values. Top to bottom: ground truth, $\lambda_{GAN} = 1$ and $\lambda_{NCE} = 1$, $\lambda_{GAN} = 10$ and $\lambda_{NCE} = 1$, $\lambda_{GAN} = 1$ and $\lambda_{NCE} = 10$

Results show that after 400 epochs, even the movie-to-movie translation of the generator drops in performance, indicating possible overfitting in the model. However, after ~60 epochs, results are acceptable, clarity wise. With regards to style-transfer, it is clear to see that the colouring of the outputs are clearly different. However, textures and models show little to no change.

Fig.4: Game-to-movie, trained for 400 epochs. Left half shows Game-to-Movie, right half shows Movie-to-Movie. Top 2 rows are after ~60 epochs, bottom 2 rows are after 400.

In the opposite direction, game-to-game identity translation is maintained after 400 epochs. However, movie-to-game results are very distorted.

Fig.5: Movie-to-game, trained for 400 epochs. Left half shows Game-to-Movie, right half shows Movie-to-Movie. Top 2 rows are after ~60 epochs, bottom 2 rows are after 400.

3.1 To adapt the dataset for a face-to-face model, we use dlib's facial-landmark detection to find faces in the game footage, extracting them as before, with equally sized datasets. For real faces, the Flickr-Faces-HQ dataset is used instead, giving a larger variety of faces to train on.

3.2 The model configuration is kept largely the same as previously. The input size was changed to be 192x192 pixels, to match the pixel count of the frame-to-frame model. After ~60 epochs, the style transfer is clearly noticeable, as in fig.6. However, it quickly overfits, and the outputs become distorted.

In comparison to the frame-to-frame model, the output image is generally brighter, and the image generally looks to be lit in a more realistic manner.
With regards to style transfer, the face-to-face model clearly succeeds at one point. For example the middle column of fig.6 makes the male game-character look more real, but inaccurately turns the character more feminine. Otherwise, it is clear to see that the middle column of fig.6 produces the most realistic results. With regards to the rightmost column, the eyes and mouth become distorted, but features like teeth were clearly transferred. In contrast, the leftmost column shows only a change in skin colour.
Overall, there seems to be some overfitting - the Flickr-Faces-HQ dataset contains many portraits of people smiling, hence teeth is often shown. The rightmost column is an example of the overfitting resulting in abnormal features.

Fig.6: Side-by-side comparison of frame-to-frame (left) vs. face-to-face (middle and right). Middle column is taken from ~60 epochs, and far right is after 75. Top: Ground Truth, Bottom: Output

3.3 Random crops, horizontal flips, rotations and colour jittering were applied to the dataset. This helps mitigate overfitting, by increasing the ‘diversity’ of the training data. Fig.7 shows the loss graphs for every configuration, and shows that colour-jittering made little difference. Horizontal flips reduce the rate of divergence in the GAN generator/discriminator losses, and random rotations resulted in the most stable training, with very little difference in GAN generator/discriminator loss, followed by random crops.

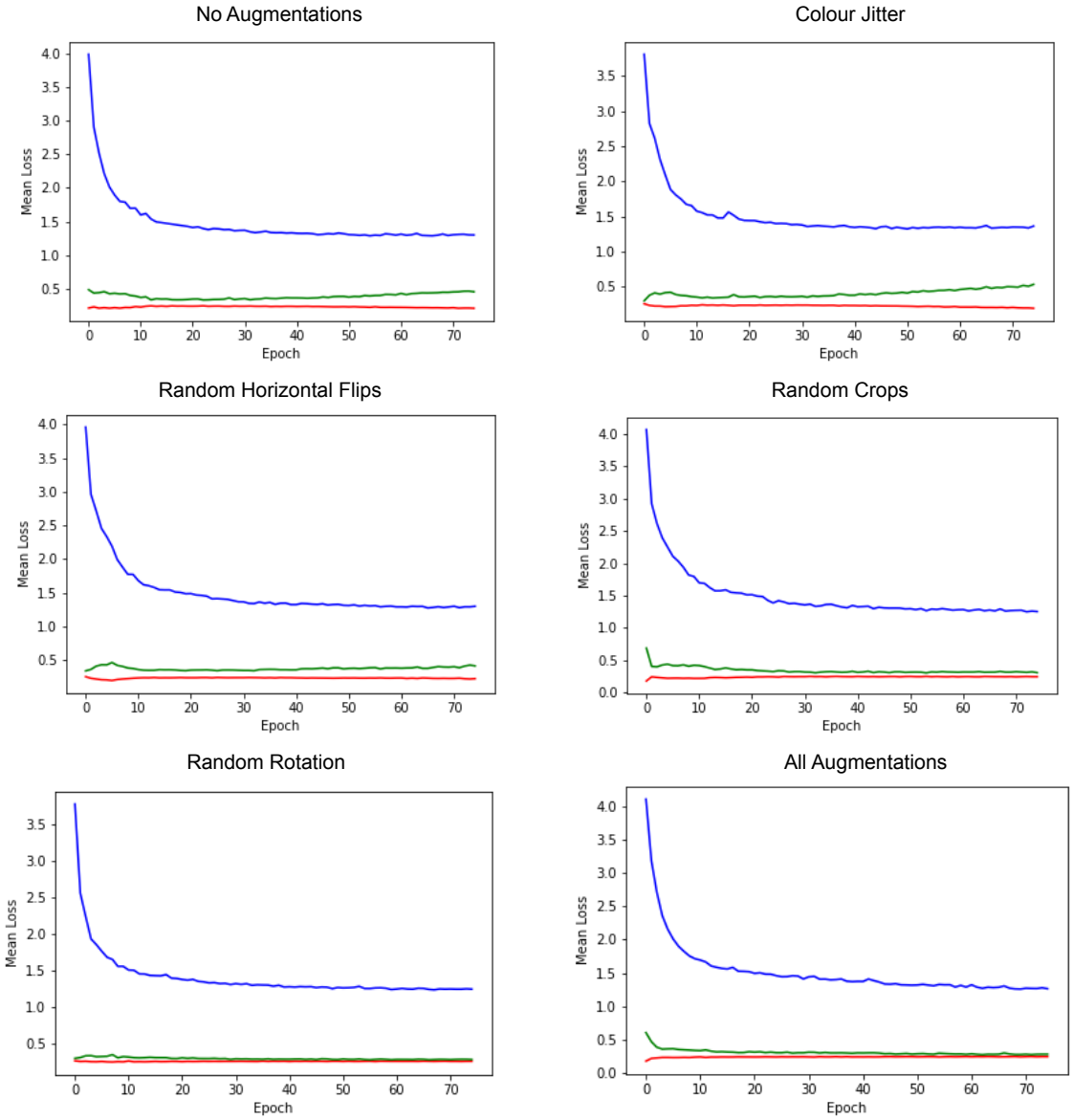

Fig.7: Loss plots for game-to-movie training on face datasets. Blue represents NCE loss, green represents Generator loss, and red represents Discriminator loss

However, looking at the outputs in fig.8, random rotations seem to add more noise. Hence, colour-jittering, horizontal flips and random crops are used in the final model.

Fig.8: Rows: Ground Truth, no augmentations, colour jitter, random horizontal flips, random crops, random rotation, all augmentations

The final model still seems to overfit slightly, once again struggling with the mouth. Often, there are situations where an eye gets confused for a mouth, commonly occurring when the input image is blurry/low-res.

Fig.9: Outputs from final model, trained for 400 epochs, Game-to-movie. Top is ground truth, bottom is output.

Fig.9 columns 2 and 4 show fairly convincing outputs, although column 4 seems to produce a more feminine result. Column 6 is an example where the teeth are improperly generated, being untrue to the ground truth. Overall, skin-colour is brightened, and facial contours become more defined. Furthermore, the lighting of the image is brighter, and the sclera of the eyes are whiter.

Fig.9: Outputs from final model, trained for 400 epochs, Movie-to-game. Top is ground truth, bottom is output.

In the opposite direction, fig.10 shows fairly good performance, largely smoothing the face and facial hair. Facial contours become less defined, and some faces become more masculine (e.g. column 5), which is expected to the entirely male cast of the game footage. Interestingly, the overall facial expression across all outputs becomes less happy, and teeth also isn't commonly shown in the outputs, regardless of the input.

To build on **3.2**, these results show that the game-dataset is lacking in a wide range of expressions, whereas the real-dataset is biased towards happy expressions. Similarly, teeth are regularly hidden in the game-dataset, whilst often visible in the movie-dataset. Finally, it can be said that either the real-dataset is slightly imbalanced towards females or the game characters look feminine.

3.4 Given a well-detailed input image, fig.9 shows that almost all facial features are identifiable, albeit at times the position of certain parts are more ambiguous (fig.9, column 6 shows misaligned eyes). With regards to realism of features, skin complexion is least realistic. In particular, how light reflects off the skin is poorly represented in the game. Similarly, facial hair is

very poorly represented in the game, having unrealistic texture, and often being mistaken as shadows on the face by the CUT models. A very minor issue can be found with the eyes being too small in the game. Finally, the shape of the face is typically well portrayed in the game.

4.1 To generate a video, we process the entire frame, followed by any faces, and faces are reapplied to the frame, thus the improved backgrounds of the frame-model are combined with the improved faces. In the output, flickering occurs, due to inconsistent colouring of consecutive frames. This inconsistency was particularly noticeable in the faces. To retain more detail, the input frame is split into four segments, processed separately, and then recombined. This resulted in very clear segmentation in the video, as indicated in fig.9.

To achieve temporal consistency, consecutive frames must be similar. Ruder et al. [4] incorporates optical flow, which allows us to identify movement between two frames, and achieves temporal stability in both short- and long-term, using separate loss functions described in their paper. Using a style-transfer algorithm such as CUT, we can generate low-resolution stylized images within reasonable time. However, predicting optical flow between two images is time consuming.

Hence, let f_i be the i^{th} frame. We add the following, and minimise the loss function:

$$loss_{temp} = MSE(W_{i-1 \Rightarrow i}(G(f_{i-1})), G(f_i))$$

where $W_{a \Rightarrow b}(x)$ is the function that warps an image x according to the optical flow between f_a and f_b, and G(x) is the stylized output for image x. Given f_a is not available, the loss is assumed to be 0. Following this, we use an AI upscaler, such as SRCNN by Dong et al. [1], trained on real-domain images, to recover ‘lost’ detail. By reducing the image size, and avoiding optical flow in real-time application, the method may be fast enough. However, this method may also result in severe overfitting, and struggle with sudden cuts between scenes.

Fig.9: Example frame from generated video

4.2 Given the entire rendering pipeline, one can improve the realism by:

a. Apply style transfer to textures instead of scenes
b. Increasing model detail with, for example, a network trained to increase the detail of a model, such as 3DFaceGAN [2]
c. Using a supersampling technology (e.g. NVIDIA's Deep Learning Super-Sampling) to upscale textures
d. Applying ray-tracing as a lighting model, instead of traditional rasterization

Bibliography

[1] D. Ulyanov, A. Vedaldi, V. Lempitsky. *Instance Normalization: The Missing Ingredient for Fast Stylization.* Nov 2017

[1] C. Dong, C.C. Loy, K. he, X. Tang. *Image Super-Resolution Using Deep Convolutional Neural Networks.* Jul 2015

[2] S. Moschoglou, S. Ploumpis, M.A. Nicolaou, *et al. 3DFaceGAN: Adversarial Nets for 3D Face Representation, Generation, and Translation.* May 2020

[3] T. Park, A.A. Efros, R. Zhang, J.Y. Zhu. *Contrastive Learning for Unpaired Image-to-Image Translation*. Aug 2020

[4] M. Ruder, A. Dosovitskiy, T. Brox. *Artistic style transfer for videos*. Oct 2016.