Neural Style Transfer for Character Motion Synthesis

MASTERS DISSERTATION

By

BRYCE JIRAN RAMGOVIND

School of Computer Science and Applied Mathematics

Supervised by: Prof. Turgay Çelik

University of the Witwatersrand

Johannesburg

FEBRUARY 16, 2021

Abstract

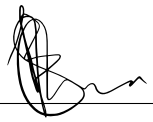
In the field of animation, three-dimensional characters are assigned skeletons, or character rigs, to simulate the naturalistic movements of their real-world counterparts. Animators manipulate the character rig to allow for smooth, naturalistic movement. This research proposes using a combination of deep learning techniques to learn the manifold of human character motion, trained on motion capture data. The motion manifold is represented by the latent space of the convolutional auto-encoder. This representation is subsequently transferred to any given locomotion, providing a high-quality motion sequence with a different style. This research builds upon the recent work of neural style transfer for images, but transports it to a new domain of human character motion. A qualitative study has been conducted on a group of participants to determine the success of the framework's ability in transferring style to a newly synthesised locomotion. Participants were able to recognise stylised aspects within the synthesised locomotions generated from the deep learning framework. An investigation into the relationship between the amount of style and content on the style transfer constraint is presented. Further, the effect of varying gaits on the style transfer process between actors running and walking in a normal and zombie style are investigated. One should view style and content as an inter-related concept. This research provides an interval of acceptable amounts of style. After a point, the more style placed on the style transfer constraint does not yield major changes on the newly synthesised locomotion. The zombie stylise motion manifold took longer to transfer when the actor was running. This research demonstrates that style transfer for human locomotion is capable.

Declaration

I, Bryce Jiran Ramgovind (Student number: 728735) am a student registered for Master of Science.

I hereby declare the following:

- I am aware that plagiarism (the use of someone else's work without their permission and/or without acknowledging the original source) is wrong.
- I confirm that the work submitted for assessment is my own unaided work except where I have explicitly indicated otherwise.
- I have followed the required conventions in referencing the thoughts and ideas of others.
- I understand that the University of the Witwatersrand, Johannesburg may take disciplinary action against me if there is a belief that this is not my own unaided work or that I have failed to acknowledge the source of the ideas or words in my writing.



16/02/2021

Signature

Date

List of Figures

List of Tables

Contents

Chapter 1

Introduction

"Dance first. Think later. It's the natural order."

- Samuel Beckett

John Lasseter, director of the beloved classic *Toy Story* (1995), said in an interview for the behind the-scenes making of the film, "One of the things that makes Toy Story so unique is the collaboration between traditionally trained artists and the animators, and these amazing computer geniuses". Toy Story (1995) was the first fully computer-generated animation film and a crucial success for the animated film industry, since it spurred the release of a plethora of financially successful films [Zorthian 2015]. Toy Story (1995) marked a pivotal moment in animation and was a major advance on Steamboat Willie (1928). Lasseter looked towards the field of computer science to solve animation's limitations, thus breaking the traditional mould of animation. This monumental step towards a technology-driven approach was the forward thinking that lead to the financial and creative success that today is Disney Pixar[1].

From the ground-breaking animation film, *Toy Story* (1995), to the more recent critically-acclaimed *Coco* (2017), the field of animation has been subservient to the laborious task of character rigging. This cumbersome task, in the pipeline of producing an animated film, is an essential part of any computer generated animation as it defines the movements of a character through a three-dimensional (3D) space. Animators are described as puppeteers during the character rigging phase [Vardanega 2013].

A character in a 3D animation is known as a 3D model or wireframe. A 3D model is a mathematical representation of any 3D object and typically describes the shape of the object [Slick 2017]. The objects are typically the characters, props or environment. In Figure 1.1, a wireframe of the character Buzz Lightyear from Pixar's animated film, *Toy Story* (1995), can be seen.

[1] https://www.pixar.com/

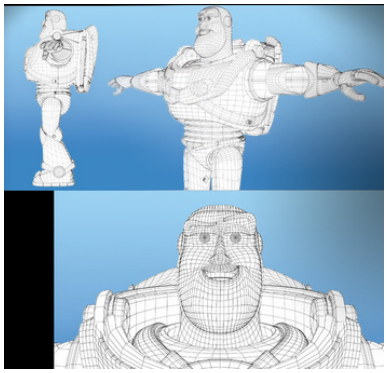


Figure 1.1: A 3D model of Buzz Lightyear from Pixar's animated film, *Toy Story* [Vardanega 2013]

Two techniques are widely used in 3D modelling. The artist can sculpt the object by hand using a clay-like substance and then the object is scanned into a computer, thus mapping the 3D object into a mathematical representation [Vardanega 2013]. This technique was employed on the *Toy Story* (1995) movie to generate the 3D model for organic structures like Buzz Lightyear's face. Eben Ostby, associate technical director of *Toy Story* (1995), described his process of creating Buzz's face by capturing points on the surface of a clay model. Once there were a sufficient number of points sampled, a mathematical surface was fitted through the points to create a 3D model , this is shown in Figure 1.2[Pixar 2016].

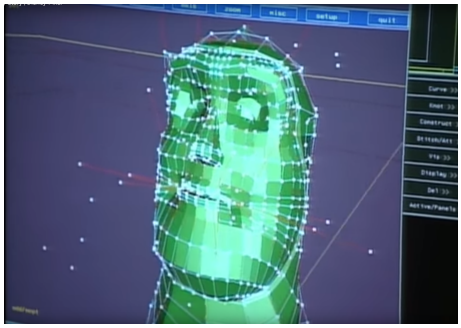


Figure 1.2: 3D modelling of Buzz's face during the behind-the-scenes making of *Toy Story*

The alternate 3D modelling process encompasses the animator directly modelling the 3D object in a 3D space within animation software. This means that the object can be rotated and viewed from a variety of angles.

The end of the modelling phase produces a static character. The subsequent phase is known as rigging. This phase gives motion to a character in a 3D space. In nature, the skeleton determines the locomotion or gait of any animal. This is particularly true when looking at bipedal versus quadrupedal locomotion in humans and our distant ancestors. A digital skeleton, character rig, is assigned to the 3D model. This rig allows technical directors to pose the model.

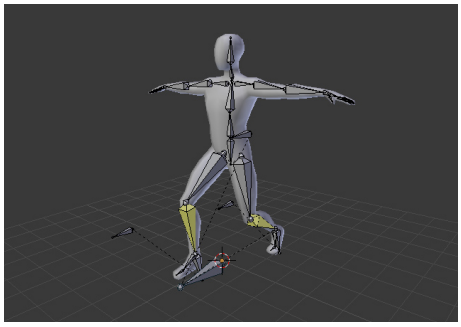


Figure 1.3: An example of a typical character rig [Shah 2012].

A character rig is similar to a real skeleton in the sense that it comprises of bones and joints. A sequence of joint movements will allow for fluid locomotion of the model in a 3D space. Figure 1.3 illustrates a basic character rig developed in Blender[2].

An approach for determining fluid locomotion is by basing it on human actors. This is known as motion capture; the simple process of recording motion data based on a human actor and mapping it to a 3D model [Pluralsight 2014]. Andy Serkis has become a household name in motion capture animation by playing characters such as Gollum in Peter Jackson's *Lord of the Rings* (2001), the titular character in *King Kong* (2005) and Caesar in the block-buster *Rise of Planet of the Apes* (2011) to name a few.
Motion capture data represents a time-series structure since each frame corresponds to some pose of a character [Holden *et al.* 2015].

A machine learning driven approach, using motion capture data to determine fluid locomotion, motion synthesis, allows animators to create realistic character movement. The key to effective character motion synthesis is the ability to learn, and then subsequently predict, the motion for the character's joint in a 3D space.

In a recent study, Satchell *et al.* [2016] investigated the relationship between a person's biological motion and their aggressive behaviour. The study analyses the thorax and pelvic movements of 29 participants by using their motion capture data. Satchell *et al.* [2016] found that locomotion is related to an individual's personality. The study indicates that a walking style of a person can be unique and there can be subtleties in the locomotion. Actors use locomotion style as a mechanism to portray different emotions within a scene. Holden *et al.* [2016] created a deep learning framework for editing motion and transforming the style of motion given an example motion clip. Their work is based on the seminal research done by Gatys *et al.* [2015] which specifically looks at transferring style of artists from a reference image to another image while maintaining its content. Holden *et al.* [2016] are able to produce a motion that maintains the timing and content of the input but has the style of another. Four locomotion styles

[2] A 3D open-source animation software https://www.blender.org/

were investigated namely, a zombie, depressed, old man and injured style. Locomotion style transfer using a deep learning framework is the core of this research. A plethora of research exists for neural style transfer for image processing however there is limited research available for style transfer for character motion synthesis [Holden *et al.* 2016].

The main objective of this research is to build and evaluate a deep learning framework for locomotion style transfer. Such an approach will allow the animators to reduce the time taken during the character rigging phase and focus on other areas of the animation process. The focus of study is to take the above factors into consideration and create realistic motion sequences by transferring locomotion style from pre-existing motions.

The rest of this dissertation is structured as follows. Chapter 2 reviews the fundamental techniques of deep learning. Chapter 3 explores the pioneering research of Gatys *et al.* [2015]. This pivotal research creates a new field of study known as neural style transfer. The rest of Chapter 3 delves into its further developments. Branches of the style transfer algorithm and its applications are presented.

Chapter 4 outlines the research methodology. The research objectives and every aspect of the problem are defined. The rest of the chapter outlines the key phases of development that were employed on this project. These development phases are data collection and preparation, locomotion style transfer and testing and evaluation. Section 4.4.1 explores the design and implementation of the deep learning framework.

The experiments conducted on this research are presented in Chapter 5. The results and analysis of a qualitative study is presented in Section 5.1. This study will provide an understanding from an audience's perspective of the success of the deep learning framework's ability to transfer style. Section 5.2 investigates the effect of the relationship between the amount of content and style needed when transferring style. The generated results and analysis are also presented in this section. An evaluation across all the metrics for each synthesised motion sequences is provided. Section 5.2.2 provides a visual analysis of the synthesised motion sequences. The performance metrics of the training process rounds off this section. Running and walking are completely different motions in the gait spectrum. Thus, Section 5.3 investigates the effect of varying the gait. The performance metrics of the training process concludes this section. These performance metrics provide insights into the computational requirements.

Chapter 6 provides a conclusion to the dissertation. Recommendations and future areas of research are given in Section 6.1 and the final remarks are shown in Section 6.2.

Chapter 2

Relevant Work

"Lesser artists borrow; great artists steal."

- Igor Stravinsky

The year is 300 BC. The sun peeks its golden head above the horizon petrified to shed some light. The water lapses. Seafoam dances with folly on the shore before retreating back into the endless ocean. The fishermen of Crete start their day early to battle the beast, that is the sea. A loud, thunderous stomping sound booms in the distance. The colours of twilight seem to have disappeared. A deep shimmering blue sprawls across the sky. The stomping starts to grow louder. A human-like figure is spotted in the distance, yet this man appears to have a warm, caramel skin colour with a hard metallic finish. The man grows taller and taller before coming into focus. The fishermen were not alarmed nor phased by this. One man waved at this giant, centurion, bronze man but the mechanical creature continued his daily patrol. Protecting the people of his island.

Talos was the guardian of the island of Crete. Created by Hephaestus at Zeus' command, he is the first *robot* in ancient writing [Mayor 2018]. Humanity has always enjoyed the idea of creating intelligence. Technology has since grown to a point where self-aware robots are a reality. The artificial neural network has propelled the field of artificial intelligence, branching from fields of statistics, mathematics and computer science.

In 1958, Frank F. Rosenblatt unveils a novel way of modelling a hypothetical nervous system or perceptron [Rosenblatt 1958]. Rosenblatt creates an analogy between the perceptron and biological systems within the human body. The perceptron is the fundamental processing unit of artificial neural networks. The human brain consists of many cells (neurons) each connected by synapses [Cohen 2013]. The neuron relies on information carried along synapses as input [Cohen 2013].

2.1 The Neuron

The neuron is the most basic component of the neural network. It is the mathematical imitation of neuron observed within the human brain. Figure 2.1 provides an illustration of this neuron. For given input signal x_j, a connecting link j is established within the neuron and by continuing the analogy, this would be known as a *synapse* [Haykin 2009]. Therefore, an input signal x_j along synapse j would be multiplied by a *synaptic weight* θ_j. A summation of the all the synapses are calculated producing a vector v which will be fed into an activation function, ϕ. It is important to apply this activation function as it limits the amplitude of the output of the neuron. This function can also be referred as a *squashing* function since it limits the amplitude range that is permissible in the output signal [Haykin 2009]. The bias θ_0 is included in the neuron as it has an effect on the net input of the activation by either increasing or decreasing depending on whether it positive or negative, respectively [Haykin 2009].

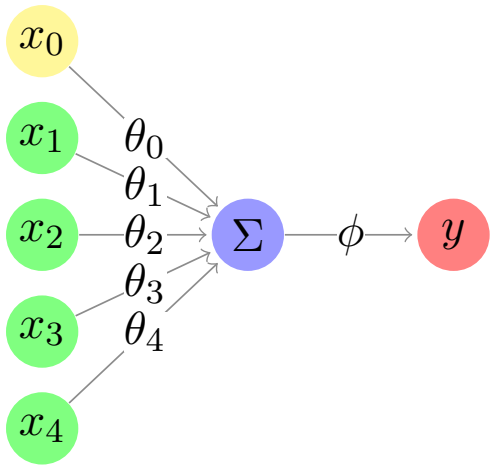


Figure 2.1: Visual representation of the non-linear computing neuron. Adapted from [Haykin 2009].

The neuron can be described in mathematical terms. The summation of the synapses including a bias term can be expressed in the form of

$$u = \sum_{j=1}^{m} \theta_j x_j \tag{2.1}$$

$$v = u + b \tag{2.2}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_m]$ is a set of input signals, $\mathbf{\Theta} = [\theta_1, \theta_2, \ldots, \theta_m]$ is a set of synaptic weights, b is the bias term and m corresponds to the number of input signals.

Therefore, output of the neuron can be expressed as follows

$$y = \phi(v) \tag{2.3}$$

where y is the output signal of the neuron and ϕ is the activation function. The entire neuron can be expressed in the following way

$$y = h(\mathbf{x}; \Theta) = \phi(\sum_{j=1}^{m} \theta_j x_j) \tag{2.4}$$

where $\mathbf{x} = [x_1, x_2, \ldots, x_m]$ and $\mathbf{\Theta} = [\theta_1, \theta_2, \ldots, \theta_m]$.

2.2 Feed-forward Networks

By connecting multiple neurons together, the true core of this new computing paradigm comes alive. Neurons are organised in layers, thus creating layered neural networks. The simplest form of a neural network is pictorially represented in Figure 2.2. The layer of neurons transform the input layer, $\mathbf{x} = [x_1, x_2, \ldots, x_m]$, into an output layer, $\mathbf{y} = [y_1, y_2, y_3, \ldots, y_n]$.

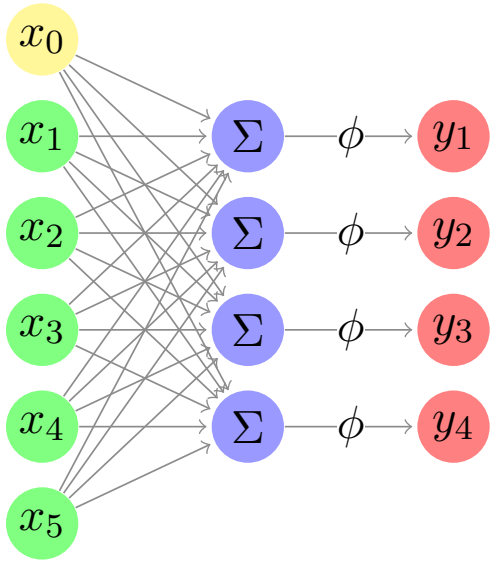


Figure 2.2: Single-layer Feed-forward neural network. For simplicity, the multiplication of the synaptic weights and input signals are indicated by the arrows into a neuron. The neuron consists of the summation of this multiplication and then the activation function being applied onto the result. Adapted from [Haykin 2009].

The single-layer feed-forward neural network directly maps the number of neurons to the number of outputs produced by the network. For this reason, the concept of hidden layers is introduced. By stacking an additional layer of neurons after the initial neuron layer, the number of outputs can be limited. This is

due to the fact, the output of the first neuron layer will now serve as input into the second neuron layer thus by controlling the number of neurons in the second layer will limit the number of outputs of the network.

Churchland and Sejnowski [2016] state that by adding a hidden layer, the network is able to gain a global perspective because of these extra synaptic connections. The network is able to receive and extract higher-order statistics from its input signal [Haykin 2009].

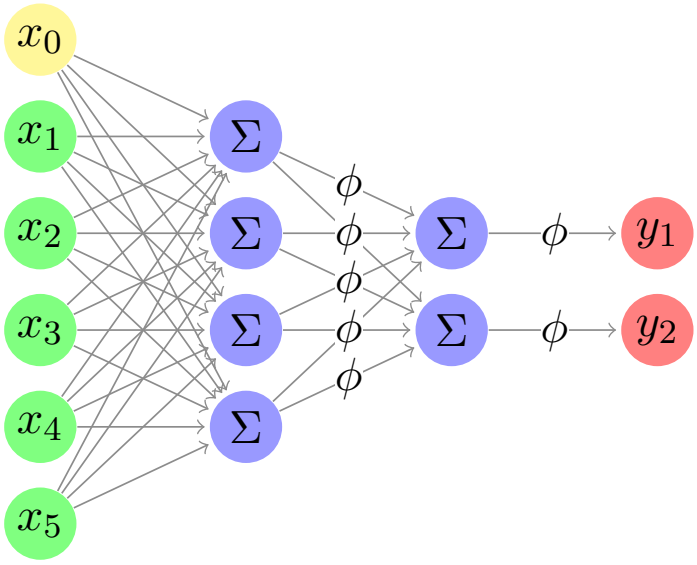


Figure 2.3: Multi-layer Feed-forward neural network. The neuron's in each layer will have their input signals as the output of the preceding layer. Only 2 neurons were included in the second layer to illustrate that the number of outputs can vary. Adapted from [Haykin 2009].

A *fully connected* neural network is depicted in Figure 2.3. That is, every node in each layer is connected to every node in the succeeding forward layer.

2.2.1 Activation Functions

Activation functions play a fundamental role to the success of a trained neural network. A typical example of an activation function can be seen in Equation 2.3. The use of different mathematical functions could result in greater accuracy. Pedamonti [2018] explore the use of different functions for classifying hand-writing using the *MNIST* dataset which was created and curated by LeCun and Cortes [2010]. The activation function is present in every neuron in the neural network.

Sigmoid Function

Considered the most common form of activation functions, the sigmoid function is an increasing function which allows for a balance between linear and nonlinear behaviour [Haykin 2009]. The sigmoid function is defined by the following

$$h_\Theta = \frac{1}{1 + e^{-\alpha(\Theta^T \mathbf{x})}} \tag{2.5}$$

where α is the slope parameter of the sigmoid function. The limits of the natural exponential function is 0 and infinity when $\alpha(\Theta^T \mathbf{x})$ tends toward infinity and negative infinity, respectively. Thus, the bounds of the sigmoid function defined in Equation 2.5 are 1 and 0 when $\alpha(\Theta^T \mathbf{x})$ tends toward infinity and negative infinity, respectively.

The parameters of a neural network are learned during the training process due to back-propagation [E. Rumelhart *et al.* 1986]. While the sigmoid function has shown success during the training process on shallow neural network architectures, networks with more layers propagate a diminishing gradient. This is because the derivatives of the sigmoid functions which are less then 1 will be multiplied many times thus tending to zero [Pedamonti 2018]. This is called the vanishing or exploding gradient problem. The Rectifier Linear Unit (ReLU) function has an ability to overcome this problem.

Rectifier Linear Unit Function

Introduced by Nair and Hinton [2010], the Rectifier Linear Unit (ReLU) has risen to prominence in recent years due to its success within deep learning neural networks [Goodfellow *et al.* 2016].

The ReLU activation function takes the form of the following equation:

$$ReLU(v) = max(0, v) \tag{2.6}$$

$$\frac{d}{dv} relu(v) = \begin{cases} 0 & \text{if } v \leq 0 \\ 1 & \text{if } v > 0 \end{cases} \tag{2.7}$$

2.2.2 Creating Learning Algorithms

The introduction of the back-propagation algorithm allows for the learning of the parameters for the best function approximation [E. Rumelhart *et al.* 1986].

Given input **x**, a feed-forward network will produce an output **y**. The information will flow in a forward direction through the network while propagating up to each layer's hidden neuron [Goodfellow *et al.* 2016]. This process is known as *forward-propagation*. A loss function, $\mathcal{L}$, is calculated during training as forward propagation will create an output of the network $\hat{y} = h_\Theta(\mathbf{x})$,

$$\mathcal{L}(\hat{y}, y) = -[(1-y)\log(1-\hat{y}) + y\log(\hat{y})] \tag{2.8}$$

The *learning* algorithm is created by updating all the weights, Θ, of the network in each layer [E. Rumelhart *et al.* 1986]. For any given layer, l, the weights can be updated using the gradient descent algorithm

$$\theta_{j,i}^{(l)} := \theta_{j,i}^{(l)} - \alpha \frac{\partial \mathcal{L}}{\partial \theta_{j,i}^{l}} \tag{2.9}$$

where α is the learning rate. So in order to calculate the new weights, the term, $\frac{\partial \mathcal{L}}{\partial \theta_{j,i}^{l}}$, must be calculated.

By employing an optimisation algorithm, the network's weights are able to iteratively update and improve the fit of the function. This step gives the learning power to machine learning algorithms. In their paper entitled "Adam: A Method for Stochastic Optimization" by Kingma and Ba [2014], a novel optimisation algorithm known as ADAM is proposed and has grown to massive popularity. It is touted as one of the best optimisation algorithm for adaptive gradient descent. ADAM boasts its computationally efficiency with very little requirement for memory. ADAM is able to store an average of exponentially decaying past squared gradients. Holden *et al.* [2016] recommend using adaptive gradient descent for locomotion style transfer.

2.3 Convolutional Neural Network

Image and video recognition has received new-found fame in recent years due to its application within industry. This perplexing task requires a new paradigm of computing often breeding new additions to the artificial neural network (ANN) library. Lecun *et al.* [1998] proposes a new addition by re-inventing the architecture to combine the convolution and max pooling functions thus creating the convolutional neural network.

The convolutional neural network (ConvNet) is a variant of the neural network which aims to mimic

the behaviours of the visual cortex within the human eye [Fukushima 1980]. The ConvNet has set the benchmark and has been praised for its use within the field of computer vision since it allows for scaling within three dimensions.

The pixel intensity of an image live within a two dimensional pixel grid. However, since images are composed of 3 distinct channels namely red, green and blue (RGB) and the pixel intensity differs in each dimension, a third dimension must be considered. An image of size 128x128x3, as an example, is 128 pixels wide, 128 pixels high and have 3 channels (RGB). Given a sample image, the corresponding red, green and blue pixel intensities can be extracted and expressed along three dimensions.If the feed-forward neural network was used, a single neuron from the first hidden layer of the network would have to have 3072 weights since $128 \times 128 \times 3 = 49152$. This becomes computationally impossible for larger images.

The ConvNet changes its architecture to consider inputs that have three dimensions.

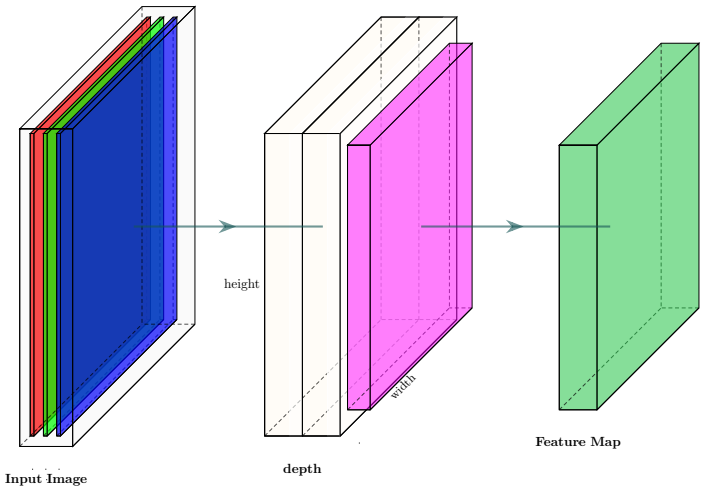


Figure 2.4: Architectural diagram of the Convolutional Neural Network with a single convolutional and max pooling layer. Adapted from [Krizhevsky *et al.* 2017].

In Figure 2.4, the neurons are arranged in a three dimensional format. The ConvNet hierarchically processes visual information through feed-forward propagation. Each layer of the network can be considered a collection of image filters and then the subsequent output of each layer is known as a feature map. In other words, the feature maps are different versions of the input image.

2.3.1 Convolution

Given a small pre-defined matrix of numbers and an image, a transformation can be created by determining the dot product for regions within the original image. This is the loose definition of the

convolution function. More formally, given an image **P**, which is a two-dimensional array of pixel data along a single RGB channel, and a filter kernel **F**, where $f_{m,n}$ is any real value for $m \in \{-M, ..., M\}$ and $n \in \{-N, ..., N\}$, the convolution of **P** with filter **F** is defined by

$$x_{i,j} = \sum_{m=-M}^{M} \sum_{n=-N}^{N} p_{i-m,j-n} f_{m,n} \tag{2.10}$$

where $x_{i,j}$ is the resulting transformed image.

The above equation merely describes the convolution function in terms of a single channel or gray-scale images, however the world lives in colour. Images have three dimensions, namely red, green and blue. In order to render a single image, these three channels have to be combined. Using this and the principles above, each channel can be convolved separately and then combined to generate the transformed image [Krizhevsky *et al.* 2017].

2.3.2 The Convolutional Layer

The convolutional layer consists of applying the convolution function on a collection of filter maps. For a given image **P**, its dimensionality is (w, h, d) where w is the width or number of columns, h is the height or number of rows and d is the depth or number of channels. In this case there are only 3 channels, namely red, green and blue.

The filter map's dimensions can then be described as (w_f, h_f, d). Let $w_f < w$ so that the filter will operate on small region of the input image. After performing the convolution function above, the output dimension would be $(w + w_f - 1, h + h_f - 1)$. This is evident when the depths of the input image and the filter map are the same.

A convolutional layer will consist of multiple filter maps which will create an output 2D matrix, known as a slice. The stacking slices into a matrix, the dimensionality of the output is $(w + w_f - 1, h + h_f - 1, n_f)$ where n_f is the number of filters applied.

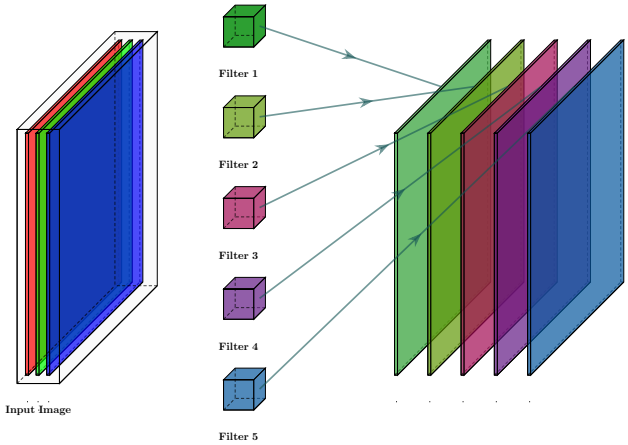


Figure 2.5: Visualisation of the convolution of multiple filter maps and its combinations in the output. Adapted from [Goodfellow *et al.* 2016].

The final step of the convolution layer will apply an activation function seen in Subsection 2.2.1.

2.3.3 The Pooling Layer

The pooling layer is used to reduce the input at each layer thus increasing speed of computation [Goodfellow *et al.* 2016]. The premise is simple, divide the image into smaller sections and then find an aggregation within each sections. The aggregate values are assembled in a grid to represent the original input. For multi-channel images, the pooling function is applied on each channel and then stacked together.

2.3.4 Convolutional Neural Network Architectures

Since its inception, the convolutional neural network has garnered many variation in its architectural design. Although put simply, the architectural configuration of the convolutional and pooling layers have near infinite arrangements which aim to solve particular problems within computer vision.

LeNet-5

The convolutional neural network was widely popularised by Lecun *et al.* [1998] in their article entitled “Gradient-Based Learning Applied to Document Recognition”. Lecun *et al.* [1998] aims to classify handwritten characters and achieves a 99.2% classification accuracy on the MNIST dataset.

The article proposes the use of the convolutional layer then a pooling layer using an average aggregation. This architecture is repeated two and half times before being passed into fully connected layers

to make the final prediction. A visual representation of the network from the paper can be seen in Figure 2.6.

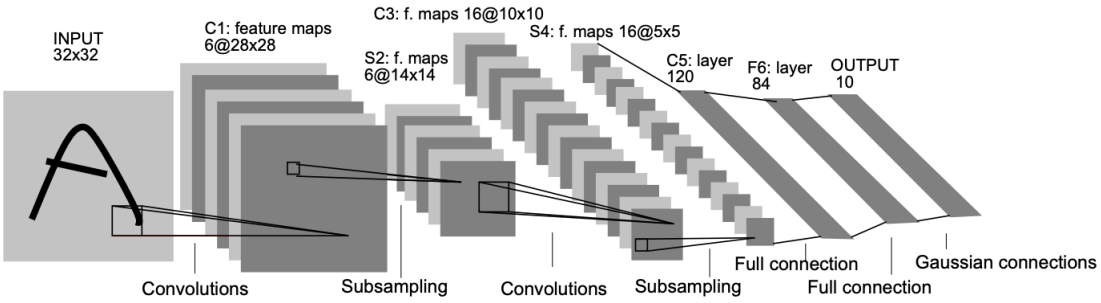


Figure 2.6: Architectural diagram of LeNet-5, a convolutional neural network, used to recognise handwritten characters. Extracted from [Lecun *et al.* 1998].

The design and architecture of the repeating convolutional layer followed by the pooling layer laid the stage for many variants of the convolutional neural network which are still used today.

AlexNet

Possibly the most pivotal contribution to deep learning, Krizhevsky *et al.* [2012] introduces the AlexNet which is considered to have sparked the Renaissance of neural networks as well as marking the beginning of deep learning applied to computer vision. AlexNet premiered at ImageNet Large Scale Visual Recognition Challenge (ILSVRC) which is considered the Olympics of object detection and image classification. All the best researchers within the field of computer vision pit their algorithms against each other in this detection battle.

Krizhevsky *et al.* [2012] built upon LeNet-5 but opted to switch out the average pooling layers with max pooling layers. AlexNet made a stride by using the rectified linear activation function (ReLU) instead of the standard convention of the logistic or tanh functions. The final layer of the network uses a softmax activation function which is now a standard for multi-class classification in any neural networks.

Five convolutional layers are present within AlexNet with three fully connected layer for objection detection.

VGG

Aptly named after their department within the University of Oxford, Visual Geometry Group (VGG), Simonyan and Zisserman [2014] presents a new variant of the ConvNets. VGG was the first runner up in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014.

Simonyan and Zisserman [2014] introduces the idea of using small filters, specifically filters with 3X3 and 1X1 with the stride of one, unlike its predecessors which opted for much larger sized filters. A number of similar variants were introduced in later years and were named after their number of layers, VGG-16 and the VGG-19. The major difference between these two architectures are attributed to the number of convolutional layers within the last 3 convolutional segments. Here, a single convolutional layer is added to each segment while still maintaining the same filter sizes of its predecessor. Figure 2.7 provides a visual representation of the network by indicating the three dimensions present in image computation.

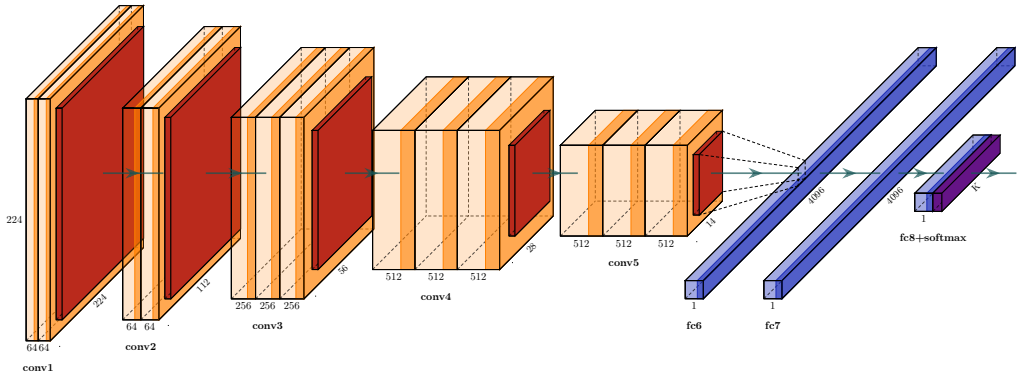


Figure 2.7: Visualisation of the ConvNet's configurations presented in [Simonyan and Zisserman 2014].

Simonyan and Zisserman [2014] compared their new-found architecture against pre-existing ConvNet architectures by entering it in the ILSVRC-2012 and ILSVRC-2013 competitions. VGG performed significantly better by decreasing the error rate for detection [Simonyan and Zisserman 2014]. VGG led the charge in releasing their model under a permissive license. This gave the rise to transfer learning within the computer vision community. Pre-trained models are used as a base and trained on new datasets to solve new computer vision problems.

GoogLeNet

Winner of the ILSVRC-2014 challenge, Szegedy *et al.* [2015] present their new convolutional architecture in their model called GoogLeNet. The inception module is the heart of this paper. Szegedy *et al.* [2015] proposes convolutional layers with differing sized filters are placed in parallel with a max pooling layer. This can be seen in Figure 2.8.

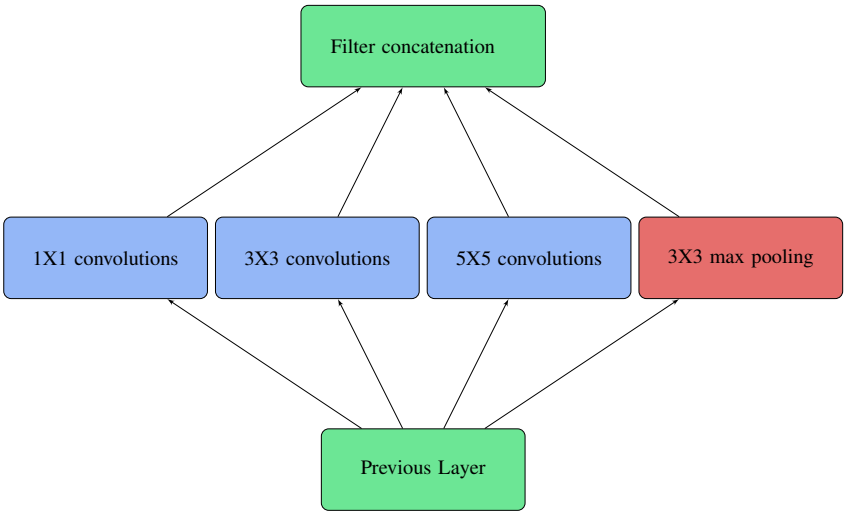


Figure 2.8: The inception module presented by Szegedy *et al.* [2015]. There are three convolutional layers with filter sizes 1x1, 3x3 and 5x5. The max pooling layer has a filter size of 3x3.

From the humble neuron to the complexities of the convolutional autoencoder, the field of artificial neural network is booming with applications within various fields. One such field is character motion synthesis.

2.4 Character Motion Synthesis

The main aim of any animator working on an animation movie is believability. Whether the story be about a world of monsters who work at a factory that harnesses the energy of a human child's scream, *Monsters, Inc.* (2001), or a family of superheroes that have to save the world, *The Incredibles*, (2004), the animators job is to create these fantasy world within a believable construct that is, elements of the film should be rooted in reality. Animators use fluid character rigging as a device to sell believability to the audience.

Holden *et al.* [2016] present a deep learning framework based on existing deep learning techniques for the purpose of creating character motion synthesis. Their data-driven approach for motion synthesis produces realistic character movement so that the animators can only provide high level parameters such as trajectory. Holden *et al.* [2016] develop a convolutional autoencoder to learn the manifold of human motion so that it can reproduce the motion data as well as produce novel motions through interpolation. Their framework does not require any manual preprocessing or labeling thus transforming it to an unsupervised algorithm.

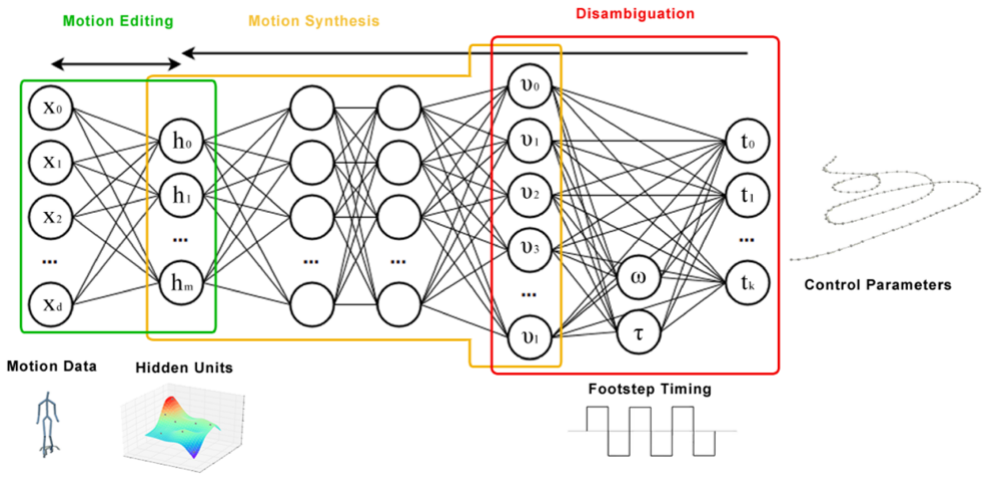


Figure 2.9: Structural diagram of the convolutional autoencoder presented by Holden *et al.* [2016].

The architectural design of the deep learning framework developed by Holden *et al.* [2016] is presented in Figure 2.9. The general motion manifold is learned in the green box in Figure 2.9. A feed-forward neural network, orange box in Figure 2.9, is the middle step in the framework which creates a mapping between the high level parameters, trajectory of the character on a terrain and the hidden units. Finally, the red box in Figure 2.9 is used to disambiguate the input by computing control parameters. By using the framework presented by Holden *et al.* [2016], the animator can simply draw a trajectory on the terrain and a motion of the character walking or running will be produced.

2.5 Conclusion

Humankind's ability to create has propelled our ingenuity to a new level with the invention of artificial intelligence. Great strides have been made within the advancement of artificial neural networks. The neuron provides a humble basis for more complicated architecture that can be built upon.

The feed-forward neural network is built by stacking multiple neurons in an organised layer. These layers feed into each other before ultimately producing the output. The hidden layers in neural networks provide a global perspective [Churchland and Sejnowski 2016]. Areas of research have been dedicated to the development and evaluation of activation functions. The ReLU function by Nair and Hinton [2010] has had monumental success within deep learning architectures due to its ability to prevent the vanishing gradient descent phenomenon during backpropagation.

By combining the convolution and max pooling function, a new neural network paradigm was invented [Lecun *et al.* 1998]. AlexNet is considered to be a pivotal contribution to deep learning for its

ability to detect visual imagery within images. AlexNet uses the ReLU activation function.

Simonyan and Zisserman [2014] presents a new variant of the ConvNet which was named after their department within the University of Oxford, Visual Geometry Group (VGG). VGG was the first runner up in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014. Here, a single convolutional layer is added to each segment while still maintaining the same filter sizes of its predecessor. Simonyan and Zisserman [2014] compared their new-found architecture against pre-existing ConvNet architectures by entering it in the ILSVRC-2012 and ILSVRC-2013 competitions. Pre-trained models are used as a base and trained on new datasets to solve new computer vision problems.

Typically designed for image processing, the ConvNet is shown to be adaptable within the context of character motion synthesis. Holden *et al.* [2016] presents a deep learning framework based on existing deep learning techniques to create character motion synthesis. Their framework does not require any manual preprocessing or labeling thus transforming it to an unsupervised algorithm. Holden *et al.* [2016] can display the use of a traditional computer vision algorithm within the context of character motion synthesis using motion capture data.

Chapter 3

Artistic Style Transfer

"A true artist is not one who is inspired, but one who inspires others."

- Salvador Dali

From the world-renowned "Mona Lisa" by Leonardo da Vinci to the surrealist "The Persistence of Memory" by Salvador Dali, the great artists of human history are able to create a harmonious marriage between style and content thus giving the audience a unique visual experience. The content of a visual art piece can be viewed as the subject matter. The content usually describes the artwork in a literal sense, for example, a painting of a woman or a painting of a landscape. The artist then uses style as a vehicle to convey an emotion or mood to the audience.

(a) Picasso, Pablo. *Head of a Horse, Sketch for Guernica (Tête de cheval, étude pour Guernica)*. May 2, 1937. Grands-Augustins. Paris.

(b) Forest, Crista. *Palomino Horse Portrait*. *fineartamerica.com*. Web. September 30th, 2015

Figure 3.1: The illustrative difference between style in artwork.

Figure 3.1a and 3.1b illustrate the difference between style and content. A horse's head is the content of both figures. In Figure 3.1a, Pablo Picasso uses his mastered style to create a high emotional mood. This is achieved by using high value contrasts as well as overly exaggerating horse features. Crista

Forest, Figure 3.1b, creates a photo-realistic depiction of a horse by using multiple layers of colours and shading. Figure 3.1b creates a harmonious rhythm which directly contrasts Figure 3.1a. Clearly, the content for both figures remain the same however the style vastly differs.

This chapter provides an extensive review of relevant literature related to the topic of artistic style transfer. Section 3.1 will examine the fundamental components of artistic style transfer. The pioneering research into photo-realistic style transfer is presented in Section 3.2. Semantic mappings are explored in Section 3.3 as a technique to preserve semantic structures within the image. Section 3.4 illustrates the ability of learning and applying multiple styles during the transferring process. The use of the neural style transfer algorithm can be applied in a variety of areas. The key concepts from image style transfer provides a foundational basis for which locomotion style transfer is possible which is presented in Section 3.5.

3.1 Neural Style Transfer

Gatys *et al.* [2015] present an algorithmic approach of composing the unique exchange between style and content in artistic creation. In their seminal work, "A Neural Algorithm of Artistic Style", Gatys *et al.* [2015] introduce a deep neural network that generates artistic images based on the style from one image and the content of another. Gatys *et al.* [2015] use a deep learning framework, convolutional neural network, to illustrate the separability of the representations of content and style. A new image is subsequently produced by manipulating both representations independently. Gatys *et al.* [2015] demonstrate this capability by generating new images based on the content representation of the "Neckarfront" in Tübingen, Germany and the style representations of well-known artists throughout history. The new images comprise of colours and local structures that were provided by the artwork while still maintaining the global scenery of "Neckarfront". Hence, this creates images in the style of the artwork.

Figure 3.2: Illustrating the artistic style transfer framework developed by Gatys *et al.* [2015]. a) original image; and (b-f) the newly generated images. The artwork that is used to generate the image is provided in the bottom left corner of each panel (b-f)

This can be seen in Figure 3.2. Gatys *et al.* [2015] use a convolutional neural network based on the VGG Network which was first introduced by Simonyan and Zisserman [2014].

3.1.1 Content Representation

A non-linear filter bank is defined at every layer within the convolutional neural network where the complexity is directly proportional to the position within the network. It can then be said that for a given image, $\bar{x}$, it will be encoded at every layer by the filter responses. Therefore, a layer has a direct correspondence between the number of distinct filter and feature maps. A layer with N_l distinct filters, as an example, will have N_l feature maps of size M_l each. M_l is the height multiplied by the width of the feature map. For a layer l, a matrix $F^l \in \mathbb{R}^{N_l \times N_l}$ can store the responses given that, for layer l, F^l_{ij} is the activation of the i^{th} filter at position j.

Thus, the following equation can be defined.

$$\mathcal{L}_{content}(\bar{x}, \bar{b}, l) = \frac{1}{2} \sum_{i,j} (F^l_{ij} - B^l_{ij})^2 \tag{3.1}$$

where $\bar{b}$ and $\bar{x}$ are the original content image and the generated image, and the subsequent feature representation in layer l are F^l_{ij} and B^l_{ij}. Equation (3.1) represents the squared-error loss function. The partial derivative of the loss function, Equation (3.1), with respect to the activation of the i^{th} filter at position j for layer l, F^l_{ij}, is defined by equation (3.2).

$$\frac{\partial \mathcal{L}_{content}}{\partial F^l_{ij}} = \begin{cases} (F^l - B^l)_{ij} & \text{if } F^l_{ij} \geq 0 \\ 0 & \text{if } F^l_{ij} < 0 \end{cases} \tag{3.2}$$

ConvNets develop a representation of an image when trained for the purpose of object recognition. Thus, the deeper layers of the network are able to capture the object information more explicitly and the actual content of the image becomes revealed by the transformation along the processing hierarchy into the representations. The content representation can be defined as the feature responses in the higher layers.

3.1.2 Style Representation

Gatys *et al.* [2015] illustrate that texture information can be captured using a designed feature space which can be built on top of the filter responses in any layer of the neural network. The gram matrix , $G^l \in \mathbb{R}^{N_l \times N_l}$, is defined as

$$G^l_{ij} = \sum_k F^l_{ik} F^l_{jk} \tag{3.3}$$

which is the inner product of feature maps i and j in layer l. Gatys *et al.* [2015] state that this is the stationary multi-scale representation of the input image. In other words, the gram matrix is the measure of the correlation of the feature maps at each layer. For the original stylised input image, $\bar{a}$, and the newly generated image, $\bar{x}$, the style representation is the gram matrix for each layer l, A^l_{ij} and G^l_{ij}, respectively. Thus, the total loss of style can be defined by

$$\mathcal{L}_{style}(\bar{x}, \bar{a}) = \sum_{l=0}^{L} w_l \left(\frac{1}{4N_l^2 M_l^2} \sum_{ij} (G^l_{ij} - A^l_{ij})^2 \right) \tag{3.4}$$

where the contribution of each layer to the total loss is defined by a weighting factor of w_l, L is a set of layers in the network.

3.1.3 Style Transfer

Once the individual style and content representations are determined, Gatys *et al.* [2015] state that for a given artwork $\bar{a}$ and a photograph $\bar{b}$, a new image $\bar{x}$ can be created which illustrates the ability to transfer the style of $\bar{a}$ while maintaining the content of $\bar{b}$ by minimizing the loss function

$$\mathcal{L}_{total}(\bar{a}, \bar{b}, \bar{x}) = \alpha \mathcal{L}_{content}(\bar{b}, \bar{x}) + \beta \mathcal{L}_{style}(\bar{a}, \bar{x}) \tag{3.5}$$

where α and β are the weighting parameters for content and style, respectively. Gatys *et al.* [2016] found that style and content within images are completely separable and can be manipulated independently. However, style and content should not be viewed separately for neural style transfer. Equation 3.5 presents a linear combination of the loss function for content and style. By regulating the weighting parameters, the amount of style and content can be controlled during the reconstruction of the image.

Gatys *et al.* [2016] show that a strong emphasis on style nearly eliminates the content from the content images and almost perfectly matches the style images. The opposite is true when a strong emphasis is placed on the content, as the style from the style image almost disappears however the content is matched perfectly. The authors opt for the expression of the weighting parameters in Equation 3.5 as ratio of the style and content. A visualisation of this can be seen in Figure 3.3.

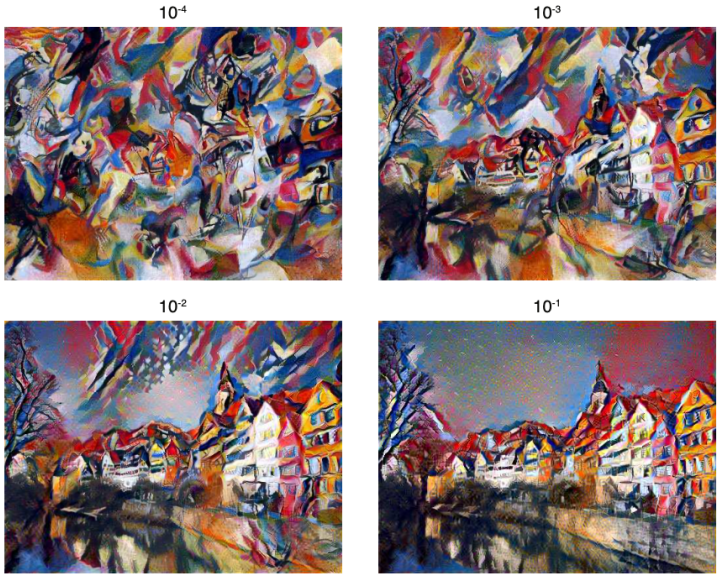


Figure 3.3: The reconstruction of an image with the style from *Composition VII* by Wassily Kandinsky, 1913 and a photograph depicting the Neckarfront, Germany. The relative weighting of the content to style ratio $\frac{\alpha}{\beta}$ was varied and indicated above each image. A high emphasis on style is placed on the images in the top left corner and a low emphasis on style is placed on the bottom right corner. Extracted from [Gatys *et al.* 2016].

The author decided to place a high emphasis on the style weighting parameter (β). Effectively lowering the ratio of weightings to 10^{-4}. This image is highly stylised and almost completely resembles the original artwork. It can be seen that little content is present. On the other end of the spectrum, the author looked at lowering the style parameters and hence a larger ratio of 10^{-1} is present. This decreased the amount style within the reconstruction and resembles the original photograph with varied colours. Thus, Gatys *et al.* [2016] adopted a middle ground weighting of 10^{-3}.

Li *et al.* [2017a] build on the neural style transfer algorithm but show that the optimisation of Equation 3.5 will result in the loss of many structures within reconstruction. This is due to the domination of the low-level feature of ConvNets from the stylised images within the reconstructed image. Low-level feature are seen at the local details such as colours, textures or shapes. High-level features are the semantic details such as the subject matters or global structures. By optimising Equation 3.5, the style reconstruction becomes inappropriate when the stylised images are not explicitly artwork. Artifacts are re-created in the newly reconstructed images as well as distortion of structures which are found within the original content image [Li *et al.* 2017a].

The introduction of a Laplacian loss term by Li *et al.* [2017a] within the total loss function illustrates that edges and contours, among other details in structures, from the original content image can be maintained. The Laplacian loss term is defined by

$$\mathcal{L}_{lap}(\bar{b}, \bar{x}) = \sum_{i,j} (D \circledast \bar{b} - D \circledast \bar{x})^2_{ij} \tag{3.6}$$

where $\bar{b}$ is the original stylised image, $\bar{x}$ is the newly created image and $\circledast$ denotes the convolution between an image and the Laplacian matrix D. Equation 3.5 can be modified accordingly

$$\mathcal{L}_{total}(\bar{a}, \bar{b}, \bar{x}) = \alpha \mathcal{L}_{content}(\bar{b}, \bar{x}) + \beta \mathcal{L}_{style}(\bar{a}, \bar{x}) + \gamma \mathcal{L}_{lap} \tag{3.7}$$

Li *et al.* [2017a] note that the stylisation process was not disturbed by introducing the Laplacian term as the individual content and style losses increases slightly.

3.2 Photo-realistic style transfer

The modern age has ushered in a new artist. This artist is known as the photographer, their 'paintbrush' is the digital camera and the entire world is their canvas. Acclaimed photographer and journalist, Steve McCurry captured one of the most recognisable photographs in the twentieth century.

Figure 3.4: McCurry, Steve. *Afgan Girl*. June, 1985. Nasir Bagh refugee camp, Pakistan.

Her hauntingly, beautiful stare graced the cover of National Geographic's June 1984 magazine issue. This photograph became the symbol of the refugee crisis within Pakistan due to the Soviet occupation of Afghanistan and garnered international attentions for female rights. The high contrast between the green background and the torn maroon scarf worn by the girl blends perfectly to evoke a vivid response from the viewer. This composition of colour creates a unique style. The most striking feature of the photograph is her green eyes which is not a typical feature of people of colour. McCurry is able to capture the power of his medium, photography.

Artists create with their particularly honed style, and Gatys *et al.* [2015] shows that this style is transferable however, their resulting images appears less realistic. The straight lines of buildings appears wavy or textured similar to the original artwork. Luan *et al.* [2017] present a deep learning framework for photographic style transfer where the reconstructed image should appear as photo-realistic as possible. Luan *et al.* [2017] build of the work of Gatys *et al.* [2015] by constructing a new term to Equation 3.5. This term should ensure that the photorealistic property is not lost during the transfer process by penalising image distortions. Luan *et al.* [2017] create a function that will ensure that the structure of the input photograph is preserved by using an affine function to map the input RBG values to the generated image's equivalent RBG value. This ensures that the transformation is locally affine in the colour space. By building on the research of the Matting Laplacian from Levin *et al.* [2006], a grayscale matte can be expressed as a local affine combination of the input channels (RGB).

Given the original input image $\bar{a}$, the newly generated image $\bar{x}$ and RGB channel c, a photorealism

regularization function can be defined.

$$\mathcal{L}_m = \sum_{c=1}^{3} V_c[\bar{x}]^{\mathbf{T}} \mathcal{M} V_c[\bar{x}] \tag{3.8}$$

where $V_c[\bar{x}] \in \mathbb{R}^{N \times 1}$ represents a vectorised version of the output image $\bar{x}$ in channel c and $\mathcal{M} \in \mathbb{R}^{N \times N}$ represents least-square penalty function that can be minimised with a standard linear system [Levin *et al.* 2006]. Luan *et al.* [2017] note that there is a limitation on Equation 3.4 and augments it to include a semantic segmentation.

$$\mathcal{L}_{style+}(\bar{x}, \bar{a}) = \sum_{l=0}^{L} \sum_{c=0}^{C} w_l \left(\frac{1}{4N_{l,c}^2 M_{l,c}^2} \sum_{ij} (G_{ij}^{l,c} - A_{ij}^{l,c})^2 \right) \tag{3.9}$$

where C denotes the number of channels, $\bar{a}$ is the original stylised input image, the newly generated image is $\bar{x}$, A_{ij}^l and G_{ij}^l represents the style representation and gram matrix along each layer l, respectively. A weighting factor is included in the form of w_l and L is a set of layers in the network.

The following can then be re-defined for the style and content images separately:

$$F^{l,c} = F^l M^{l,c} \tag{3.10}$$

where $M^{l,c}$ is the segmentation mask for channel c in layer l. Luan *et al.* [2017] define a new loss function for a given stylised photograph $\bar{a}$ and a content photograph $\bar{b}$. The new image $\bar{x}$ which contains the style of $\bar{a}$ while maintaining the content of $\bar{b}$ is generated by minimizing the loss function

$$\mathcal{L}_{total}(\bar{a}, \bar{b}, \bar{x}) = \alpha \mathcal{L}_{content}(\bar{b}, \bar{x}) + \beta \mathcal{L}_{style+}(\bar{a}, \bar{x}) + \lambda \mathcal{L}_m \tag{3.11}$$

Luan *et al.* [2017] show their approach is able to reconstruct the image with added style and has no distortions while maintaining a similar texture compared to the reference images. However, in some uniform regions of images an over stylisation was applied creating artifacts which should not ordinarily be present. This is due to the fact that a semantic segmentation is applied on the style constraint.

Inspired by Luan *et al.* [2017], the authors of an article entitled, "Photorealistic Style Transfer with Screened Poisson Equation" by Mechrez *et al.* [2017] strive to improve this photorealism transfer technique. They adopt a similar two-stage approach to Luan *et al.* [2017] however, they use different post-processing technique. Luan *et al.* [2017] show that using the Screen Poisson Equation (SPE), first introduced by Bhat *et al.* [2008], will create more photo-realistic images. The use of SPE has demonstrated

a decrease in unnatural textures within uniform regions, remove distorted structures such as wavy line which should be straight and preserve fine details which are present within the original content image. However, 'halo' effects are created around strong edges. The SPE post-process takes 1.7 seconds for 640×400 images and performs significantly better than existing post-processing techniques [Mechrez *et al.* 2017].

Li *et al.* [2018] propose an novel algorithm which aims to improve efficiently and effectiveness of photo-realistic style transfer. They design a two stage approach where the first steps looks at performing the style transfer and the second step is a smoothing function which removes any inconsistent artifacts created in uniform semantic regions. Their novel architecture can be seen in Figure 3.5.

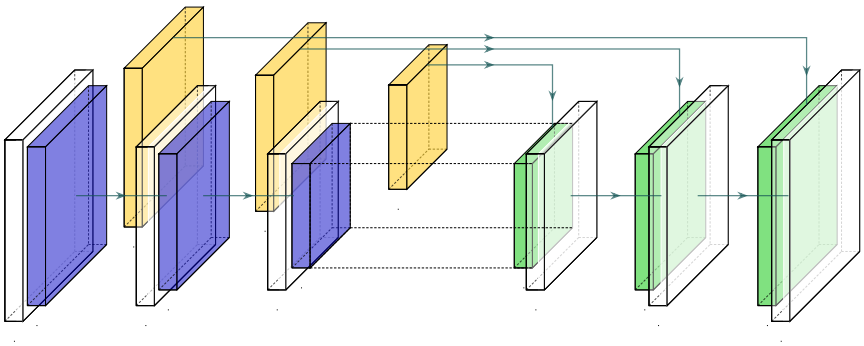


Figure 3.5: A novel architectural diagram for achieving photo-realistic style transfers. The first three layers consists of a convolutional layer (white) followed by a max pooling layer (blue). The max pooling masks are indicated in yellow with the unpooling layers being green. Adapted from [Li *et al.* 2018].

Li *et al.* [2018] aptly named their model PhotoWCT since it was primarily based on the WCT model by Li *et al.* [2017c]. PhotoWCT uses a technique familiar to a branch of neural networks known as auto-encoders. The input image is gradually condensed along the computational hierarchy within the encoding segment of the network before being expanded within the decode segment. This is achieved by using a convolutional layer followed by a max pooling layer within the encode segment and an unpooling layer followed by a convolutional layer within the decode segment. This is visualised within Figure 3.5. The original WCT model use upsampling techniques within the decode segment however details structures are not recovered. Li *et al.* [2018] show the need for using a max pooling masks within the decode segment on the unpooling layers. This provides spatial information to the decode segment that would be ordinarily lost. Thus, ensuring that the image can be reconstructed with fine details. It is important to note that the input image's representation has to be maintained along the entire network such that a reconstruction of the image becomes the output of the model.

After the image is pushed through the network, Li *et al.* [2018] describe the output image as having

inconsistent stylisation within certain regions of the image. This translates to uniform regions such as the night sky having partly light and dark blue areas instead of a uniform blue. Li *et al.* [2018] introduce the use of a smoothing function that will use pixel affinities to maintain similarities between the original content image and the output produced from PhotoWCT. Local neighbourhoods of pixels within the content image should be styled similar while not deviating significantly from the output of PhotoWCT. These two constraints will create a smoothness within uniform regions while maintaining the initial stylisation.

3.3 Semantic Mappings

The convolutional neural network is built for image classification and performs particularly well for this task however, it becomes ill suited on image generative tasks such as style transfer where fine details need to reconstructed [Champandard 2016]. This is due to the fact that ConvNets are able to extract semantic information for classification but is hardly utilised for generative tasks. Semantic information are parts of the image which logically belong together. This can be hair, skin tone or facial features within portraits or certain vegetation within landscapes.

Semantic mappings has shown success in applying colour information to image which are gray-scale [Chia *et al.* 2011]. The authors instruct users to label the image and segment the background and foreground of objects in the image which creates a semantic mapping. This information is used to gather a large corpus of images relating to the label from image sharing websites. Only images which match the foreground objects and background relative to the original gray-scale image are selected. These semantic mappings help the model understand the global structure of the image and will then match local colour structures within the images.

By building on this idea, Champandard [2016] propose using VGG-16 for neural style transfer but augmenting it with semantic information. The use of the semantic information will be exploited during the reconstruction of the image and constrain the output to build on top of these features. Here, the semantic mappings can contain any number of channels provided they are the same for the style and content image. It is important that each image has to have its own semantic map of the same aspect ratio. At each layer of the ConvNet, the semantic channels are concatenated. This guides the neural network to preserve these semantic features and bound the stylisation process within these areas. An example of the input image and the semantic mapping can be seen in Figure 3.6a and Figure 3.6b, respectively.

(a) Renoir, Pierre-Auguste. *The Banks of the River*.

(b) Semantic mapping of *The Banks of the River* painting by Renoir.

Figure 3.6: An example of an input image and its corresponding semantic mapping generated by Champandard [2016].

The semantic mapping outlines four distinct areas of the original image which are the river, river banks, tree and sky. Colour is used to differentiate the semantic information which will bind the stylisation process to these areas. These semantic mappings are needed for both the content as well as the style image. The generation of these semantic mappings are somewhat cumbersome but show that the model becomes less sensitive to style weighting [Champandard 2016]. The use of semantic mappings increase the amount of memory the computer will need but produce pleasing results when the content and style image are completely different and specific details need to be reconstructed. Content-aware style transfer is explored by Chen and Hsu [2016] while using the neural style transfer algorithm by Gatys *et al.* [2015]. Two improvements to the algorithm are made. The authors propose a masking out process to constrain the transfer layout as well as improve content characteristics and coherency by defining a new style feature from high-order statistics. In a similar method to Champandard [2016], the predefined semantic maps are concatenated at each layer however, the entries in these maps are simply binary indicators. If an entry is a 1 then this indicates that pixel in the corresponding output layer of the ConvNet should be fully aggregated in the cross-feature dependency and then back-propagated to previous layers. An entry of 0 indicates that the pixel removed and can be replaced during the image stylisation. This new semantic map guides the stylisation process and instructs the model where style transfers should occur. In order to introduce high order statistics, the feature maps defined at each output of the network is transformed. Here, the feature map is convolved with the Toeplitz matrix constructed by high-order filter coefficients. Content-aware style transfer is able to characterise semantic structures from the content image more effectively without overly smoothing out the reconstructed image, seen in [Champandard 2016].

3.4 Transferring Multiple Styles

Neural style transfer is achievable however, separate models need to be trained for a particular style. This is inflexible for applications that need to respond to user demands. Training separate models for similar styles becomes redundant. Artistic movements such as impressionism share very similar attributes. This can be seen in the short, thick paint strokes. Chen *et al.* [2017] propose a model which would be able to learn and store multiple styles. Chen *et al.* [2017] augment the architecture of the convolutional auto-encoder to include a *StyleBank*. Multiple convolutional filter banks are placed in sequence where each filter bank corresponds to a specific style. In order to transfer style, a forward pass through the encoder segment of the autoencoder is performed producing multiple feature maps, this is subsequently convolved with the corresponding filter bank before it is finally reconstructed by the decode segment.

The novel architecture is able to improve flexibility, control the style transfer process and even create fusion styles. Reconstructed images could possess elements of a van Gogh painting while still in the style of Picasso. Chen *et al.* [2017] illustrates the ability to explicitly create a representation of style.

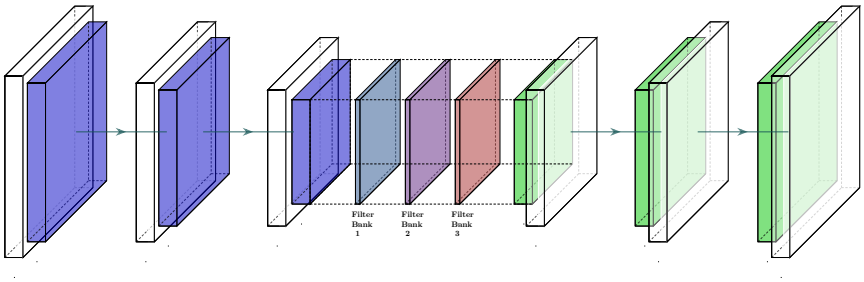


Figure 3.7: An novel architectural diagram for StyleBank. The first three layers consists of a convolutional layer (white) followed by a max pooling layer (blue). The sequence of layers are called the encoder. The subsequent section consists of a number of convolutional filter banks corresponding to specific styles. The final segment consist of three convolutional layers and its unpooling layer (green). Adapted from [Chen *et al.* 2017].

Each specific style is encoded in their convolutional filter bank. The model will contain a number of filter banks. In order to create a linear fusion of style, a weighting parameter can be placed on each filter bank before the subsequent convolutions with the output of the encoder. These weighting parameters should range between zero and one and the summation of all of them should be one. This will ensure that a linear combination of styles will be fused.

Chen *et al.* [2017] demonstrate that the model is able to create region based style transfer thus the reconstructed image will fuse styles within specific regions. The encoder is able to decompose the content image into multiple feature maps, which essentially creates logical regions within the image.

These regions could be based on a combination of colour, edges and or textures. Further, a clustering algorithm can be used to create region masks. The combination of the region mask and feature map with the applied convolution of varying filter banks will create regions that correspond to specific styles.

3.5 Locomotion Style Transfer

Gatys *et al.* [2015] state that their work should be viewed as a way to propel the algorithmic understanding of artistic imagery creation as well as the perception of that imagery. This same ideal can be extended to that of human-like locomotion.

Holden *et al.* [2016] builds on this research by applying it to the field of animation. The fact that Gatys *et al.* [2015] illustrates the *Gram matrix* from the hidden layer of the neural network encodes the artistic style of an image, Holden *et al.* [2016] were able to adapt the notion of artistic style transfer on images to motion data and produce a motion that retained the timing and content of the input but has the walking style of a reference motion. This deviates from the traditional image processing to a new motion domain. Holden *et al.* [2016] illustrates this style transference using the walking style of zombie, depressed, old person and injured.

The motion manifold contains the essence of the walking style and is found over the hidden units $\mathbf{H}$ of the network. Holden *et al.* [2016] employ a convolutional autoencoder to learn this motion manifold. The forward operation (encoding) is defined by Φ whereas the backward operation (decoding) is defined by $\Phi^{\dagger}$. The encoding segment consists of a convolutional layer, max pooling layer and non-linear *ReLU* layer. An unpooling layer and convolutional layer define the decoding segment. Formally, the forward operation of the network can be defined below:

$$\Phi(\mathbf{X}) = ReLU(\Psi(\mathbf{X} \circledast \mathbf{W}_0 + \mathbf{b}_0)) \tag{3.12}$$

where $\mathbf{X}$ is any input vector in the visible space, the convolution function is denoted by $\circledast$, $\mathbf{W}_0$ is the weight matrix and the bias is $\mathbf{b}_0$. The max pooling operation is indicated by Ψ.

The backward operation of the network can be seen below:

$$\Phi^{\dagger}(\mathbf{H}) = (\Psi^{\dagger}(\mathbf{H}) - \mathbf{b}_0) \circledast \tilde{\mathbf{W}}_0 \tag{3.13}$$

where $\Psi^{\dagger}$ is the inverse pooling operation over the hidden units $\mathbf{H}$. The bias term $\mathbf{b}_0$ is subtracted before the convolutional of the weighting matrix $\tilde{\mathbf{W}}_0$. Simply, $\tilde{\mathbf{W}}_0$ is the reflection of $\mathbf{W}_0$ on the temporal axis

and further transposed on the first two axes. The style transfer is constrained over hidden units **H** by the following

$$\mathcal{L}_{style} = \alpha||G(\Phi(\mathbf{S})) - G(\mathbf{H})||_2^2 + \beta||\Phi(\mathbf{C}) - \mathbf{H}||_2^2 \tag{3.14}$$

where **C** and **S** is defined as the content and style of the produced output, respectively. The weighting parameters of the content and style are defined by α and β and the function G computes the Gram matrix. Holden *et al.* [2016] state that, in the context of locomotion, the Gram matrix can be considered as the average similarity or co-activation of the hidden units. Hence, they define the function G to be

$$G(\mathbf{H}) = \frac{\sum_i^n \mathbf{H}_i \mathbf{H}_i^T}{n} \tag{3.15}$$

where i is the temporal domain.

Holden *et al.* [2016] illustrate that neural style transfer is possible between locomotion sequences. They are able to transfer the style of the character's locomotion by using stylised locomotion sequences where the character walks in a zombie style, depressed style, old man style and injured style.

3.6 Conclusion

The great artists of human history have finely crafted their skill to create unique and wonderful artistic compositions. This can be seen especially within the medium of painting. Artists create sensorial experiences by mastering the interplay between style and content. Neural style transfer seeks to algorithmically separate and transfer the style and content between images. Pioneered by Gatys *et al.* [2015], the field of neural style transfer has seen a meteoric rise in recent years due to its applications within industry. By isolating the content and style representations within the convolutional neural network, a new image can be synthesised which includes the style of an artist's painting and the content of an image. This is achieved by minimising a loss function across the individual style and content losses with varying weighting.

The neural style transfer algorithm by Gatys *et al.* [2015] performs well when the reference stylistic image is a painting or artwork. The newly synthesised image is expected to be less realistic and more of a hybrid between the real world and the original artist's imaginary world. It is observed that local structures within the content image such as buildings tend to appear wavy or textured in the newly synthesised image. Luan *et al.* [2017] introduce two novel concepts to help achieve photorealistic style transfer. Luan *et al.* [2017] propose a regularisation term to be added to the loss function during optimisation.

This will constrain the reconstruction to prevent image distortions such that the image will be represented by locally affine colour transformations. Finally, semantic segmentation is introduced which guides the transfer process of local structure so to align content matchings. Li *et al.* [2018] illustrates the ability of the convolutional auto-encoder to learn of the manifold of style. The resulting image is described to have inconsistent stylisation within regions of the image. Li *et al.* [2018] show that a smoothing function is necessary to maintain similarities in content between the original image and the newly synthesised image.

The addition of semantic mappings in specific regions of the images allow the transfer process to maintain rich semantic information. Semantic mappings are laborious and require an intense pre-processing step to label both the content and style images. This transfer process is inefficient when the content and style are completely different since more memory is required.

The neural style transfer is transported into a whole new domain by Holden *et al.* [2016]. The authors demonstrate the ability to learn the style and content representations of locomotion sequences and how they can be manipulated to create newly synthesised locomotion. A convolutional auto-encoder is defined to learn the manifold of human locomotion. By similarly constraining the loss function to Gatys *et al.* [2015] during the optimisation process, a sequence can be created which includes the style and content of two different locomotion sequences. Holden *et al.* [2016] demonstrate this ability by transferring the zombie style, depressed style, old man style and injured style to a character's locomotion.

Chapter 4

Research Methodology

"Create your own method. Don't depend slavishly on mine.

Make up something that will work for you! But keep breaking traditions, I beg you."

- Konstantin Stanislavski

This chapter will outline the objectives this research is investigating. The previous chapter provides an in-depth review of the relevant research relating to neural style transfer. This will form the basis for locomotion style transfer.

A clear, focused definition of the research is presented in Section 4.1 by exploring a set of objectives that will be explored through further investigation. Section 4.2 provides the research methodology which will be followed. The data collection and preparation is outlined in Section 4.3. Section 4.4 will illustrate the locomotion style transfer process. This section delve into the details surrounding the construction of the neural network as well as document the computing environment and considerations needed. Finally, a testing and evaluation section is provided in Section 4.5.

4.1 Research Objectives

The aim of this research is to apply the techniques of neural style transfer in the field of character locomotion. As outlined above, artificial neural networks demonstrate their capabilities for character locomotion style transfer. Hence the following objectives can be made:

Objective 1:

A human character's locomotion style is transferable using a neural network in the same way that Gatys *et al.* [2016] was able to transfer style between images.

Objective 2:

A trade-off between the style and content weighting within the total loss function (Equation 4.6) exists and by varying these parameters, the newly synthesised locomotion adapts accordingly.

Objective 3:

A running gait poses difficulties for locomotion style transfer as compared to the walking gait. Therefore, the stylised motion manifold is harder to transfer for different locomotion gaits for human characters acting like a zombie.

4.2 Research Process

The following section will outline the key phases of development in the life cycle of this research. Each phase is defined by a set of unique activities that will collectively contribute to its conclusion. Figure 4.1 provides a visual description of the methodology adopted on this research.

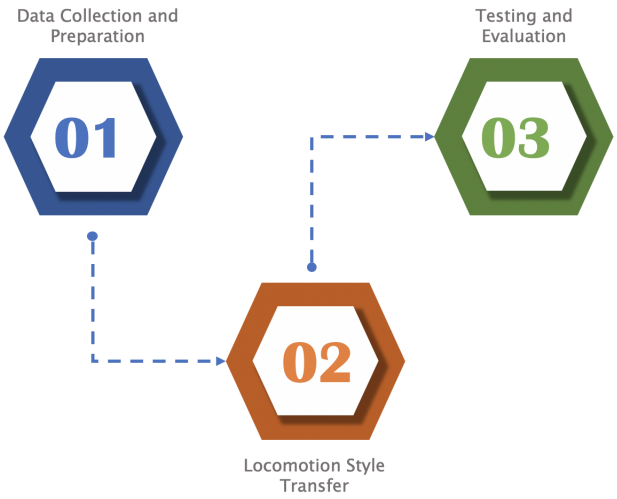


Figure 4.1: The research methodology.

The methodology has been divided into key phases of development. The first phase involves the data collection and preparation phase which is outlined in Section 4.3. This section explicitly outlines the entire process needed to transform the raw data into the final dataset that will be used during modelling. The motion capture data that is used by the model is collected from open-source research resources. Section 4.4 describes the entire modelling process from the mathematical inception to the implementation. The approach used to achieve style transfer between characters is investigated within this section. The final phase of development, testing and evaluation, is presented in Section 4.5.

4.3 Data Collection and Preparation

Motion capture data is used during this research. Motion capture is the process of transferring the motion from an actor to a 3D model. Points are strategically placed on the actor's body. These points are tracked by cameras or sensor equipment and relayed to a computer by translating it to a three dimensional grid as the actor moves around. The data used in this research was created and curated by Lab [2017].

4.3.1 Data Selection

Carnegie Mellon Graphics Lab, University of Carnegie Mellon, has created a database of motion capture data. The data has been created using 12 Vicon infrared MX-40 cameras [Lab 2017]. The human would wear a black jumpsuit and 41 small grey markers placed on their body. The cameras detect the markers using infrared. The images are recorded separately, thus they have to be triangulated to create 3D data.

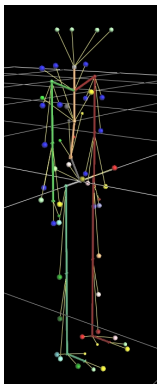
The Graphics Lab presents the data in two ways

- **Marker Positions** - this is a file of marker positions within a 3D space, i.e. X, Y and Z positions are provided for each marker.
- **Skeleton Movement** - the skeleton movements are presented in two different types of presentations

    1. Skeleton and its joints - This describes the connections lengths, degrees of freedom and mathematical transformations
    2. Movement Data

The skeleton movement data will be used since there is information about how each marker is related to each other. The Graphics Lab has indexed their database against the type of motion namely, running, walking, jumping and varied.

(a) The set of markers used to track the movement of the human actor.



(b) A visualization of the skeleton movement file mentioned previously. The balls represent the markers where as the coloured connections represent the bones.

Figure 4.2: The recording process of motion capture from the real-world to its corresponding digital representation [Lab 2017].

4.3.2 Data Preparation

Carnegie Mellon Graphics Lab recorded their motion capture data in a Biovision Hierarchy (*bvh*) format. Each recording captures the motion as well as the actor's skeletal hierarchy. Thus, there are two sections of a standard *bvh* file, namely the hierarchy and motion. The hierarchy section describe the layout of the skeleton. It labels the individual joint as well as provides information as to the specific channel that was recorded. Channels that are typically recorded are the x, y and z co-ordinates but also the x, y and z rotations. This hierarchical information as well as the order is important when trying to extract the motion data. A single frame is a snapshot of an actor at a period in time. A sequence of many frames will show the actor moving. Each frame represents a line within the motion section which has been delimited by a space. The motion data is placed in an order corresponding to the order within the hierarchy section. The motion section of the *bvh* file also includes two crucial pieces of information namely, the frame time and the number of frames. This metadata informs later rendering tools about the conditions of the recording. Since there are many files in the CMU Graphics Lab Motion Capture Database, which describes different motions played by different actors, the following data transformations had to occur in order to allow for a standardised dataset.

Standardising floor height

The *bvh* file format has a major disadvantage, in the fact that it does not record a basis pose or a ground to place the object on. Therefore, it is important to determine the floor height of the object and

subtract it. This will ground the object and standardise it across all the files.

For each frame in each file, the minimum y co-ordinate is determined for the left and right foot base and toe, respectively.

The smooth maximum function described in Equations 4.1 was used to determine the floor height [Lange *et al.* 2014].

$$\mathcal{S}_\alpha(x_1, \ldots, x_n) = \frac{\sum_{i=1}^{n} x_i e^{\alpha x_i}}{\sum_{i=1}^{n} e^{\alpha x_i}} \tag{4.1}$$

where α is the the largest possible y co-ordinate of the left and right foot. Since, the minimum height is required, as simple inversion of the function if used.

Joint Selection

The CMU Graphics Lab record a plethora of information regarding the various joints/markers within the human body. For the purposes of this research, only the standard joints were selected. These joints make up the most basic human skeleton and will be easily recognisable when recreated on rendering software. A list of the selected joints are provided below:

Joints	
Central Hip	Neck
Left hip	Head
Right hip	Left shoulder
Left knee	Right shoulder
Right knee	Left elbow
Left foot (heel)	Right elbow
Right foot (heel)	Left palm base
Left foot (toe)	Right palm base
Right foot (toe)	Left hand index finger
Lower spine	Right hand index finger
Upper spine	

Table 4.1: The selected joints when reconstructed creates the basic human skeleton.

A reference joint is added to the above list of joints. This is a fictitious joint which is derived by applying the Hadamard Product to a selected joint to remove the vertical movement as the joint should be at floor height [Liu and Trenkler 2008]. The Hadamard Product is a simple element-wise multiplication operation conducted on the central hip joint and some filter matrix. The Gaussian filter, or commonly known as the Gaussian blur, is then applied to this newly created reference joint. The Gaussian function

is defined in one dimension by

$$g(x) = \frac{1}{\sqrt{2\pi\theta^2}} e^{-\frac{x^2}{2\theta^2}} \tag{4.2}$$

where θ is the standard deviation. Thus, the convolution function in Equation 2.10 can be applied to the joint to create a filtered reference joint. Applying the Gaussian filter removes any high frequency movements from the joint in which this reference joint was based on. This joint is used to track the central movements of the character within a scene.

Standardising frames per second

As stated above, frames should be sequenced in order to make sense of the motion. Therefore, the frame time is an important concept which encodes the time between frames. This tells the rendering software the time interval between frames. Using a frame time of 0.008333, the frames per second can be determined.

$$\begin{aligned} FPS &= \frac{1}{0.0008333} \\ FPS &= 120.0480192076831 \end{aligned} \tag{4.3}$$

Therefore, there are 120 frames in a single second of recorded motion. In this research, the standard 60 frames per second (FPS) will be used. This standardisation is important to ensure that all the data extracted from the *bvh* files are of equal time.

Foot Contacts

The velocities of the left and right foot joints (toe and heel) were calculated. This is done by subtracting subsequent foot joint's position from the previous joint's position along the x, y and z axis.

Should the foot joint's position fall below a certain height and have a small enough velocity, this will be considered to be in contact with the ground. This is a binary response for each frame for the left and right toe and heel joint. This is added to the input representation. A similar technique has been used by Lee *et al.* [2002] in their work entitled, "Interactive Control of Avatars Animated with Human Motion Data". The authors show that using this information in conjunction with the x, y and z co-ordinate data of the character is possible.

Velocities

In order to calculate the forward direction of the character, the distance between the left and right shoulders and hips must be calculated for each frame within the sequence. By further, averaging this

vector and applying the cross product along the Y axis, the forward direction can be calculated. The Gaussian filtering process above, is used to remove high frequency movement and smooth the positions.

The input representation also includes the rotational velocity around the vertical axis for each frame within the sequence as well as the global velocity in the XZ-plane.

4.3.3 Data structure

For ease of training and standardisation of the model, a fixed window size is introduced. The length of raw motion clips can be of variable size, so it is important to segment these clips into a fixed window size which the model can ingest. For the purposes of this research, the standard window size of 240 frames was selected [Holden *et al.* 2016]. That is, for a single clip can be represented as $\mathbf{X} \in \mathbb{R}^{n \times d}$ where n is the window size that corresponds to the number of frames within a clip and d is the degrees of freedom which are described above. This research will use $d = 73$ since the first 66 degrees corresponds to the x, y and z co-ordinates of the 22 select joints. The next two degrees of freedom comes in the form of the global velocity along the x axis and z axis. The next degree of freedom comes in the form of the rotational velocity along the vertical axis (y axis). Finally, the last four degrees of freedoms are attributed to the foot contact for the left and right toe and heel joint.

To create the locomotion database, single clips are stacked along a new dimension. This makes referencing clips much easier. A visualisation of the resulting dataset after the above transformations is presented in Figure 4.3.

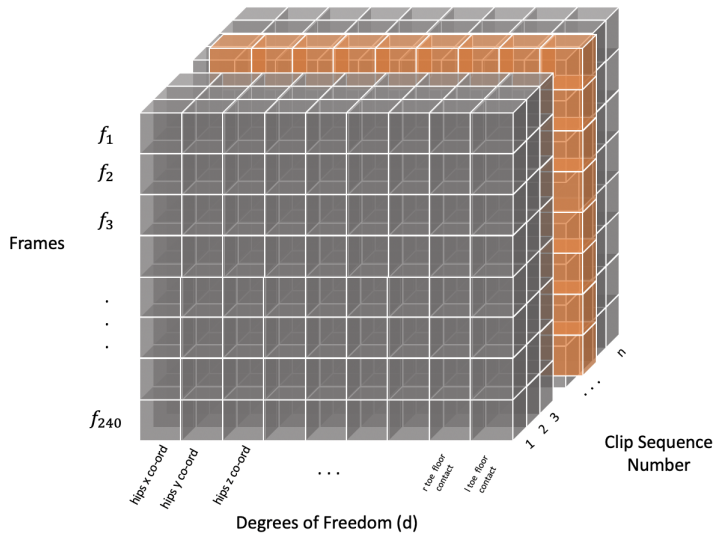
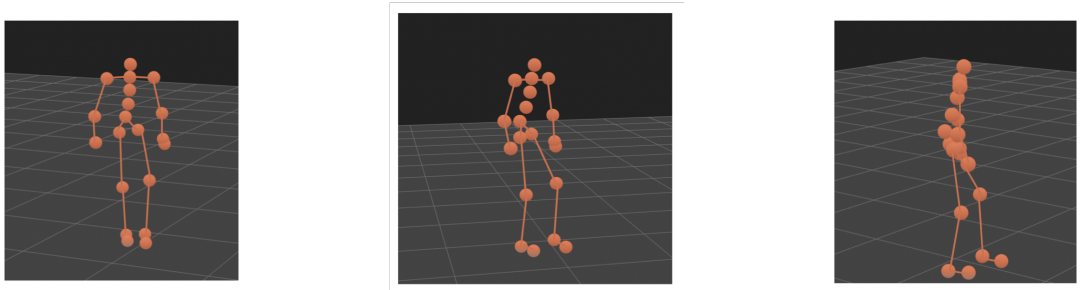


Figure 4.3: A visual representation of the extracted motion dataset after the transformations with a single clip being selected.

The first axis corresponds of the clip sequence number whereas the secondary axis is the frames which has been fixed to the window size for a single clip. The degrees of freedom make up the final axis which is of size 73.

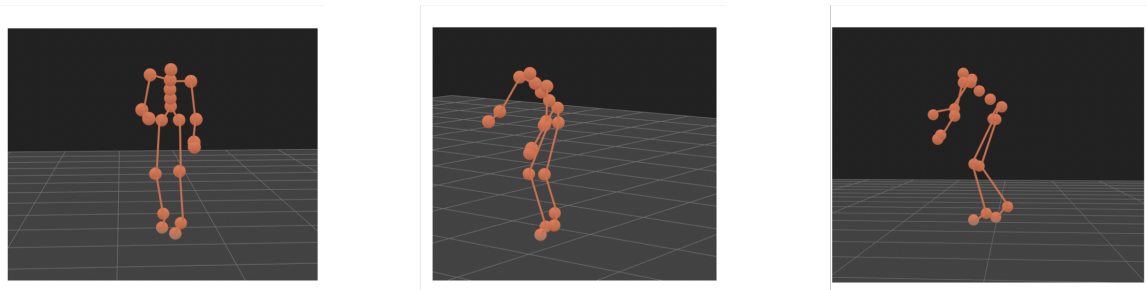
4.3.4 Style and Content Datasets

Since this research is focused on neural style transfer of locomotion, two important dataset arise that is, the style and content dataset. The content dataset consists of several extracts from *bvh* files of actors walking in a naturalistic human manner where they follow a particular path and have an up-right gait. The style dataset consists of stylised motion extracted from several *bvh* files centred around a specific theme. This theme corresponds to a specific style transfer, namely zombie's walking. An extract from the two datasets can be visualised in Figure 4.4 and Figure 4.5.



(a) (b) (c)

Figure 4.4: A single frame extracted from the content dataset and rendered from various angles.



(a) (b) (c)

Figure 4.5: A single frame extracted from the style dataset and rendered from various angles.

Figure 4.4 and Figure 4.5 illustrate the impact of style on locomotion. The actor's gait is projected forward, since the axis of the spinal column is offset in a forward direction. There is also a slight curvature of the spine, which can be seen in Figure 4.5c. These traits indicate that the actor is not rooted in their stance and their hips are not progressing their motion. This is in direct contrast with Figure 4.4 where the actor has opted for a more naturalistic gait seen through almost straight line created by the

spinal axis. Winter [1991] state that human gait is centered around the pelvic bone and thus, when the spine is projected forward a flexion is created. The degree of flexion is important and the actor in Figure 4.5 as opted for a greater flexion.

The arms of the actor in Figure 4.5 are seen to be a distance from the actor's body, as if they were trying to grab something. This is very typical with actor's portraying the zombie style in film. The zombie is seen as the predator where as humans would be the prey therefore they would place their arms in front of them to try and catch their prey. This is evident in Figure 4.5. The actor within Figure 4.4 place their arms to their side. This is seen as a more natural locomotion. Arms are used to progress the motion and stabilise a stance in bipedal animals [Winter 1991]. The elbow joint is seen to be in line with the spinal column in Figure 4.4c. Deviations in the locomotion style from the natural human being is seen as threatening to our perception thus the style of the zombie tries to directly contrast natural motion creating a menacing predator-prey cinematic experience.

4.4 Locomotion Style Transfer

The previous section outlines the entire data collection and transformation process applied during this research. At this point, the data is structured in correct format and ready to be used during the training process. This phases aims to design and implement locomotion style transfer. Section 4.4.1 defines the deep learning framework which will be used to achieved locomotion style transfer while Section 4.4.2 will outline how the model is able to learn the manifold of human motion. This section will round off with a discussion around the computing environment used during this phase. A brief impact of hardware on the machine learning training process is presented as well as the libraries and packages used within this research.

4.4.1 Constructing the Motion Manifold

Holden *et al.* [2015] construct an amalgamation of two computational neural networks namely, the convolutional neural network and the autoencoder. These two neural networks have been traditionally used for vastly different tasks, however Holden *et al.* [2015] propose a hybrid architecture for learning the human motion manifold.

A visualisation of the proposed architecture can be seen in Figure 4.6.

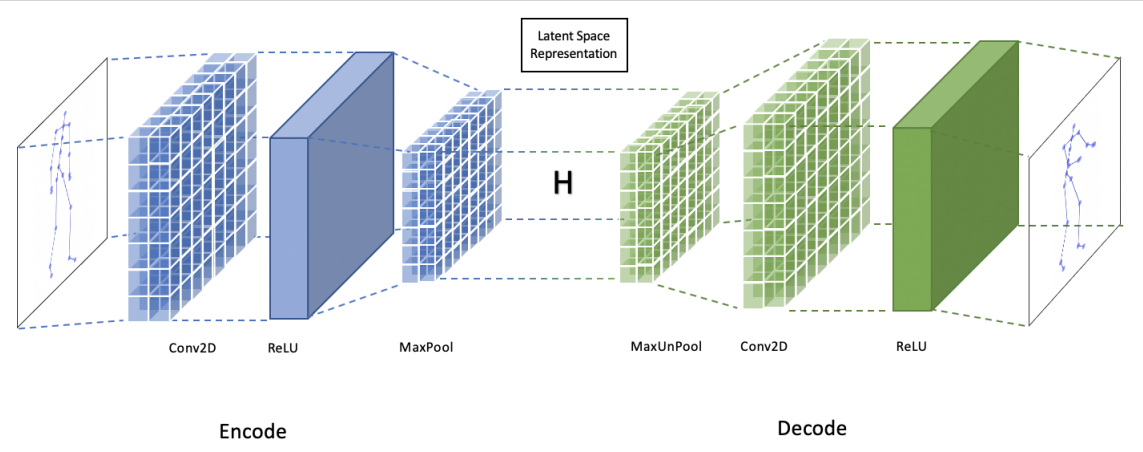


Figure 4.6: A conceptual rendering of the convolutional encoder used for locomotion synthesis. Adapted from [Holden *et al.* 2015].

The input data flows through an 'encode' segment followed by the 'decode' segment. This is a typical architecture of the autoencoder, however Holden *et al.* [2015] propose using a convolutional and max pooling layer which is typically reminiscent of the convolutional neural network. The encoding sequence of the network is defined by a two-dimensional convolutional layer, an activation layer using the nonlinear operation ReLU and finally the max pooling layer. At this point the model has learned the characteristic of human motion and **H** is defined as the latent space of human motion representation. The next segment of the network is known as the decoding segment. Here, **H** follows a process of max unpooling, two-dimensional convolution and finally passed into an activation layer using the ReLU operation.

The mathematical definition for the encoding segment of the network can be described as a function Φ on input $\mathbf{X}$ by

$$\Phi(\mathbf{X}) = \Psi(ReLU(\mathbf{X} \circledast \mathbf{W} + \mathbf{b}_0)) \tag{4.4}$$

where $\circledast$ denotes the convolutional function defined in Equation 2.10 using the weight matrix $\mathbf{W} \in \mathbb{R}^{m \times d \times w_0}$. A bias is included as $\mathbf{b}_0$ and Ψ denotes the max pooling layer defined in Section 2.3.3. The activation layer ($ReLU$) uses the nonlinear operation defined in Equation 2.6. Here, m is set to 256 and Holden *et al.* [2016] experimentally found this to provide good reconstruction of human motion where as w_0 corresponds to the filter width and was aptly set to 25. The output of this segment is **H**.

The decoding segment of the network can be mathematically defined as a function $\hat{\Phi} \in \mathbb{R}^{\frac{n}{2} \times m}$ over

the latent space representation of human motion **H**

$$\hat{\Phi}(H) = ReLU((\hat{\Psi}(H) - \mathbf{b}_0) \circledast \hat{\mathbf{W}}) \tag{4.5}$$

where $\hat{\Psi}$ is the inverse operation of the pooling operation which aims to expand the representation thus broadcasting it to its original size. This is then subtracted from the bias $\mathbf{b}_0$. The convolution function is applied with the weight matrix $\hat{\mathbf{W}} \in \mathbb{R}^{d \times m \times w_0}$ where the first two dimensions have been inverted.

Hinton [2012] provides a guide to initialising the weighting matrix **W** using small random numbers which have a zero mean Gaussian distribution and a standard deviation of 0.01. Hinton [2012] suggest using the 'fan-in' and 'fan-out' principle which was applied when setting the weighting matrix.

4.4.2 Training the Network for Locomotion Style Transfer

Now that the convolutional autoencoder is designed, a mechanism to learn the manifold of locomotion and transfer style is required. This will allow the model to approximate the locomotion of a character as well as the style representation of stylised locomotion. The weighting matrix **W** defines the shape of the model, Φ and $\hat{\Phi}$, within a multi-dimensional space, thus by iteratively updating the weight matrix, the model will be able to learn motion. A function should be pre-defined and placed upon the updating process so to constrain the possibilities of weights such that the desired results are produced.

Style Transfer Cost Function

As Gatys *et al.* [2015] state, the artistic style of an image can be located within the hidden layers of neural network using the Gram matrix defined in Equation 3.3. Thus, by editing the latent space representation of human motion **H** within a locomotion, style transfer is possible.

Using this information, Holden *et al.* [2016] define a constraint or *cost function* over the latent space representation **H** as follows

$$\mathcal{L}_{style}(\mathbf{H}) = \alpha \|G(H) - G(\Phi(X_{style}))\|_2^2 + \beta \|\mathbf{H} - \Phi(X_{content})\|_2^2 \tag{4.6}$$

where G is the function which produces the gram matrix defined in Equation 3.3, X_{style} and $X_{content}$ are the input locomotion data which contains stylised motion and a normal, relaxed locomotion. Two weighting parameters are given in the form of α and β which dictate a relative importance given to

content and style.

$$\alpha = \frac{s}{c+s} \tag{4.7}$$

$$\beta = \frac{c}{c+s} \tag{4.8}$$

Here, c and s refer to the amount of content and style, respectively.

4.4.3 Computing Environment

This section will outline the computing environment that this research utilised during development and collection of results.

Impact of hardware on machine learning

The new computing paradigm which is machine learning, specifically deep learning, requires massive amounts to computational power. For example, the machine learning model which beat the famous European Go champion, Mr Fan Hui, took 4-6 weeks to train [Silver *et al.* 2016]. Therefore, the computing environment and the choice of hardware are at the utmost importance. Considered to be the "brain" of the modern computer, the central processing unit (CPU) has been used to trained neural networks. The CPU receives instructions sequentially by the operating system which is simply a numerical calculation. Most modern CPU's have multiple cores in order to process tasks in parallel thus speeding up the time of computation. The CPU is designed to provide fast response times for a single task for a wide variety of applications. Emerging from the field of video rendering and gaming, this computational processing unit has been widely used within deep learning for its ability to process tasks in parallel. The Graphical Processing Unit (GPU) is designed for graphical applications which require parallelism, as each pixel on a screen can be processed independently. The GPU has been able to decrease the the training times of most neural networks. Lee *et al.* [2010] show that the performance increase between a CPU, Intel Core i7 960 processor, and the GPU, Nvidia GTX280 processor, averages to 2.5 times faster for different types of workload ranging from fast Fourier transforms to collision detection algorithms. The performance of the GPU would vary depending on the task at hand as well as the processor itself.
For these reasons, the main computation will be performed on a GPU.

Cloud Computing

A popular buzz-word within the IT community, cloud computing has emerged to become more than a mere dream. Cloud computing has enabled software engineers with innovative ideas to get to market much quicker without the need to have large capital outlays [Armbrust *et al.* 2010]. Large scale organisations are embracing the cloud computing platforms to streamline their IT infrastructure to easily pivot to fit the needs of their customers [Matthews 2019].

Cloud computing is delivery of powerful computing engines, storage and applications over the Internet to end users whom subscribe to a pay-as-you-use service. Armbrust *et al.* [2010] outline three pertinent reasons as to why an individual or corporate might adopt the services of cloud. The first reason is the on-demand factor of these services. Companies with needs for temporary or sporadic demands, need not provision an entire data centre leading to under-utilisation. Companies can, instead, opt in for a rental services per day or hour. The second reason speaks to the fact that individuals are able to spin up computing environments instantaneously when demand is unknown in advance. Finally, the last reason is a associated with cost. Companies are able to utilise high performing hardware to finish a computing task in a shorter time allowing for faster delivery.

Google provides a free cloud computing environment, known as Colaboratory (Colab), with support for GPU computation. Any user is able to use this python based Jupyter Notebook development environment to train deep learning models. Carneiro *et al.* [2018] performed an analysis of using Colab for deep learning application in their paper entitled "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications". The authours concluded that using Colab was beneficial in accelerating many deep learning and GPU-centric applications, the allotment of hardware resources was perfectly suited for researchers and the Jupyter Notebook development environment was ideal for sharing knowledge. It is for these reasons that Colab is used as the computing environment for this research.

Libraries

Below are the packages/libraries used within this research to achieve locomotion style transfer.

- **Theano** is a python-based deep learning computing framework.
- **Numpy** is a python native library used to calculate mathematical operations on multi-dimensional matrices and arrays [Oliphant 2006].

4.5 Testing and Evaluation

The aim of this section is to set out the methodological approach undertaken to investigate the research objectives. Due to the nature of this research both qualitative and quantitative techniques were employed. This will allow for a robust understanding of the problem domain and fully investigate each research objective.

4.5.1 Qualitative

A qualitative research approach was chosen as a technique to investigate the first research objective. This research objective questions the possibility of transferring locomotion style between two human characters. By employing a qualitative approach, the core experiences facing the audience can become evident. "Qualitative research is especially helpful when it provides us with someone's perceptions of a situation that permits us to understand his or her behavior" [Krathwohl 2009, p. 230]. Character motion synthesis lies in the perception of the audience. Whether it be from an evolutionary process or a fine tuned ability, humans are able to rapidly pick up on subtle differences in naturalistic movement. Qualitative data are well suited for understanding the success of locomotion transfer.

Audience Selection

The popular survey and questionnaire platform, SurveyMonkey, was used to collect responses from 50 participants. The inclusion criterion was based on participants who are currently a citizen of the United States of America (USA). The choice of selecting participants from a single country was crucial as it would avoid cognitive bias or subtle varying cultural biases between participants. The USA is a powerhouse in the film and television industry, so it would be naturalised within its citizens. Figure 4.7 illustrates the percentage breakdown of the participants involved with this study by their geographical region within the United States of America. This indicates that there is a good spread of participants across the USA.

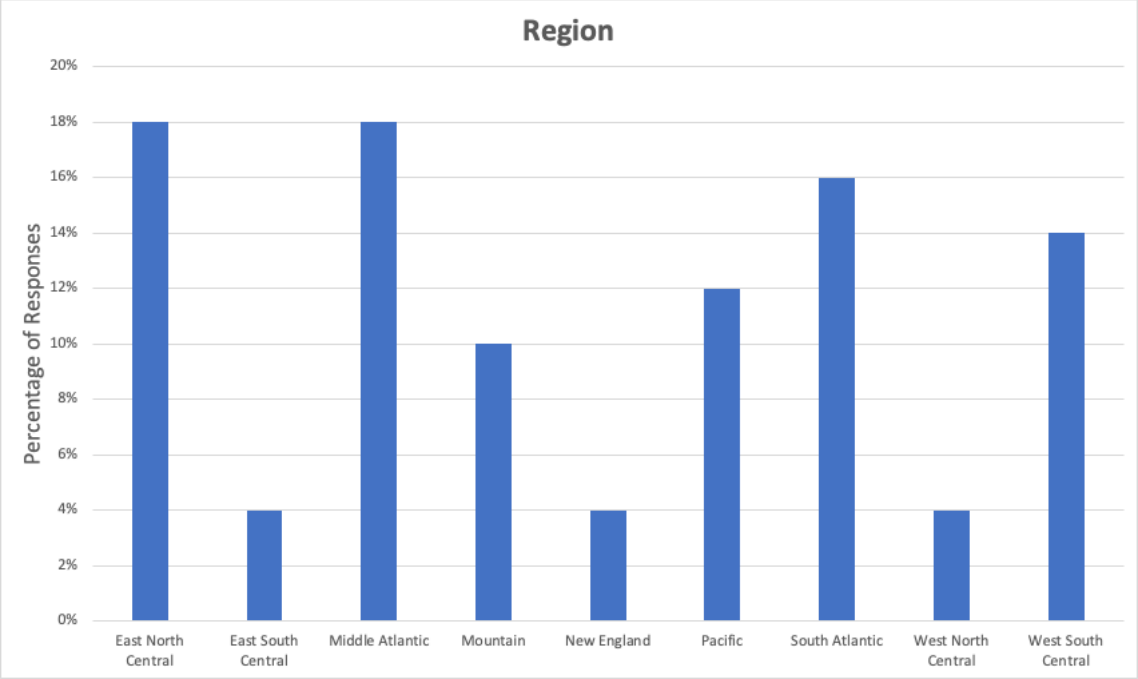
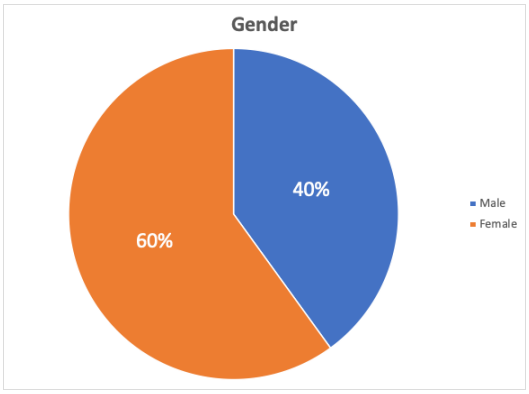


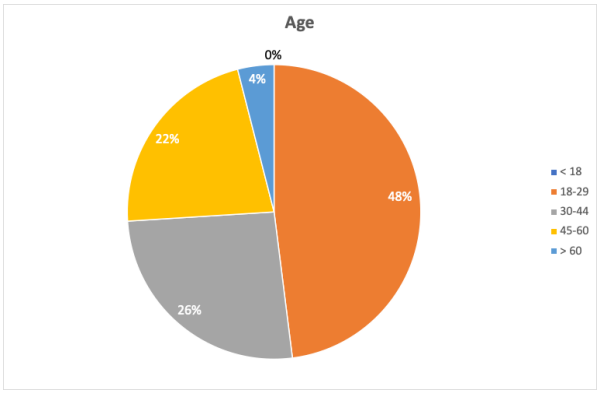
Figure 4.7: The breakdown of regions with the USA where the participants reside.

In undertaking this study, only participants who were above the age of 18 were considered due to the assumption that they would have been exposed to a vast array of film or television in their lifetime and not only a particular style of animation suited for children. It is also important to maintain the objective of this study. The age demographics of participants can be seen in Figure 4.8b. The age category of 18-29 encompasses the majority of participants. This breakdown is suitable for this research as animation films are generally consumed by younger audiences.

All genders were considered and this was not an exclusionary criterion. More females participated in this study than males and this can be seen in Figure 4.8a. That is, approximately 30 out of 50 participants that contributed to this study were female.



(a) The breakdown of gender of the participants included in this study.



(b) The breakdown of age of the participants included in this study.

Figure 4.8: The age and gender demographics of the participants of this study.

The distribution of household income bands across the participants in this study is displayed in Figure 4.9. SurveyMonkey collects this information for each participants as an entry questionnaire. This was not explicitly asked in this survey and is not an inclusionary criterion however it is provided by SurveyMonkey to understand the demographics of participants. A large number of participants fall within the $50000 - $74999 income band. This would indicate our participants are middle income.

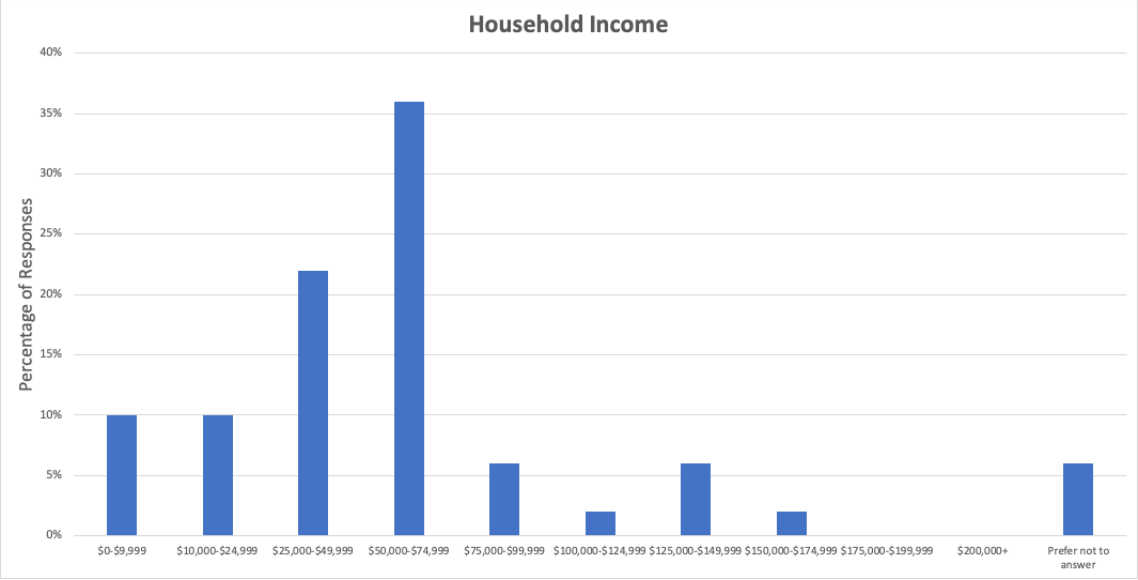


Figure 4.9: The breakdown of household income of the participants included in this study.

The participants used in this study are fairly well balanced. The variability in demographics allows for diversity in the responses.

Survey

The survey was completed on August 2020 on the SurveyMonkey platform. All interactions with participants were facilitated through the platform and no direct communication between participants and the researcher was established. All participants completed the survey using the platform remotely on their devices. Each participant was given three labeled videos. The first video was the content locomotion or a character with neutral walking gait, the second video displayed the reference style locomotion or a character performing a stylised gait and finally the synthesised locomotion generated from the locomotion style transfer process proposed in this research was presented in video 3. Participants were asked to rate the success of the transfer of style to the neutral character. The survey consisted of five questions which align to a specific locomotion style. The locomotion styles are as follows:

- Walking Zombie
- Running Zombie
- Walking Gorilla
- Running Gorilla
- Running Monkey

The content locomotion was kept the same throughout the questions to maintain a focus on the style transfer. A neutral walking gait was selected as the content locomotion. The questions were designed to encourage the participants to rate the likelihood of the synthesised locomotion's resemblance to the locomotion style on a four point rating scale. The first point represents a complete lack of resemblance to the locomotion style where as point four represents a complete resemblance to said style. An example of the rating scale for the walking zombie locomotion can be seen below.

1. Does not walk like a zombie
2. Barely walks like a zombie
3. Somewhat walks like a zombie
4. Walks like a zombie

The same rating scale was used across the five questions however adapted for the style transfer in question. A four point rating scale was chosen purposefully to prevent participants from remaining neutral. This forces participants to make a definitive decision on the success of a style transfer.

Style transfer is highly subjective in the sense that the generated locomotion is meant to exhibit elements from both the reference style and content without recreating either. For this reason, a disclaimer was placed at the bottom of each question informing the participants that the objective of this research is not simply to rate whether the generated locomotion is identical to the reference style and content locomotion but rather how well does it resemble the style.

4.5.2 Quantitative

Model evaluation is paramount to determine the efficacy of a model's predictive power hence determining the right evaluation criterion is no small task. Quantitative metrics were selected to evaluate the second and third research objectives. A quantitative approach will provide a grounding in the understanding of these investigations. Since this research delves into a fairly new field of research, very little to no techniques have been established for locomotion style transfer. Thus, techniques from the broader field of style transfer for images will be used. This section will highlight common image quality techniques used within computer vision to compare similarities between images. It is important to note that these metrics will be used in an unconventional manner. The objective of this research is not simply to recreate a locomotion but rather impart stylised elements into a character's walking gait. Therefore, the purpose is not to have identical locomotions since variations should exist. The use of these quantitative metrics is to simply gauge the level of variations and ensure a perfect score is not met. A perfect score would indicate no style transfer has been made.

Quantitative metrics

Two well known metrics, namely peak-signal-to-noise ratio (PSNR) and the mean squared error (MSE), stand out as obvious choices to evaluate the output generated from the deep learning framework.

Traditionally, PSNR is typically used as an image quality metric. PSNR is defined by the following mathematical formula for a given reference matrix, **f**, and test matrix, **g**, where $\mathbf{f} \in \mathbb{R}^{M \times N}$ and $\mathbf{g} \in \mathbb{R}^{M \times N}$

$$PSNR(f,g) = 10\log_{10}\left(\frac{255^2}{MSE(f,g)}\right) \tag{4.9}$$

where

$$MSE(f,g) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(f_{ij} - g_{ij})^2 \tag{4.10}$$

As MSE approaches zero, it can be seen that PSNR will approach infinity, thus larger values of PSNR translates to better quality and smaller values will indicate that there is a larger numerical dif-

ference. Used across of a wide variety of regression problems, Mean Squared Error (MSE) measures the average value of the squared difference between the actual observations and predictions estimated through a mathematical model or classifier. Unconventionally, this research will use the original content locomotion as the actual observation and the predictions would be considered as the newly synthesised locomotion. The MSE and PSNR would calculate the deviation of the synthesised locomotion from the original content locomotion.

This research will attempt to model the locomotion data in a similar fashion to images so that these image quality metrics can be used. As mentioned above, these metrics will be used in an unconventional way. A perfect score across these metrics are not desired, however they will simply provide a gauge on the deviation of the newly synthesised locomotion from the reference content locomotion. Should the metric produce a perfect score, this will indicate that no style has been imparted into the synthesised locomotion.

4.6 Conclusion

Locomotion style transfer is core to this research. By investigating the three research objectives, the possibility of locomotion style transfer can be determined. The research will implement a simple three-phase process. Each phase will guide the development of this research. Phase one will focus on the collection and preparation of the data. This is considered the most pivotal phase and if done correctly, it will ensure a seamless transition into phase two. This phase is focused on building the neural network to achieve locomotion style transfer. Phase three speaks to the evaluation of the result from phase two.

Motion capture data collected by the Carnegie Mellon Graphics Lab, University of Carnegie Mellon, was used on this research Lab [2017]. The university has a rigorous recording process and has categorized motion sequences based on the type of movement. To train a neural network, the raw motion capture data has to be transformed. These transformations come in the form of standardising floor height, joint selection, standardising frames per second, establishing foot contacts and calculating velocities. The resulting dataset is ready for the modelling phase.

An amalgamation of the convolutional neural network and the autoencoder is constructed to achieved locomotion style transfer. The style and content representations are learned within the hidden units of the network or otherwise known as the latent space of human motion representation. The optimisation process is constrained by defining a loss function over the latent space representation. This loss function will use the Gram matrix which has shown prowess in image style transfer [Gatys *et al.* 2015]. Weighting

parameters are included to control the amount of style and content the newly synthesised locomotion should include.

Model evaluation is the most pivotal part of a model's implementation. In order to investigate the research objectives, both qualitative and quantitative approaches were utilised. A survey was conducted on a group of 50 participants across the USA. The participants were asked to rate the success of the transfer process by being shown three videos. Video 1 displayed a neutral walking gait, video 2 presented the stylised locomotion and the synthesised locomotion generated from this research was provided in video 3. Each participant was asked to rate the synthesised locomotion on a four point rating scale. The qualitative approach allows for an understanding of the possibility of locomotion transfer style viewed from an audience lens. Quantitative metrics were employed to investigate the second and third research objectives from a rigorous mathematical perspective. Although this research is focused on locomotion, similar parallels can be drawn to the field of image processing by similarly modelling the data. One such extension is image quality metrics. This research will employ two well-known metrics namely, peak-signal-to-noise ratio (PSNR) and mean squared error (MSE). A perfect metric score is not desired in this research however an understanding into the deviation from the original reference content locomotion is required.

Chapter 5

Experiments and Results

"Art is not a mirror held up to reality but a hammer with which to shape it."

- Bertolt Brecht

In the previous chapter, the process of designing and building the locomotion style transfer framework is discussed. This chapter will delve into the investigations performed in this research. Three experiments were conducted in order to assess the research objectives outlined in the previous chapter. The results and analysis of the qualitative study is housed in Section 5.1. A quantitative approach is followed to investigate two important aspects of locomotion style transfer namely, the trade off between style and content and the effect of varying gait on the style transfer process. This is covered in Section 5.2 and Section 5.3, respectively.

5.1 Experiment 1: Neural Style Transfer for Locomotion Synthesis

The purpose of this study is to understand the success of the adaption of neural style transfer for human locomotion. Style transfer is highly subjective and value is placed in the eyes of the beholder. For this reason a qualitative study was conducted using 50 participants. Each participant was provided with three videos per question. The first video was the content locomotion or a normal walking gait of a character, the second video focused on the stylised locomotion and finally the synthesised locomotion from the proposed neural style transfer for locomotion was housed in video 3. All videos were a 3D rendering of the joints using the same colours and backgrounds so to avoid any biases. The survey consisted of five questions where each question centered around a different stylised locomotion. The stylised locomotions were as follows:

- Walking Zombie
- Running Zombie
- Walking Gorilla
- Running Gorilla
- Running Monkey

The content locomotion was kept the same throughout the questions to maintain a focus on the style transfer. A neutral walking gait was selected as the content locomotion. The participants were then asked to rate the success of video 3 as it relates to the stylised locomotion on a four point scale. The four point scale was purposefully selected to avoid neutral stances. An example of the rating scale for the walking zombie locomotion can be seen below.

1. Does not walk like a zombie
2. Barely walks like a zombie
3. Somewhat walks like a zombie
4. Walks like a zombie

At the end of each question the participants were provided with a disclaimer to rate the synthesised locomotion in video 3 based on the resemblance of the particular style and not whether it is an identical match. Style transfer is not meant to create identical matches from the reference style and content locomotions but rather impart elements from each. It is critical to note that the videos were a rendering of selected human joints recorded from a single angle. This angle was uniform across all videos, so to avoid any differing interpretations. Participants were not provided with the 3D renderings but rather static video clips. This was due to the limitations present on the SurveyMonkey platform.

The original 3D renderings for each stylised locomotion and corresponding synthesised locomotion can be found at the following link:

`http://www.bryceramgovind.ml/`[1]

[1]Optimised for Chrome, Safari, Firefox, and Microsoft Edge. Older versions of Internet Explorer will suffer from compatibility issues. WebGL is used for the interactive animation renderings. This could be blocked by some organisations.

5.1.1 Results

The first question of the survey centered around the zombie walking style. The responses from the participants can be found in Figure 5.1.

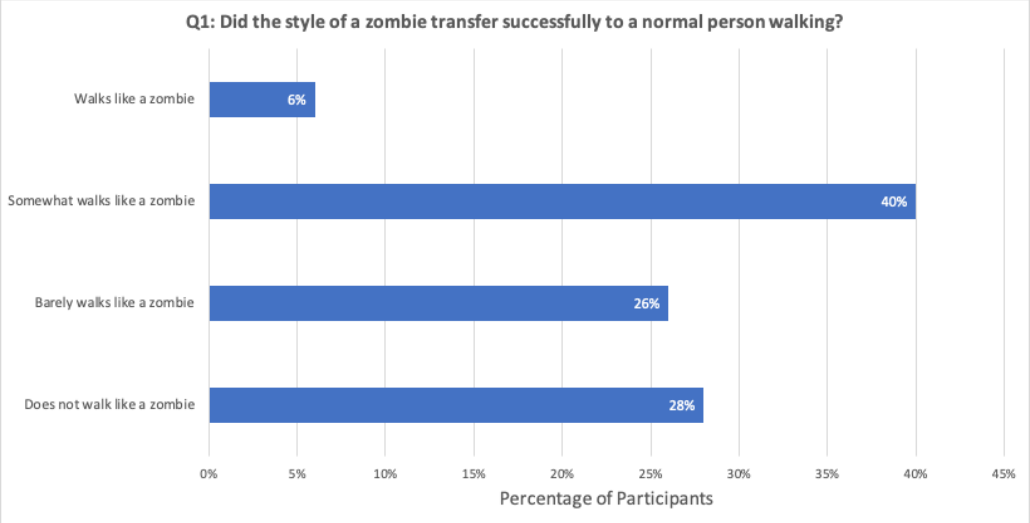


Figure 5.1: Participant's responses for question 1 of the survey.

A majority of participants decided that the synthesised locomotion had a somewhat resemblance to a zombie walking. This accounted for 40% of the votes placed. In terms of ranking order, "somewhat walks like a zombie" is considered three out of a possible four stars. The highest possible score namely, "walks like a zombie", had a response of 6%. On the bottom end of the spectrum, 26% and 28% of the participants felt that the synthesised locomotion "barely walks like a zombie" and "does not walk like a zombie", respectively. This indicates that audiences have a middle tier reaction to the synthesised locomotion however they are able to recognise some characteristics of a zombie walking.

The second question in this survey looked at the zombie running style. The responses from the participants can be found in Figure 5.2.

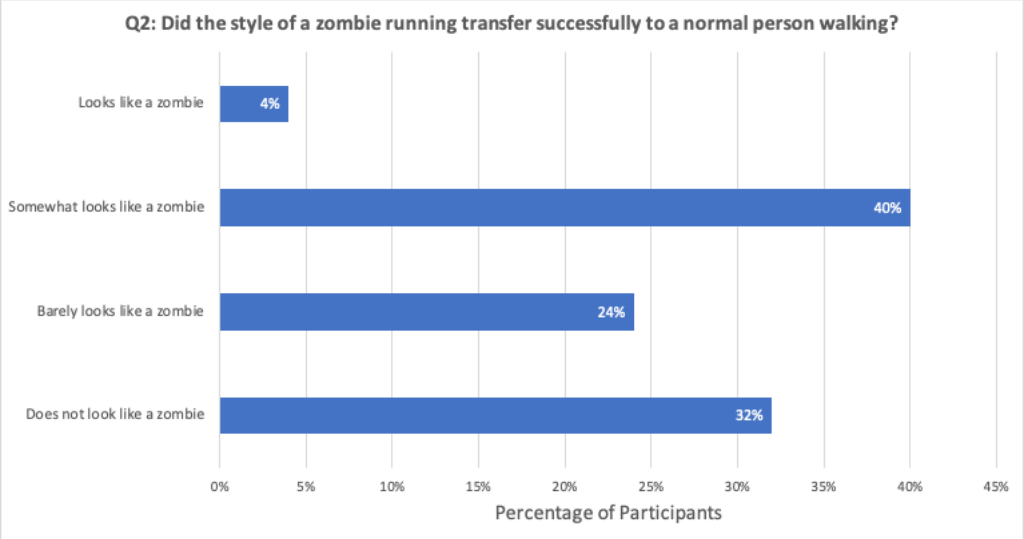


Figure 5.2: Participant's responses for question 2 of the survey.

The majority of participants viewed the synthesised locomotion as "somewhat looks like a zombie". Interestingly, the next highest category participants voted for was "does not look like zombie". The synthesised locomotion creates a polarising view for participants. A small percentage of participants did completely recognise a zombie. Comparatively question 1 and question 2 focused on the same stylised locomotion however the gaits differed. When comparing the responses from question 1 and 2, it can be seen that majority of participants could somewhat recognise a zombie. The top tier category did slightly drop in question 2 and more responses were placed in the bottom tier category. This provides an indication that the running gait for zombies is less successful during the transfer process. It is worth noting that the reference content locomotion was kept constant to a neutral walking gait across all the questions.

The third question in this survey asked participants to rate the success of style transfer for a gorilla walking. The responses from the participants can be found in Figure 5.3.

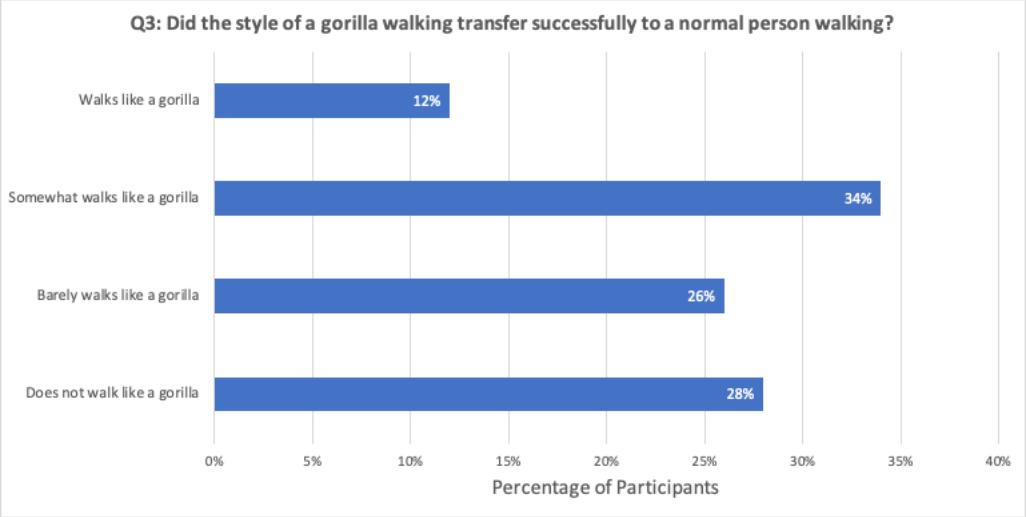


Figure 5.3: Participant's responses for question 3 of the survey.

It can be seen that the dominant category is "somewhat walks like a gorilla", however a close second category was given to "does not walk like a gorilla". A stronger response was given to the top tier category with 12% of the participants deciding that the synthesised locomotion "walks like a gorilla".

The fourth question in this survey focused on the stylised locomotion of a gorilla running. The responses from the participants can be found in Figure 5.4.

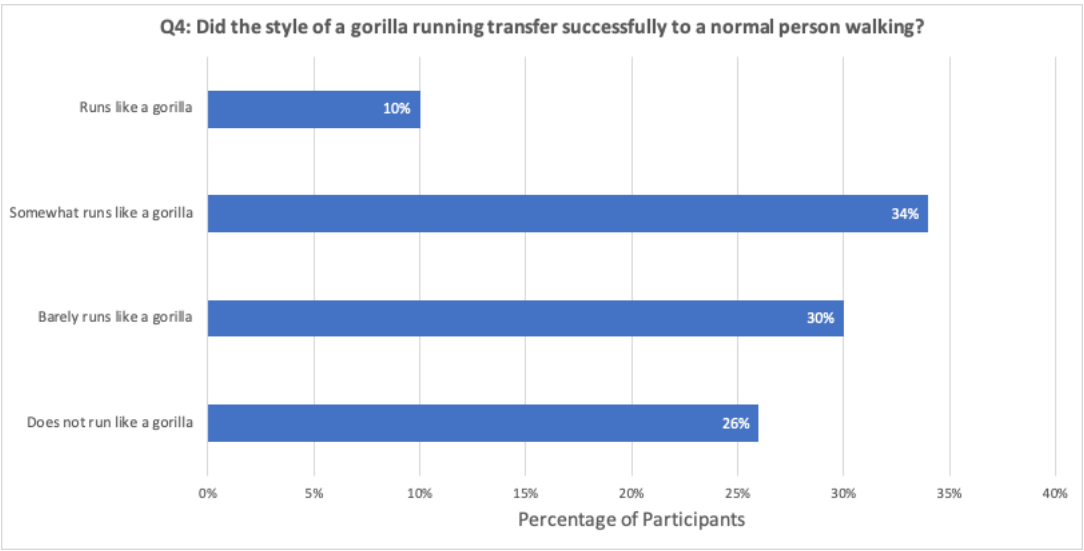


Figure 5.4: Participant's responses for question 4 of the survey.

The dominant category was given to "somewhat runs like a gorilla". The next major categories which participants voted was "barely walks like a gorilla" and "does not run like a gorilla", respectively. Comparatively, question 3 and question 4 focused on the same stylised locomotion of a gorilla however the gaits varied. Interestingly, it can be seen that majority of participants could recognise some gorilla

characteristic in the synthesised locomotion. There is a drop in the top tier category in question 4 as compared to question 3. A similar result was seen for zombies walking compared to running. This could be attributed to the varying gaits between the reference content and style locomotion.

The final question in this survey asked participants to rate the success of style transfer of a monkey running. The responses from the participants can be found in Figure 5.5.

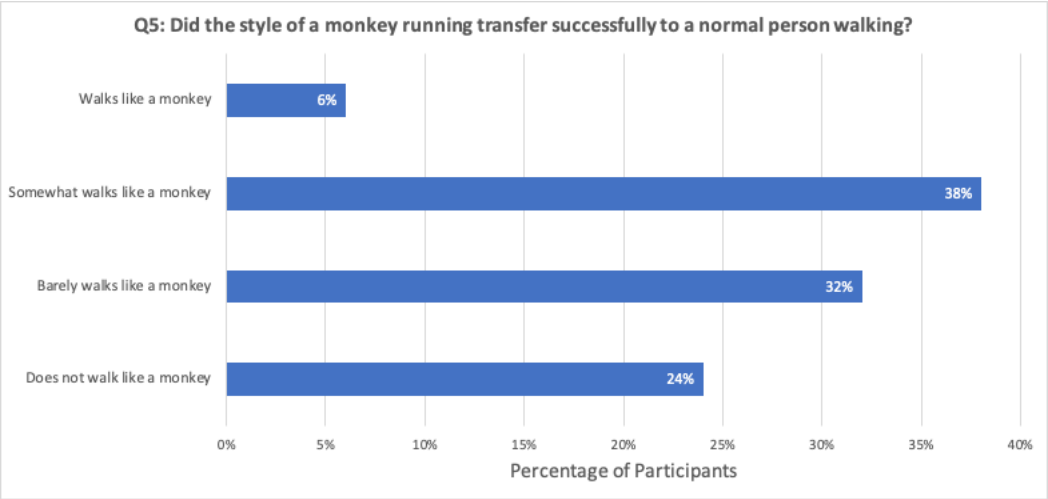


Figure 5.5: Participant's responses for question 5 of the survey.

A majority of participants voted for the "somewhat walks like a monkey" category. The next major categories were "barely walks like a monkey" and "does not walk like a monkey" with 32% and 25%, respectively. The smallest category voted for by participants was given to "walks like a zombie" with 6% of the vote.

The table below illustrates the average response for each question.

Question	Average	Standard Deviation
1	2.24	0.93
2	2.16	0.92
3	2.3	1.00
4	2.28	0.96
5	2.26	0.89
Average Total	2.25	0.94

Table 5.1: The average and standard deviation of the responses from participants generated from the survey.

The four point rating scale was multiplied by the ratio of responses per rating by the total number of

responses. The standard deviation was calculated by squaring the rating scale and multiplying it by the number of response in each category. This was subsequently subtracted from the squared average and square rooted.

5.1.2 Analysis

A qualitative approach has provided valuable insights into the perception of the viewer. This survey was tasked with answering the first research objective whether locomotion style is transferable. Each question in the survey centered around a different locomotion style with varying gaits. The reference content locomotion was kept constant throughout the questions so to avoid additional complexity and changing interpretations. This allows for a focused to be placed on the locomotion style being transferred. Essentially, demanding the participants to rate the success of the algorithm to transfer style.

Across all five questions in the survey, the participants could recognise some semblance of the locomotion style as the majority category selected most often was "somewhat walks/runs like a zombie/gorilla/monkey". Participants struggled to fully commit the synthesised locomotion to the top tier category. That is, the synthesised definitely walked/ran like the stylised locomotion in the question. Interestingly, the highest response in that category was given in question 3 where the style in question was gorilla walks. The reference content locomotion of a human neutral walking gait is in stark contrast to a gorilla walking. The gorilla walking style can be characterised as upright walking and not the well-known knuckle walking. This choice was intentional as this research is focused on bipedal locomotion in human and the reference style locomotion's performed by humans. The gorilla walking style is highly energetic in nature and very distinct characteristics are present. By comparing the weighted average across all five questions, the total average is 2.248 which indicates that viewers have polarising interpretations of the synthesised locomotion. Half of the audiences can recognise some stylised aspects however the other half can not. Individually, each question received a weighted average of above 2. This indicates that the average response from participants leans slightly towards the categories where style is present.

This research purposefully selected a four point rating scale. A clear division can be drawn. That is, the bottom two points indicate that style transfer was not achieved in the synthesised locomotion whereas the top two points indicate that style transfer was achieved. Anything with a rating of two and above indicates that the participants can establish some characteristics of style within the synthesised locomotion. This is extremely important to highlight as the reference content locomotion was a neutral walking gait and it was used as a base to generate the synthesised locomotion. Any semblance of stylised

locomotion in the synthesised locomotion is thus transferred from the reference style locomotion.

5.1.3 Conclusion

A qualitative survey was conducted on this research using the SurveyMonkey platform. A group of 50 participants across the United States of America was asked to rate the success of the style transfer process on a synthesised locomotion. Five questions were provided to the participants where each question centered around a different stylised locomotion. These stylised locomotions were namely, walking zombie, running zombie, walking gorilla, running gorilla and running monkey. This research included varying gaits in the stylised locomotion.

A four point rating scale was presented to each participant to rate the success of the style transfer process. This ranged from the synthesised locomotion not resembling the reference style to definitely resembling the reference style. The highest response received from participants was that the synthesised locomotion somewhat resembled the reference style. This indicates that some stylised aspects of locomotion was transferred to the synthesised locomotion. By analysing the average response score across all five questions, it can be seen that the participants had a polarising interpretations. However, there is a slight lean toward categories where style is present. The standard deviation is relatively low indicating that most of the responses are centered around the mean.

The first research objective concerns itself with the possibility of locomotion style transfer in the same way images have achieved style transfer. Therefore the recognition of characteristics of style within the synthesised locomotion is sufficient to conclude the deep learning framework's ability for locomotion style transfer. Given that the reference content locomotion is used as a base and that some participants can recognise some stylised aspects in the generated synthesised locomotion, this indicates that the transfer of style is possible for locomotion. The degree of believability is not included in this research and therefore a perfect score from the survey is not required. Additional enhancements to the framework can be made in subsequent studies.

5.2 Experiment 2: Trade-off Between the Amount of Style and Content

Fundamentally, style and content are completely intertwined. Gatys *et al.* [2015] state that there does not exist a newly synthesised image which perfectly matches the constraint of the content from one image and the style of another. Style transfer can be achieved by minimising the cost function in Equation 4.6. This cost function contains two important weighting parameters, namely α and β which are derived from

the style and content amount and are presented below in Equation 5.1 and Equation 5.2.

$$\alpha = \frac{s}{(c+s)} \tag{5.1}$$

$$\beta = \frac{c}{(c+s)} \tag{5.2}$$

where c represents the numeric amount of content and s represents the numeric amount of style.

By expressing these weighting parameters as a ratio of each other, it can be seen that there is an inverse relationship between style and content.

$$\begin{aligned} \frac{\alpha}{\beta} &= \frac{\frac{s}{(c+s)}}{\frac{c}{(c+s)}} \\ &= \frac{s}{(c+s)} . \frac{(c+s)}{c} \\ &= \frac{s}{c} \end{aligned} \tag{5.3}$$

Therefore, it can be seen that the more amount of content present on this constraint (Equation 3.5) will diminish the amount of style within the system, and vice versa. The weighting parameter α directly equates to the amount of style whereas β equates to the amount of content in the Equation 5.3. The synthesised artwork, or locomotion for the purposes of this research, will respond accordingly to the changing of the amount of content and style. These parameters can be seen as an interplay between each other and the effect of varying these quantities should be investigated in terms of its contribution to the overall synthesis.

5.2.1 Experimental Set-up

This investigation is focused on understanding the acceptable numeric range of the ratio of $\frac{\alpha}{\beta}$ for locomotion synthesis where α and β are weighing parameters for the style transfer loss function, Equation 3.5. Further, this will inform the acceptable range for the amount of style and content needed to achieved style transfer for locomotion.

The original content locomotion describes an actor walking with a naturalistic gait whereas the style locomotion describes an actor walking in a zombie style. Both the style and content locomotions very kept constant while the amount of style on the transfer constraint varied. This allows for a investigation into the effect of the amount of style on the transfer constraint.

In order to conduct this experiment, the content amount was kept constant and the style amount was adjusted with varying degree of base 10. The corresponding Mean Squared Error (MSE) and Peak Signal-to-Noise Ratio (PSNR) evaluation metrics were generated by comparing the newly synthesised locomotion with the original content locomotion.

This experiment used the ADAM algorithm as the method for optimising Equation 4.6. The model was trained using 250 epochs.

5.2.2 Results

In order to quantify the effects of the ratio of the weightings on locomotion style transfer, the two metrics (Section 4.5.2) are used to describe the locomotion and deviation from the original. The results of the experiments are recorded in Table 5.2. It can be seen that content amount is controlled during the experiments and the style amount differs by powers of 10.

Content Amount	Style Amount	α	β	$\frac{\alpha}{\beta}$	PSNR	MSE
1	10000	9,99900E-01	9,99900E-05	10000	9,00650	104,38407
1	1000	9,99001E-01	9,99001E-04	1000	9,00648	104,38326
1	100	9,90099E-01	9,90099E-03	100	9,00507	104,41464
1	10	9,09091E-01	9,09091E-02	10	9,00126	104,40784
1	1	5,00000E-01	5,00000E-01	1	8,99750	104,23783
1	0,1	9,09091E-02	9,09091E-01	0,1	9,12267	100,21577
1	0,01	9,90099E-03	9,90099E-01	0,01	9,21586	96,13782
1	0,001	9,99001E-04	9,99001E-01	0,001	10,27370	72,35302
1	0,0001	9,99900E-05	9,99900E-01	0,0001	15,80841	25,68663
1	0,00001	9,99990E-06	9,99990E-01	0,00001	20,07842	11,99392

Table 5.2: The effect of varying the amount of style while keeping the amount of content the same. The corresponding PSNR and MSE were used to evaluate the deviation of the synthesised locomotion compared to the original content locomotion.

Mean Squared Error (MSE)

MSE measures the average squared difference between the actual observations and the predicted estimates of a mathematical model. MSE is standard metric employed by many skilled professionals in

order to assess the validity and quality of a model. This research will use MSE to gauge the deviation of the synthesised locomotion from the original content locomotion.

Since MSE is essentially a difference between two quantities that have been squared, it is strictly positive and ranges from zero to infinity. A MSE score of zero indicates that the two locomotions in question are identical and there are no deviations. However, a very high MSE score indicates that there exists large deviations between the reference locomotion and the generated locomotion. MSE is defined in Equation 4.10. It may perhaps be observed that a perfect MSE score should not be the ideal goal-post for style transfer as this indicates that very little style was transferred and the newly synthesised locomotion is identical to the original content locomotion. For style transfer, the MSE score should be used to indicate the level of style transferred to the newly synthesised locomotion.

Figure 5.6 illustrates the varying ratio of the weighting parameters in Equation 4.6 and its corresponding MSE score by comparing the original locomotion with the newly synthesised locomotion generated by the model.

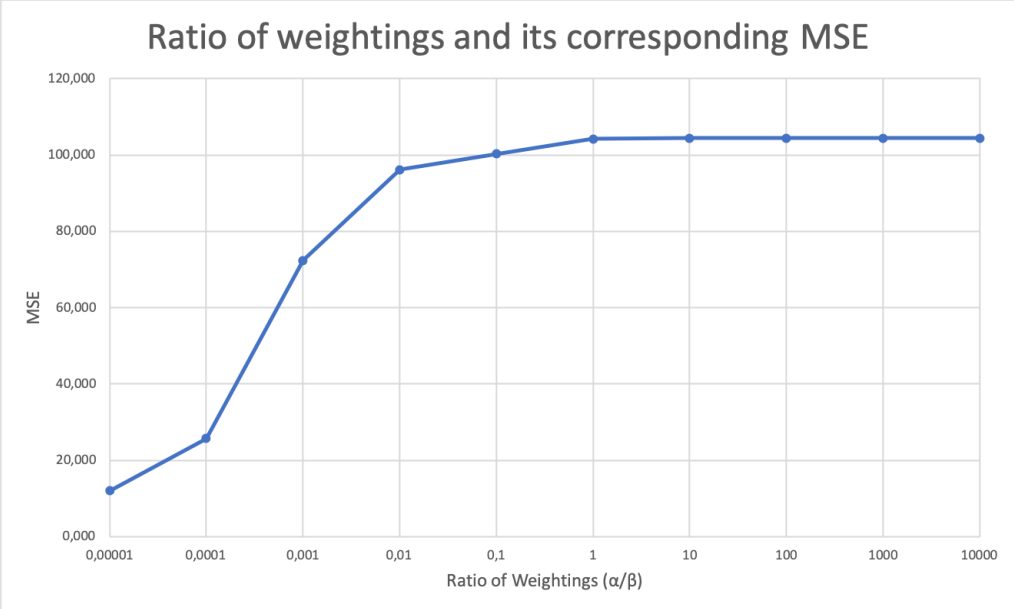


Figure 5.6: The Mean Squared Error (MSE) produced by comparing the original content locomotion to the newly synthesised locomotion using the corresponding ratio of weighting parameters in Equation 4.6. The exact value of the content and style amounts are described in Table 5.2.

It can be seen that the synthesised locomotion has a very low MSE when the ratio is 0.00001. Traditionally, a low MSE score indicates a model is able to accurately estimate the target. However, since this experiment is comparing the original reference content locomotion to the newly synthesised locomotion, a low MSE score would indicate that there is very little deviations. Thus, it can be concluded that the synthesised locomotion is very similar to the original content locomotion so very little style is present

within the synthesised locomotion. This is not desirable when trying to achieve style transfer.

When a ratio of 0.0001 is used, the graph can be described as having a logarithmic growth. This mean that beyond a certain ratio, there will be very little gain in style during the style transfer process. Beyond the ratio of 1.0, the growth has subsided and an asymptote could be present. This indicates that the model has learned the full spectrum of the style within the original style locomotion and transferred what it could to the synthesised locomotion. Ratios greater than, and including, 0.01 appear to produce maximal style within the newly synthesised locomotion and only marginal style gains will be achieved further. This is due to the fact that a high MSE is recorded which indicates that the synthesised locomotion deviates greatly from the content locomotion.

Peak Signal-to-Noise Ratio (PSNR)

PSNR is used to calculate the ratio between maximum power of a signal to the power of any corrupting noise. PSNR is expressed on the logarithmic decibel scale and defined in Equation 4.9. Higher quality reconstructions, have higher PSNR.

Figure 5.7 shows the relationship between the PSNR and the varying ratio of the weighting parameters in Equation 4.6. The PSNR was determined by comparing the original locomotion with the newly synthesised locomotion generated by the model.

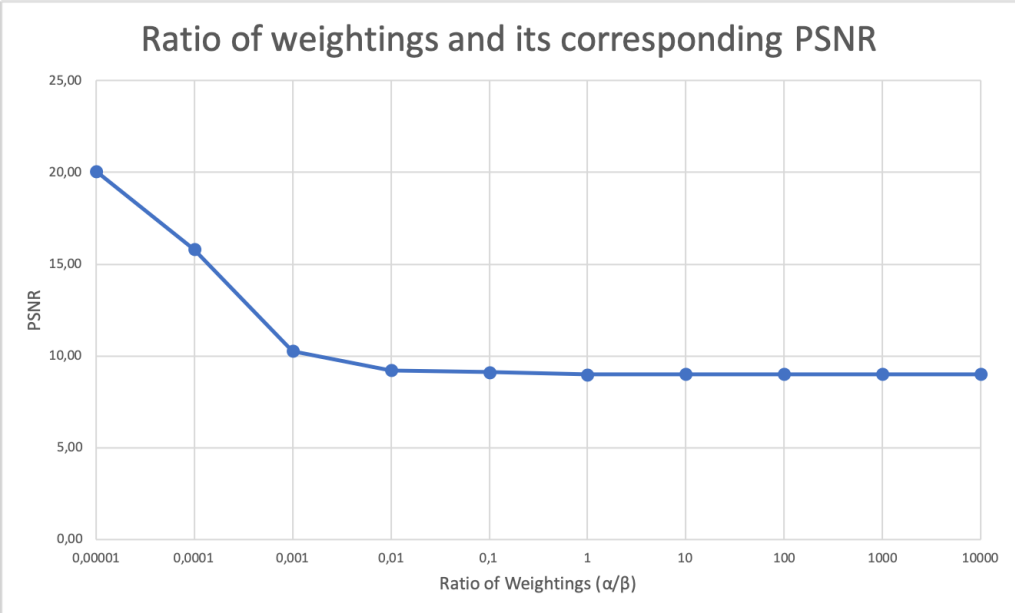


Figure 5.7: The Peak Signal-to-Noise Ratio (PSNR) produced by comparing the original content locomotion to the newly synthesised locomotion using the corresponding weighting ratio. The exact value of the content and style weighting is described in Table 5.2.

The lowest PSNR generated corresponds to a weighting ratio of 0.00001 and this will become the

baseline for comparison. This ratio indicates that the newly synthesised locomotion is identical in comparison to the content locomotion. That is, there is high similarity between the synthesised locomotion and the original content locomotion. It can be seen in Equation 4.9 that as the MSE approaches zero, PSNR will approach infinity. As mentioned above, a MSE score of zero indicates identical images and larger MSE scores indicates high deviations between locomotions. This is confirmed in Figure 5.6 as the weighting ratio of 0.00001 has an extremely low MSE score.

PSNR scores start to decrease between ratios 0.00001 and 0.01 however a stability is reached beyond 0.01. This indicates that deviations within the synthesised locomotion become more present when a ratio greater than 0.00001 is used. As well, there is very little gain in style during the style transfer process when a ratio greater than 0.01 is used.

Therefore, PSNR indicates the ideal ratio for locomotion style transfer is greater than 0.01 and consequently the ideal style amount is greater than 0.01

For an interactive demonstration of the above graphs and results, please proceed to the following link:

`http://www.bryceramgovind.ml/tables_stylevcontent.html`[2]

Analysis of Locomotion Rendering

A single frame from each synthesised locomotion was selected and visualised on a three-dimensional Cartesian plane with the corresponding content amount, style amount and ratio. This is provided below in Figure 5.8.

[2]Optimised for Chrome, Safari, Firefox, and Microsoft Edge. Older versions of Internet Explorer will suffer from compatibility issues. WebGL is used for the interactive animation renderings. This could be blocked by some organisations.

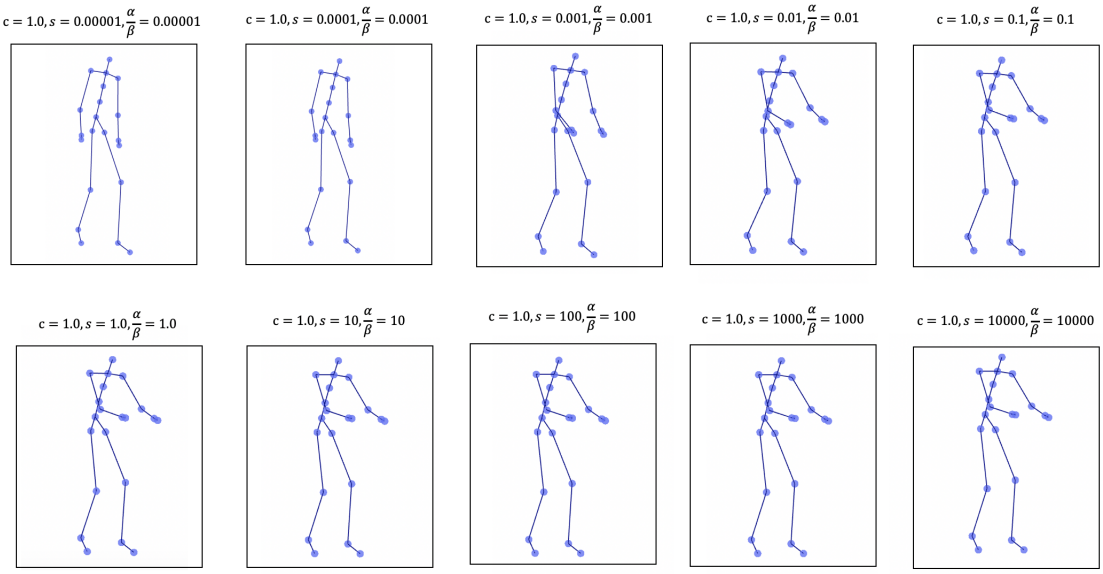


Figure 5.8: Reconstruction of a single frame from the newly synthesised locomotion using the corresponding ratio of the weighting parameters, α and β

A single frame from the original style and content locomotion which was used during the modelling process are presented in Figure 5.9. The frame-number was kept constant in all the renderings as well as the camera angle and scale. This allows for a uniform comparison of each rendering.

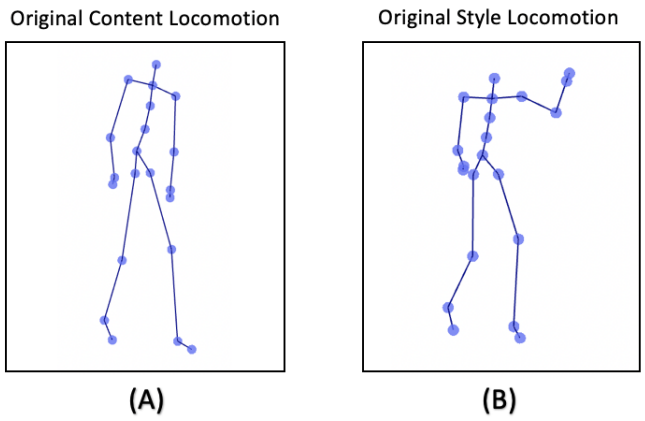


Figure 5.9: Reconstruction of a single frame from the original content and style locomotion.

Figure 5.9(A) presents the original content locomotion of the character where the character follows a circular trajectory in a relaxed manner. Within the entire locomotion sequence, their arm joints are always parallel to their body with very minimal swinging as they walk. This is a neutral locomotion with no stylisation by the actor. Figure 5.9.B presents the original stylised locomotion of the character where the actor is said to be imitating a zombie. Here, the character is shuffling along with out-stretched as if

they were trying to grab something in their path. The actor has chosen to slouch their shoulders while still maintaining a straighten angle between shoulder joints. The actor has a lower center of gravity when imitating a zombie, as there is a bend in the knee as they walk. These zombie characteristic will be used to assess the synthesised locomotion from a visual perspective in Figure 5.8.

The synthesised locomotion with a ratio $\frac{\alpha}{\beta}$ of 0.00001 provides the least desirable results. This locomotion resembles the original content locomotion which can be seen in Figure 5.9(A). The arms of the character are parallel to the body in the same the manner as the original content locomotion. This indicates that very little style was transferred during the modelling process.

The arm joints start to resemble the zombie style when the ratio $\frac{\alpha}{\beta}$ is greater than 0.01. These synthesised locomotion are the most appealing and provide the greatest resemblance to the original content locomotion. The angle of the shoulder joints to the horizontal starts to become flat as seen in original stylised locomotion. The locomotion appears to have outstretched arms very reminiscent of the zombie style. When the entire locomotion is rendered, a harmonious sequence is created for each joint; contributing to a naturalist locomotion. From a visual perspective, there is very little difference between the synthesised locomotion that have a ratio greater than 0.01. Thus, the amount of style needed to achieved style transfer for locomotion can be anything greater than 0.01.

It is important to note that the style transfer process did not exactly imitate the original style but rather learned the essence of the zombie style. This can be seen in the difference between the arm joints within the synthesised locomotion and the arm joints within the original stylised locomotion. The exact pose was not transferred, but the model learned to interpret the zombie's out-stretched arms from the stylised locomotion and transferred this keeping in line with the original content locomotion.

For an interactive visualisation comparison of the above renderings, please proceed to the following link:

`http://www.bryceramgovind.ml/stylevcontent_visualise.html`[3]

Performance

Table 5.3 illustrates the performance of the training process across varying weighting ratios by the time elapses and resource usage.

[3]Optimised for Chrome, Safari, Firefox, and Microsoft Edge. Older versions of Internet Explorer will suffer from compatibility issues. WebGL is used for the interactive animation renderings. This could be blocked by some organisations.

Content Amount	Style Amount	$\frac{\alpha}{\beta}$	Epochs	Training Time (s)	Memory Usage (bytes)
1	10000	10000	250	58.2466	68267604
1	1000	1000	250	60.9635	68267603
1	100	100	250	57.4722	68267602
1	10	10	250	58.4929	68267601
1	1	1	250	58.9687	68267600
1	0,1	0,1	250	60.7147	68267600
1	0,01	0,01	250	58.6635	68267601
1	0,001	0,001	250	58.3371	68267602
1	0,0001	0,0001	250	57.9396	68267601
1	0,00001	0,00001	250	57.986	68267602

Table 5.3: Performance breakdown on the training process

It can be seen that there is very little effect on the elapsed time by altering the style amount. The model is able to successfully transfer style within a time range of 60 seconds for 250 training epochs. The slight deviations of time could be attributed to scheduling by the operating system of the computer.

The memory usage across the test are almost identical. The average memory usage is 68267602 bytes which roughly converts to 65.11 megabytes. The change within the memory usage across the tests are not substantial enough to make a conclusion thus they should be treated as identical.

For an interactive demonstration of the performance table and its accompanying graphs, please proceed to the following link:

`http://www.bryceramgovind.ml/tables_stylevcontent_performance.html`[4]

5.2.3 Conclusion

There is an interconnected relationship between the content and style amounts when performing locomotion style transfer. These quantities should always be viewed as a trade-off between each other and expressing them as ratio allows for an analysis on the acceptable ranges which should be used during locomotion style transfer.

By keeping the content amount constant and varying the style amount, the model produced corresponding synthesised locomotions. These locomotions were compared to the original content locomotion

[4]Optimised for Chrome, Safari, Firefox, and Microsoft Edge. Older versions of Internet Explorer will suffer from compatibility issues. WebGL is used for the interactive animation renderings. This could be blocked by some organisations.

and a MSE and PSNR score was generated. The quality of the locomotion after the transfer process is indicated by these metrics. It is important to note that a perfect score within each metric is not desired for locomotion style transfer as this indicates that very little to no style was transferred. That is, the newly synthesised locomotion is identical to the original locomotion when a perfect score for each metric is calculated.

Weighting ratios greater than 0.01 produced highly desirable results for each evaluation metrics. However, there was very little gain in style during the style transfer process when ratio's greater than 0.01 were used. It can be seen that ratio weightings less than 0.01 produced locomotions which were very similar to the original content locomotion. The smaller the style weighting; the more identical the synthesised locomotion was produced.

Visual analysis is always important and hence, the accompaniment of the renderings to the generated results from each evaluation metrics. When a weightings ratio of 0.00001 was used, a locomotion was produced with identical resemblance to the human skeletal structure in the original content locomotion. The entire locomotion bared complete similarity to the original locomotion.

When a weightings ratio greater than 0.01 was used, locomotions appeared to have the unique 'zombie' style and mannerism. Locomotions appeared to resemble the original human skeletal structure and naturalistic locomotions was present. From a visual perspective, there was very little difference in style when a ratio weighting greater than and including 0.01 was used. Thus, a ratio greater than and including 0.01 should be adopted. On this basis, research objective 2 can be accepted.

5.3 Experiment 3: Style Transfer for Varying Gaits

The mathematical technique, known as optimisation, is fundamental to most machine learning algorithms. Gatys *et al.* [2015] illustrate that style can be transferred to a new image while maintaining the content by constraining the optimisation over a function. This research will use adaptive gradient descent as the optimisation algorithm constrained over the style transfer cost function in Equation 4.6.

5.3.1 Experimental Set-up

This investigation is focused on understanding the effect on convergence for stylised locomotions where the actor has adopted the zombie style but is using a running or walking gait. Convergence is the point at which there is marginal gain between successive evaluations of the cost function within the iterative optimisation process. Essentially, convergence dictates the extend of the optimisation process

as it informs about the number of iterations needed for successful style transfer. Running and walking are on opposite sides of the bipedal locomotion spectrum. Different muscles in the human body are needed to create a running motion, which further alter the joint positions. Lee and Farley [1998] state that the kinematics and kinetics of locomotion are vastly different between the walking gait and running gait for humans. When a human transitions from a walk to a run their foot contact with the ground decreases by 35 % [Nilsson *et al.* 1985]. In a paper entitled "Groucho running," the researchers explore the difference between the trajectory of the center of mass for walking and running gaits [McMahon *et al.* 1987]. The defining difference between the walking and running gait is attributed to the vertical position or movement of the center of gravity. A walker will possess a high center of gravity at the middle of the stance phase where as a runner will have a lower center of gravity during this phase [McMahon *et al.* 1987].

(a) Running zombie

(b) Walking zombie

Figure 5.10: Three-dimensional rendering of the zombie style where the character is running or walking through the scene.

Figure 5.10a and Figure 5.10b illustrate the difference between the running and walking locomotion and the effect on the joint positioning. The 'running' character is seen to have much larger strides in their feet positions where as the 'walking' character seems to be shuffling around scene as a zombie. Figure 5.10a shows a steeper bend in the knee and a further displacement of the knee joint in comparison. This translates to the character having a lower center of gravity and reaffirms the difference in the walking gait and running gait explored by McMahon *et al.* [1987]. The characteristic out-stretched arms of the zombie style is seen in both characters. The distance of the character between successive frames is calculated and presented in Table 5.4. This table provides a summary of the entire distance travelled for four different characters for the left and right foot joints. Two characters have opted for a

running gait and the other two have opted for the walking gait however, all are imitating the zombie style.

Joints	Distance (Walking 1)	Distance (Walking 2)	Distance (Running 3)	Distance (Running 4)
Right Heel	38.27	43.24	99.58	87.14
Right Toe	42.49	47.50	111.92	97.29
Left Heel	46.00	53.33	117.71	99.51
Left Toe	51.64	59.62	129.34	108.36

Table 5.4: The total distance of the left and right foot joints cover by each zombie character. The full list of joints are presented in Table B.1.

It can be seen in Table 5.4, that the 'running' character has covered almost double the distance the 'walking" character as the left toe for character 1 has a total distance of 51.64 units where the left toe for characters 3 and 4 have travelled a distance of 129.34 units and 108.36 units, respectively. The same trend can be seen for other joints.

To conduct this investigation, the reference content locomotion was kept constant and describes a character walking in a naturalistic, relaxed, up-right gait. The style of zombie was kept constant however the gaits differed. Two locomotions of the zombie walking and two locomotions of the zombie running were selected as the reference style locomotions. The deep learning framework generated four newly synthesised locomotions from the reference content locomotion and a reference style locomotion. As per the previous investigation, a style amount of 0.01 and content amount of 1.0 was used during the modelling process. The standard 250 iterations (epochs) was used across all the tests. This investigation is solely tasked with providing an understanding on the computational requirement and complexities of varying the gait of the reference style locomotion. Here, the loss generated from the deep learning framework during the optimisation will provide an understanding of how the gait of a style locomotion affects the convergence of the framework.

5.3.2 Results

The exact values generated from the evaluation of the cost function in Equation 4.6 can be seen in Appendix A. The graph below provides a summary of this table by illustrating the trend of the total loss generated.

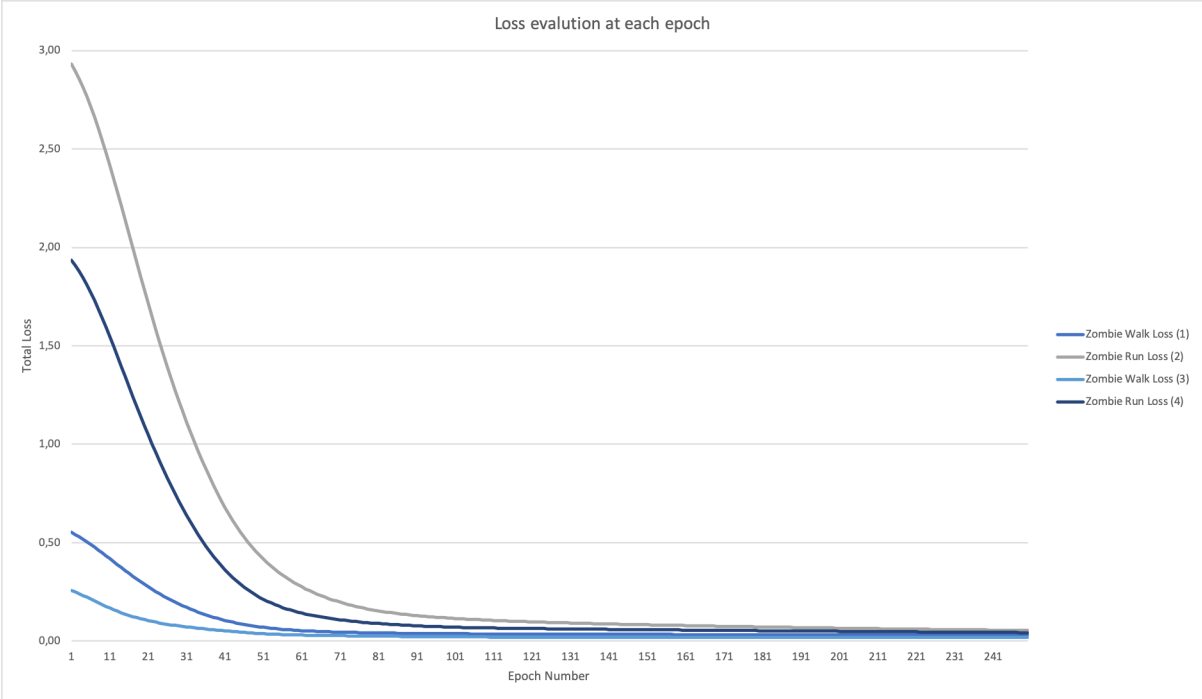


Figure 5.11: A graph showing the value generated at epoch by Equation 3.14 during the locomotion style transfer process for different locomotion style sequences.

For an interactive demonstration of the convergences graph and results, please proceed to the following link:

http://www.bryceramgovind.ml/convergence_results.html[5]

5.3.3 Analysis

A steep negative gradient is present for characters 2 and 4 in the initial stages of optimisation. That is, when the epoch number is relatively small. The 'running' character have a larger loss value on the first epoch as compared to the 'walking'. The loss value for the the running character ranges between 2.9 and 1.9 however the walking character's have a loss value between 0.5 and 0.2. This indicates that the 'running' motion manifold is vastly difference from the original content motion manifold and thus significant changes need to be made to achieve style transfer.

It can be seen that all the characters have a similar trend and that there is a point at which convergence is met. This convergence point indicates that style transfer has been achieved since the optimisation process was constrained using the cost function for style transfer in Equation 4.6.

Individual graphs for the loss value over the number of epochs are presented in Figure 5.12 and

[5]Optimised for Chrome, Safari, Firefox, and Microsoft Edge. Older versions of Internet Explorer will suffer from compatibility issues. WebGL is used for the interactive animation renderings. This could be blocked by some organisations.

Figure 5.13 for the ‘running’ and ‘walking’ characters, respectively.

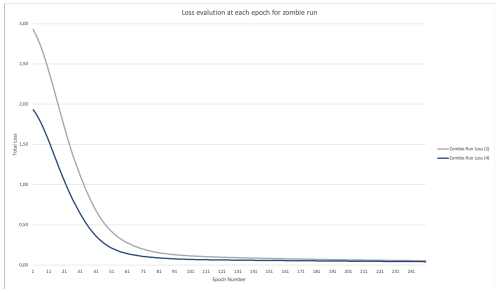


Figure 5.12: A graph showing the value generated at epoch by Equation 4.6 at epoch during the locomotion style transfer process for the ‘running’ characters.

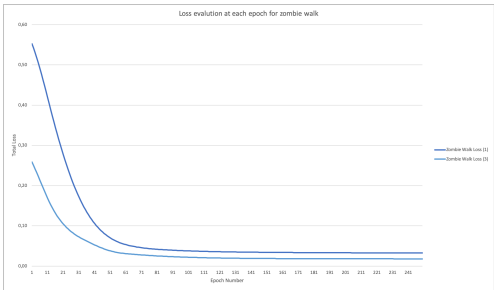


Figure 5.13: A graph showing the value generated at epoch by Equation 4.6 at epoch during the locomotion style transfer process for the ‘walking’ characters.

The convergence point for the ‘walking’ characters centered around 60 epochs as the graph stabilised in decay showing that there was marginal gain in the change of loss for further epochs. Style was transferred at this point. The ‘running‘ characters appeared to require more training time as their convergence point was delayed to around 90 epochs. Therefore, it can be seen that more training time is needed when a character is ‘running’ in a scene.

Although all four characters have adopted the zombie style, the additional training epochs required to achieve convergence as well as the larger loss values indicate that the running gait should be considered a separate sub-class of stylised motion.

Performance

Table 5.5 illustrates the performance of the training process for each character for the test above. The character gait and number is indicated in the first column of the table.

Character	Epochs	Training Time (s)	Memory Usage (bytes)
Zombie Walk (1)	250	59.1756	68267601
Zombie Walk (2)	250	58.0362	68267601
Zombie Run (3)	250	58.4033	68267600
Zombie Run (4)	250	58.7461	68267600

Table 5.5: Performance breakdown of the training process for each character.

It can be seen that there is very little effect on the elapsed time by altering the gait. The model is able to successfully transfer style within a time range of 60 seconds for 250 training epochs. The slight deviations of time could be attributed to scheduling by the operating system of the computer. The

memory usage across the test are almost identical. The average memory usage is 68267600.5 bytes which roughly converts to 65.11 megabytes. The change within the memory usage across the tests are not substantial enough to make a conclusion thus they should be treated as identical.

For an interactive demonstration of the above performance, please proceed to the following link: http://www.bryceramgovind.ml/gaits_performance.html[6]

5.3.4 Conclusion

Style transfer for zombie characters where the actor has opted for different gaits is achievable. That is, the model is able to learn the motion manifold of characters who are running or walking. The underlying motion capture data have fundamental changes when the actor is running or walking. In order to increase locomotion speed necessary for running, the character has to lower their center of gravity and lessen their foot to ground contact. These alterations affect the positions of the joints within a three dimensional space.

The loss values produced from the evaluation of the style transfer constraint in Equation 4.6 during the optimisations for each style transfer are presented above in Figure 5.11. This graph indicates that the model is able to learn the stylised motion manifold and transfer this to the newly synthesised locomotion for zombie characters who are either running or walking. The loss values for each style transfer describes a exponentially decaying shape over each iteration until stability is reached, at which the convergence is met. The convergence of the graph indicates that there is minimal gain in 'training' the model further as the style has been learned.

The difference between the running gait and the walking gait of a character adds a further complexity to the style transfer process for locomotion. The running style transfers had a larger loss value at the first iteration compared to the walking style transfers. The style motion manifold for the running characters took longer to learn as the convergence point was delayed by around 30 epochs.

Although style transfer is achievable, a new dimension should be considered when transferring locomotion style. Gaits such as running and walking describe similar characteristics the actor has to undertake regardless of the style. These characteristics have an effect of the motion manifold during style transfer. Motions that greatly deviate from the content locomotion require a longer training time. On this basis, research objective 3 can be accepted.

[6]Optimised for Chrome, Safari, Firefox, and Microsoft Edge. Older versions of Internet Explorer will suffer from compatibility issues. WebGL is used for the interactive animation renderings. This could be blocked by some organisations.

Chapter 6

Conclusion

"To infinity and beyond!"

- Buzz Lightyear, *Toy Story* (1995)

This chapter will provide a guide to further this area of study by outlining the recommendations and areas of future investigations. The final remarks are provided in Section 6.2 which draws the curtain on this research.

6.1 Recommendations and Future Work

Locomotion style transfer is achievable through the use of the convolutional autoencoder designed in Section 4.4.1. The model is able to construct style within the latent space representation and transfer this to the content locomotion creating a new synthesised locomotion.

The possibility of style transfer using a convolutional autoencoder was investigated by conducting a qualitative study. This would provide a view into an audience's interpretation of the synthesised locomotion generated by the deep learning framework. The survey focused on five stylised locomotions and asked participants to rate the success of the transfer process by looking at three videos per questions. All the videos were a static recording of the original 3D rendering of the selected joints from a single uniform angle. As this research is primarily focused on the character rigging process, it is recommended to provide future participants with a layer of animation skinning on each of the videos. It is difficult to interpret renderings when not given context. Therefore as an example, by applying a zombie appearance to the zombie locomotion, it might allow audiences to make the logical leaps of interpretation.

This research also looked at investigating the relationship between the amount of style and content required to achieve style transfer. The motion capture data was recorded and made available by Carnegie

Mellon Graphics Lab, University of Carnegie Mellon [Lab 2017]. There are limited stylised motion capture recordings available. Investigations into different stylised locomotion and its effect on the style and content amounts is recommended to further this research.
Stylised locomotion can include, but are not limited, to the following:

- Depressed
- Tired
- Old person
- Clown
- Injured

It can be seen that varying the gait of the character within the stylised locomotion effects the rate of style transfer and further training iterations are required. The running gait has a vastly different effect on the joint positioning in comparison to the walking gait. An investigation into the entire gait spectrum will provide a rigorous understanding of the effect the gait has on style transfer.
A list of possible gaits are listed below, this is not an exhaustive list but provides a good, common basis:

- Crawling
- Skipping
- Jogging
- Speed walking
- Crouch

By adapting this research for the gaming industry, developers will be able to alter characters to react with new styles without explicitly programming these changes. This will allow for faster game development cycle. The video game, Diablo III, took an estimated 11 years to complete. Game developers are able to create new narratives around characters quickly and thus gaming companies can be more reactive to changing trends within society. Holden *et al.* [2017] implement a novel neural network architecture which can handle real-time character control. Combining this or similar approaches with locomotion style transfer could further this field of research. The convolutional autoencoder should include an initial segment where high-level parameters are interpreted. This will allow for the processing of character controls to create naturalistic motions while imparting style.

This research provides a launch pad for further studies within many fields. With additional data resources, extensive research can be conducted which highlight optimal usage of the convolutional autoencoder. But further development of the deep learning framework can ultimately see practical applications within the game industry for character development. The style transfer algorithm by Gatys *et al.* [2015] is a mere seedling which has the potential to sprout a majestic forest.

6.2 Final Remarks

Natural human motion in character rigging is vital to maintain believability within an animated scene. Traditional character rigging is a long and laborious task within the animation process. This can only be completed by a group of skilled animators.

The motion capture framework has provided a way for the efficient recording of human motion. Motion capture data can be considered time series data since motion sequences are determined by the previous steps and a single movement is related to a time construct. That is, the data must be ordered in a time sequence for meaning to be generated. The raw motion capture data, recorded within the industry-standard *bvh* file, has to undergo rigorous transformations. During the recording process, many markers are placed on the actors to record their motion however only the most important 22 joints were selected for this research. These joints make up the most basic recognisable human skeleton. A reference joint was created which was based upon the central hip joint and is used to remove translation along the x and z axis. Additional derived fields were calculated namely, rotational velocity and binary response variables indicating whether the left or right foot joint was in contact with the floor for a specific frame in the sequence.

Artificial neural networks have shown dominance as techniques used in character motion synthesis. The amalgamation of the convolutional neural network and the 'encode-decode' technique used within autoencoders provide a deep learning framework in which locomotion style transfer is possible. The gram matrix provides a latent space representation for locomotion style and thus by transferring the motion manifold to the content locomotion a newly synthesised locomotion can be created which will possess the original character trajectory but in a stylised locomotion.

A survey conducted on this research revealed that participants were able to recognise stylised aspects within the synthesised locomotions generated from the deep learning framework. Participants across the United States of America were asked to rate the success of the transfer process on a four point rating scale. Each question focused on a different reference stylised locomotion while keeping the reference

content locomotion constant to a neutral walking gait. Across all five questions, majority of participants were able to somewhat identify the stylised aspects within the synthesised locomotion. This indicates that the deep learning framework was successful in transferring style to the neutral walking gait from an audience interpretation perspective.

There exists a relationship between the amount of style and content placed on the style transfer constraint in Equation 4.6. Thus, one has to view style and content as inter-related concepts. Two evaluation metrics were used to understand the deviation present within the newly synthesised locomotion in comparison to the original content locomotion. Perfect scores for PSNR and MSE are not desired in this case as they indicate an identical match to the original locomotion. Locomotions with extremely low style amounts performed poorly as little to no style was transferred. There was very little change in the synthesised locomotion across all metrics when the style amount was 0.01 or greater. Therefore, the style amount of 0.01 is considered the point at which maximum style transfer is achievable. The deep learning framework is remarkable in its ability to edit the motion manifold without exactly copying the stylised poses. The synthesised locomotion was seen to create its own poses which are not exact replicas of the original stylised locomotion. The deep learning framework captures the essence of locomotion style.

The kinematics of locomotion between the running gait and the walking gait presents a difference in motion manifolds. Although the four characters imitated the zombie style, two characters opted for the running gait where as the other two opted to walk. This had an impact on the training of the model. It was seen that the model had to undergo further iterations of training in order for the stylised motion manifold to be transferred of the running characters. Larger loss values during the initial training iterations for the running characters indicate that the locomotion vastly deviates from the original content locomotion. Therefore, sub-classes of stylised locomotion should exist that include the type of gait. The different types of gaits dictate the learning time needed for locomotion style transfer.

The combination of the traditional artistic techniques and computer science paved the way for arguably the best animated movies in film history, Toy Story. By adopting a new approach to character rigging using deep learning, animators are able to cut down on the animation process and potentially funnel this into more creative endeavours. The convolutional autoencoder allows for a creation of the latent stylised space representation. This can be easily transferred to any actor within a scene. The application of this research is far-reaching within the animation and live-action film space.

Bibliography

[Armbrust *et al.* 2010] Michael Armbrust, Ion Stoica, Matei Zaharia, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, and David et al. Patterson. A view of cloud computing. *Communications of the ACM*, 53(4):50, 2010.

[Avcibas and Sankur 2015] Ismail Avcibas and Bulent Sankur. Statistical analysis of image quality measures. *European Signal Processing Conference*, 2015, 03 2015.

[Bhat *et al.* 2008] Pravin Bhat, Brian Curless, Michael Cohen, and C. Lawrence Zitnick. Fourier analysis of the 2d screened poisson equation for gradient domain problems. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Computer Vision – ECCV 2008*, pages 114–128, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[Carneiro *et al.* 2018] Tiago Carneiro, Raul Victor Medeiros Da Nobrega, Thiago Nepomuceno, Gui-Bin Bian, Victor Hugo C. De Albuquerque, and Pedro Pedrosa Reboucas Filho. Performance analysis of google colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, 6:61677–61685, 2018.

[Cavagna and Kaneko 1977] G. A. Cavagna and M. Kaneko. Mechanical work and efficiency in level walking and running. *The Journal of Physiology*, 268(2):467–481, 1977.

[Champandard 2016] Alex J. Champandard. Semantic style transfer and turning two-bit doodles into fine artworks. *CoRR*, abs/1603.01768, 2016.

[Chen and Hsu 2016] Yi-Lei Chen and Chiou-Ting Hsu. Towards deep style transfer: A content-aware perspective. In *BMVC*, 2016.

[Chen *et al.* 2017] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stylebank: An explicit representation for neural image style transfer. *CoRR*, abs/1703.09210, 2017.

[Chen *et al.* 2018] Dongdong Chen, Lu Yuan, Jing Liao, Nenghai Yu, and Gang Hua. Stereoscopic neural style transfer. *CoRR*, abs/1802.10591, 2018.

[Chia *et al.* 2011] Alex Yong-Sang Chia, Shaojie Zhuo, Raj Kumar Gupta, Yu-Wing Tai, Siu-Yeung Cho, Ping Tan, and Stephen Lin. Semantic colorization with internet images. *ACM Trans. Graph.*, 30(6):156:1–156:8, December 2011.

[Churchland and Sejnowski 2016] Patricia S Churchland and Terrence J Sejnowski. *The Computational Brain*. The MIT Press, 2016.

[Cohen 2013] Rochelle S. Cohen. *Cell Biology of the Synapse*, pages 309–349. Springer New York, New York, NY, 2013.

[E. Rumelhart *et al.* 1986] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back propagating errors. *Nature*, 323:533–536, 10 1986.

[Fukushima 1980] K Fukushima. Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193—202, 1980.

[Gatys *et al.* 2015] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.

[Gatys *et al.* 2016] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, June 2016.

[Gatys *et al.* 2017] Leon A. Gatys, Alexander S. Ecker, Matthias Bethge, Aaron Hertzmann, and Eli Shechtman. Controlling perceptual factors in neural style transfer. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[Goodfellow *et al.* 2016] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

[Haykin 2009] Simon S Haykin. *Neural networks and learning machines*. Prentice Hall, 3 edition, 2009.

[He *et al.* 2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[Hinton 2012] Geoffrey E. Hinton. *A Practical Guide to Training Restricted Boltzmann Machines*, pages 599–619. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[Holden *et al.* 2015] Daniel Holden, Jun Saito, Taku Komura, and Thomas Joyce. Learning motion manifolds with convolutional autoencoders. In *SIGGRAPH Asia 2015 Technical Briefs*, SA '15, pages 18:1–18:4, New York, NY, USA, 2015. ACM.

[Holden *et al.* 2016] Daniel Holden, Jun Saito, and Taku Komura. A deep learning framework for character motion synthesis and editing. *ACM Trans. Graph.*, 35(4):138:1–138:11, July 2016.

[Holden *et al.* 2017] Daniel Holden, Taku Komura, and Jun Saito. Phase-functioned neural networks for character control. *ACM Trans. Graph.*, 36(4):42:1–42:13, July 2017.

[Hore and Ziou 2010] A. Hore and D. Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pages 2366–2369, Aug 2010.

[Kingma and Ba 2014] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*, 2014.

[Krathwohl 2009] David R Krathwohl. *Methods of educational and social science research*. LWaveland Press, 3 edition, 2009.

[Krizhevsky *et al.* 2012] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[Krizhevsky *et al.* 2017] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[Lab 2017] Carnegie Mellon Graphics Lab. `http://mocap.cs.cmu.edu/`, 2017.

[Lange *et al.* 2014] Mandy Lange, Dietlind Zühlke, Olaf Holz, and Thomas Villmann. Applications of lp-norms and their smooth approximations for gradient based learning vector quantization. In *ESANN*, 2014.

[LeCun and Cortes 2010] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[Lecun *et al.* 1998] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[Lee and Farley 1998] C R Lee and C T Farley. Determinants of the center of mass trajectory in human walking and running. *Journal of Experimental Biology*, 201(21):2935–2944, 1998.

[Lee *et al.* 2002] Jehee Lee, Jinxiang Chai, Paul S. A. Reitsma, Jessica K. Hodgins, and Nancy S. Pollard. Interactive control of avatars animated with human motion data. *ACM Transactions on Graphics*, 21(3), 2002.

[Lee *et al.* 2010] Victor W. Lee, Changkyu Kim, Jatin Chhugani, Michael Deisher, Daehyun Kim, Anthony D. Nguyen, Nadathur Satish, Mikhail Smelyanskiy, Srinivas Chennupaty, Per Hammarlund, Ronak Singhal, and Pradeep Dubey. Debunking the 100x gpu vs. cpu myth: An evaluation of throughput computing on cpu and gpu. *SIGARCH Comput. Archit. News*, 38(3):451–460, June 2010.

[Levin *et al.* 2006] Anat Levin, Dani Lischinski, and Yair Weiss. A closed form solution to natural image matting. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '06, page 61–68, USA, 2006. IEEE Computer Society.

[Li and Wand 2016] Chuan Li and Michael Wand. Combining markov random fields and convolutional neural networks for image synthesis. *CoRR*, abs/1601.04589, 2016.

[Li *et al.* 2017a] Shaohua Li, Xinxing Xu, Liqiang Nie, and Tat-Seng Chua. Laplacian-steered neural style transfer. *CoRR*, abs/1707.01253, 2017.

[Li *et al.* 2017b] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2230–2236, 2017.

[Li *et al.* 2017c] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *CoRR*, abs/1705.08086, 2017.

[Li *et al.* 2018] Yijun Li, Ming-Yu Liu, Xueting Li, Ming-Hsuan Yang, and Jan Kautz. A closed-form solution to photorealistic image stylization. *CoRR*, abs/1802.06474, 2018.

[Liu and Trenkler 2008] Shuangzhe Liu and Gotz Trenkler. *Hadamard, Khatri-Rao, Kronecker and other matrix products*, 2008.

[Luan *et al.* 2017] Fujun Luan, Sylvain Paris, Eli Shechtman, and Kavita Bala. Deep photo style transfer. *CoRR*, abs/1703.07511, 2017.

[Matthews 2019] Kayla Matthews. *6 Big Companies That Succeeded With Cloud Computing*, 2019.

[Mayor 2018] Adrienne Mayor. *Gods and Robots: Myths, Machines, and Ancient Dreams of Technology*. Princeton University Press, 2018.

[McMahon *et al.* 1987] T. A. McMahon, G. Valiant, and E. C. Frederick. Groucho running. *Journal of Applied Physiology*, 62(6):2326–2337, 1987.

[Mechrez *et al.* 2017] Roey Mechrez, Eli Shechtman, and Lihi Zelnik-Manor. Photorealistic style transfer with screened poisson equation. *CoRR*, abs/1709.09828, 2017.

[Nair and Hinton 2010] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress.

[Nilsson *et al.* 1985] Johnny Nilsson, Alf Thorstensson, and JñÑ Jalbertsma. Changes in leg movements and muscle activity with speed of locomotion and mode of progression in humans. *Acta Physiologica Scandinavica*, 123(4):457–475, 1985.

[Oliphant 2006] Travis Oliphant. *NumPy: A guide to NumPy*. USA: Trelgol Publishing, 2006–.

[Pedamonti 2018] Dabal Pedamonti. Comparison of non-linear activation functions for deep neural networks on MNIST classification task. *CoRR*, abs/1804.02763, 2018.

[Pixar 2016] Disney Pixar. *Modeling and Rigging — Toy Story — Disney Pixar*. https://www.youtube.com/watch?v=z_FYmhkOf4Q, 2016.

[Pluralsight 2014] Pluralsight. *The Rise of Motion Capture and What it Means for You*, May 2014.

[Rosenblatt 1958] Frank F. Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65 6:386–408, 1958.

[Satchell *et al.* 2016] Liam Satchell, Paul Morris, Chris Mills, Liam O'Reilly, Paul Marshman, and Lucy Akehurst. Evidence of big five and aggressive personalities in gait biomechanics. *Journal of Nonverbal Behavior*, 41(1):35–44, 2016.

[Shah 2012] Karan Shah. *Building A Basic Low Poly Character Rig In Blender*. https://cgi.tutsplus.com/tutorials/building-a-basic-low-poly-character-rig-in-blender--cg-16955?_ga=2.171069920.1313632652.1602184501-1002854970.1602184501, 2012.

[Silver *et al.* 2016] David Silver, Aja Huang, Christopher J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016.

[Simonyan and Zisserman 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[Slick 2017] Justin Slick. *How Are 3D Models Prepared for Animation?* https://www.lifewire.com/what-is-rigging-2095, 2017.

[Sporns *et al.* 2005] Olaf Sporns, Giulio Tononi, and Rolf Kötter. The human connectome: A structural description of the human brain. *PLoS Computational Biology*, 1(4):e42, 2005.

[Szegedy *et al.* 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[Vardanega 2013] Joshua Vardanega. *Pixar's Animation Process*. http://pixar-animation.weebly.com/pixars-animation-process.html, 2013.

[Wang and Bovik 2009] Z. Wang and A. C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, Jan 2009.

[Winter 1991] David A Winter. *The biomechanics and motor control of human gait*. University of Waterloo Press, 1991.

[Zhou Wang *et al.* 2004] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.

[Zorthian 2015] Julia Zorthian. *How 'Toy Story' Changed Movie History*. http://time.com/4118006/20-years-toy-story-pixar/, 2015.

Appendix A

Loss Values by Optimising the Style Transfer Constraint

Table A.1 displays the value generated by evaluating Equation 3.14 at each epoch. Each column within the table represents a different locomotion sequence. Each sequence has the actor imitate the zombie style of locomotion however, sequences (1) and (3) has the actor walking as a zombie where as sequence (2) and (4) the actor is running in the zombie style.

Epoch Number	**Zombie Walk Loss (1)**	**Zombie Run Loss (2)**	**Zombie Walk Loss (3)**	**Zombie Run Loss (4)**
1	0,5528351353	2,9306649618	0,2585457737	1,9343926540
2	0,5419413029	2,8961892370	0,2500568935	1,9078903040
3	0,5304173546	2,8575285770	0,2413431636	1,8783322720
4	0,5182462691	2,8147753314	0,2324133084	1,8457826219
5	0,5054367832	2,7680610537	0,2232965571	1,8103399075
6	0,4920222245	2,7175593390	0,2140411861	1,7721392540
7	0,4780587703	2,6634876678	0,2047123162	1,7313541153
8	0,4636231656	2,6061073202	0,1953889225	1,6881969960
9	0,4488096611	2,5457204475	0,1861598202	1,6429184940
10	0,4337258674	2,4826637330	0,1771183499	1,5958041014
11	0,4184872772	2,4172988867	0,1683556868	1,5471683925
12	0,4032104461	2,3500010742	0,1599532285	1,4973466071

Continued on next page

Table A.1 – *Continued from previous page*

Epoch Number	**Zombie Walk Loss (1)**	**Zombie Run Loss (2)**	**Zombie Walk Loss (3)**	**Zombie Run Loss (4)**
13	0,3880052932	2,2811468849	0,1519752928	1,4466841698
14	0,3729676756	2,2111040973	0,1444638924	1,3955250215
15	0,3581738900	2,1402259878	0,1374370350	1,3441996324
16	0,3436784853	2,0688510326	0,1308908792	1,2930137668
17	0,3295157410	1,9973052740	0,1248048281	1,2422393046
18	0,3157039798	1,9259039310	0,1191479472	1,1921076829
19	0,3022512337	1,8549510407	0,1138851148	1,1428056385
20	0,2891608173	1,7847367217	0,1089818121	1,0944734929
21	0,2764357591	1,7155311832	0,1044070931	1,0472070941
22	0,2640815364	1,6475758661	0,1001347807	1,0010638739
23	0,2521070204	1,5810744157	0,0961432436	0,9560720550
24	0,2405239035	1,5161866873	0,0924142472	0,9122413099
25	0,2293450870	1,4530272892	0,0889313892	0,8695731955
26	0,2185825945	1,3916681418	0,0856785755	0,8280700236
27	0,2082455676	1,3321435388	0,0826388677	0,7877412115
28	0,1983388121	1,2744562718	0,0797938862	0,7486065124
29	0,1888621850	1,2185840604	0,0771237961	0,7106958913
30	0,1798108738	1,1644864151	0,0746077742	0,6740463912
31	0,1711763810	1,1121125767	0,0722247852	0,6386971193
32	0,1629478756	1,0614107528	0,0699544740	0,6046840287
33	0,1551135486	1,0123376490	0,0677780148	0,5720360725
34	0,1476616906	0,9648662721	0,0656788024	0,5407735845
35	0,1405813470	0,9189900529	0,0636429273	0,5109087494
36	0,1338625361	0,8747225122	0,0616594236	0,4824472416
37	0,1274961094	0,8320931285	0,0597203083	0,4553899135
38	0,1214733853	0,7911409175	0,0578204562	0,4297337652
39	0,1157856952	0,7519072803	0,0559573544	0,4054719749

Continued on next page

Table A.1 – *Continued from previous page*

Epoch Number	**Zombie Walk Loss (1)**	**Zombie Run Loss (2)**	**Zombie Walk Loss (3)**	**Zombie Run Loss (4)**
40	0,1104239705	0,7144293085	0,0541307867	0,3825931723
41	0,1053784593	0,6787343516	0,0523424826	0,3610803231
42	0,1006386217	0,6448363033	0,0505957586	0,3409096032
43	0,0961931987	0,6127336750	0,0488951687	0,3220495198
44	0,0920304119	0,5824091385	0,0472461667	0,3044603928
45	0,0881382257	0,5538300299	0,0456547826	0,2880942402
46	0,0845045968	0,5269493755	0,0441273075	0,2728951551
47	0,0811176522	0,5017072436	0,0426699831	0,2588003238
48	0,0779657658	0,4780324494	0,0412886940	0,2457417810
49	0,0750375306	0,4558447094	0,0399886671	0,2336487989
50	0,0723216601	0,4350572243	0,0387741826	0,2224505694
51	0,0698068608	0,4155795125	0,0376483117	0,2120787132
52	0,0674817273	0,3973202483	0,0366126937	0,2024691883
53	0,0653346934	0,3801899096	0,0356673715	0,1935633485
54	0,0633540604	0,3641031358	0,0348107008	0,1853081227
55	0,0615280968	0,3489807386	0,0340393469	0,1776554959
56	0,0598451928	0,3347512640	0,0333483775	0,1705615874
57	0,0582940384	0,3213519057	0,0327314518	0,1639856377
58	0,0568637961	0,3087285209	0,0321810961	0,1578891477
59	0,0555442421	0,2968345982	0,0316890484	0,1522353118
60	0,0543258611	0,2856292967	0,0312466464	0,1469887845
61	0,0531998915	0,2750750090	0,0308452260	0,1421157346
62	0,0521583254	0,2651351051	0,0304764995	0,1375840917
63	0,0511938780	0,2557724736	0,0301328826	0,1333638791
64	0,0502999386	0,2469491863	0,0298077471	0,1294275359
65	0,0494705156	0,2386272219	0,0294955863	0,1257501603
66	0,0487001827	0,2307698546	0,0291920871	0,1223096144

Continued on next page

Table A.1 – *Continued from previous page*

Epoch Number	**Zombie Walk Loss (1)**	**Zombie Run Loss (2)**	**Zombie Walk Loss (3)**	**Zombie Run Loss (4)**
67	0,0479840271	0,2233431740	0,0288941171	0,1190864525
68	0,0473175996	0,2163172640	0,0285996380	0,1160636566
69	0,0466968633	0,2096667626	0,0283075654	0,1132262112
70	0,0461181406	0,2033707549	0,0280175938	0,1105605937
71	0,0455780586	0,1974121244	0,0277300083	0,1080542885
72	0,0450734966	0,1917765783	0,0274454977	0,1056954245
73	0,0446015394	0,1864515759	0,0271649839	0,1034725919
74	0,0441594416	0,1814253527	0,0268894755	0,1013748358
75	0,0437446042	0,1766861755	0,0266199507	0,0993917690
76	0,0433545672	0,1722218912	0,0263572704	0,0975137205
77	0,0429870136	0,1680197613	0,0261021218	0,0957318480
78	0,0426397834	0,1640665182	0,0258549869	0,0940381745
79	0,0423108917	0,1603485525	0,0256161355	0,0924255538
80	0,0419985458	0,1568521457	0,0253856341	0,0908876000
81	0,0417011575	0,1535636942	0,0251633691	0,0894186258
82	0,0414173463	0,1504698981	0,0249490778	0,0880136147
83	0,0411459336	0,1475579119	0,0247423842	0,0866682334
84	0,0408859274	0,1448154557	0,0245428358	0,0853788621
85	0,0406364993	0,1422308819	0,0243499384	0,0841426129
86	0,0403969576	0,1397931950	0,0241631866	0,0829573090
87	0,0401667172	0,1374920306	0,0239820903	0,0818214132
88	0,0399452723	0,1353176148	0,0238061936	0,0807339069
89	0,0397321709	0,1332607284	0,0236350887	0,0796941375
90	0,0395269965	0,1313126943	0,0234684236	0,0787016555
91	0,0393293537	0,1294653877	0,0233059048	0,0777560618
92	0,0391388616	0,1277112531	0,0231472960	0,0768568839
93	0,0389551506	0,1260433067	0,0229924134	0,0760034887

Continued on next page

Table A.1 – *Continued from previous page*

Epoch Number	**Zombie Walk Loss (1)**	**Zombie Run Loss (2)**	**Zombie Walk Loss (3)**	**Zombie Run Loss (4)**
94	0,0387778642	0,1244551123	0,0228411195	0,0751950368
95	0,0386066621	0,1229407356	0,0226933150	0,0744304730
96	0,0384412242	0,1214946922	0,0225489318	0,0737085440
97	0,0382812539	0,1201119092	0,0224079246	0,0730278288
98	0,0381264800	0,1187877079	0,0222702653	0,0723867695
99	0,0379766569	0,1175178040	0,0221359362	0,0717836903
100	0,0378315626	0,1162983104	0,0220049260	0,0712168027
101	0,0376909964	0,1151257268	0,0218772262	0,0706841989
102	0,0375547746	0,1139969085	0,0217528280	0,0701838436
103	0,0374227278	0,1129090156	0,0216317204	0,0697135742
104	0,0372946975	0,1118594538	0,0215138890	0,0692711174
105	0,0371705340	0,1108458210	0,0213993150	0,0688541262
106	0,0370500951	0,1098658703	0,0212879743	0,0684602315
107	0,0369332457	0,1089174924	0,0211798378	0,0680871014
108	0,0368198576	0,1079987142	0,0210748711	0,0677324989
109	0,0367098101	0,1071077061	0,0209730346	0,0673943276
110	0,0366029898	0,1062427880	0,0208742841	0,0670706635
111	0,0364992913	0,1054024283	0,0207785711	0,0667597700
112	0,0363986164	0,1045852341	0,0206858437	0,0664601012
113	0,0363008742	0,1037899348	0,0205960469	0,0661702959
114	0,0362059803	0,1030153638	0,0205091236	0,0658891673
115	0,0361138562	0,1022604421	0,0204250149	0,0656156917
116	0,0360244280	0,1015241672	0,0203436610	0,0653489967
117	0,0359376262	0,1008056067	0,0202650016	0,0650883503
118	0,0358533846	0,1001038961	0,0201889757	0,0648331482
119	0,0357716393	0,0994182372	0,0201155227	0,0645829016
120	0,0356923282	0,0987478977	0,0200445817	0,0643372223

Continued on next page

Table A.1 – *Continued from previous page*

Epoch Number	**Zombie Walk Loss (1)**	**Zombie Run Loss (2)**	**Zombie Walk Loss (3)**	**Zombie Run Loss (4)**
121	0,0356153903	0,0980922091	0,0199760923	0,0640958076
122	0,0355407647	0,0974505630	0,0199099938	0,0638584241
123	0,0354683902	0,0968224062	0,0198462258	0,0636248931
124	0,0353982046	0,0962072327	0,0197847277	0,0633950766
125	0,0353301447	0,0956045754	0,0197254387	0,0631688655
126	0,0352641458	0,0950139961	0,0196682980	0,0629461699
127	0,0352001421	0,0944350765	0,0196132445	0,0627269110
128	0,0351380668	0,0938674103	0,0195602169	0,0625110138
129	0,0350778526	0,0933105991	0,0195091536	0,0622984024
130	0,0350194319	0,0927642500	0,0194599930	0,0620889949
131	0,0349627375	0,0922279767	0,0194126732	0,0618827004
132	0,0349077034	0,0917013999	0,0193671327	0,0616794176
133	0,0348542643	0,0911841491	0,0193233097	0,0614790344
134	0,0348023569	0,0906758621	0,0192811430	0,0612814289
135	0,0347519193	0,0901761846	0,0192405716	0,0610864716
136	0,0347028917	0,0896847678	0,0192015351	0,0608940275
137	0,0346552162	0,0892012677	0,0191639737	0,0607039587
138	0,0346088367	0,0887253429	0,0191278285	0,0605161258
139	0,0345636994	0,0882566548	0,0190930417	0,0603303902
140	0,0345197527	0,0877948676	0,0190595565	0,0601466151
141	0,0344769470	0,0873396491	0,0190273175	0,0599646668
142	0,0344352349	0,0868906720	0,0189962706	0,0597844154
143	0,0343945716	0,0864476153	0,0189663633	0,0596057362
144	0,0343549143	0,0860101662	0,0189375448	0,0594285104
145	0,0343162228	0,0855780216	0,0189097660	0,0592526251
146	0,0342784590	0,0851508901	0,0188829796	0,0590779743
147	0,0342415868	0,0847284927	0,0188571400	0,0589044582

Continued on next page

Table A.1 – *Continued from previous page*

Epoch Number	**Zombie Walk Loss (1)**	**Zombie Run Loss (2)**	**Zombie Walk Loss (3)**	**Zombie Run Loss (4)**
148	0,0342055725	0,0843105649	0,0188322037	0,0587319834
149	0,0341703842	0,0838968569	0,0188081289	0,0585604629
150	0,0341359918	0,0834871345	0,0187848758	0,0583898150
151	0,0341023669	0,0830811791	0,0187624061	0,0582199640
152	0,0340694827	0,0826787885	0,0187406837	0,0580508393
153	0,0340373141	0,0822797758	0,0187196740	0,0578823757
154	0,0340058371	0,0818839704	0,0186993442	0,0577145122
155	0,0339750290	0,0814912174	0,0186796631	0,0575471927
156	0,0339448684	0,0811013771	0,0186606010	0,0573803650
157	0,0339153349	0,0807143251	0,0186421297	0,0572139809
158	0,0338864092	0,0803299512	0,0186242227	0,0570479958
159	0,0338580726	0,0799481589	0,0186068547	0,0568823685
160	0,0338303074	0,0795688644	0,0185900015	0,0567170615
161	0,0338030968	0,0791919957	0,0185736405	0,0565520407
162	0,0337764244	0,0788174916	0,0185577502	0,0563872753
163	0,0337502745	0,0784453009	0,0185423101	0,0562227378
164	0,0337246323	0,0780753816	0,0185273009	0,0560584037
165	0,0336994831	0,0777076996	0,0185127043	0,0558942513
166	0,0336748130	0,0773422287	0,0184985030	0,0557302616
167	0,0336506087	0,0769789489	0,0184846805	0,0555664180
168	0,0336268572	0,0766178463	0,0184712213	0,0554027060
169	0,0336035461	0,0762589120	0,0184581106	0,0552391133
170	0,0335806633	0,0759021416	0,0184453344	0,0550756294
171	0,0335581974	0,0755475344	0,0184328794	0,0549122456
172	0,0335361370	0,0751950931	0,0184207330	0,0547489553
173	0,0335144716	0,0748448231	0,0184088833	0,0545857534
174	0,0334931907	0,0744967317	0,0183973188	0,0544226367

Continued on next page

Table A.1 – *Continued from previous page*

Epoch Number	**Zombie Walk Loss (1)**	**Zombie Run Loss (2)**	**Zombie Walk Loss (3)**	**Zombie Run Loss (4)**
175	0,0334722843	0,0741508280	0,0183860288	0,0542596036
176	0,0334517428	0,0738071226	0,0183750030	0,0540966543
177	0,0334315568	0,0734656267	0,0183642317	0,0539337902
178	0,0334117174	0,0731263523	0,0183537054	0,0537710145
179	0,0333922157	0,0727893118	0,0183434155	0,0536083315
180	0,0333730435	0,0724545176	0,0183333533	0,0534457470
181	0,0333541926	0,0721219822	0,0183235110	0,0532832679
182	0,0333356550	0,0717917177	0,0183138809	0,0531209024
183	0,0333174233	0,0714637360	0,0183044556	0,0529586600
184	0,0332994900	0,0711380485	0,0182952283	0,0527965510
185	0,0332818479	0,0708146659	0,0182861922	0,0526345870
186	0,0332644903	0,0704935985	0,0182773410	0,0524727806
187	0,0332474104	0,0701748558	0,0182686687	0,0523111453
188	0,0332306018	0,0698584467	0,0182601694	0,0521496956
189	0,0332140581	0,0695443794	0,0182518378	0,0519884470
190	0,0331977732	0,0692326613	0,0182436684	0,0518274155
191	0,0331817413	0,0689232994	0,0182356562	0,0516666182
192	0,0331659567	0,0686162995	0,0182277963	0,0515060728
193	0,0331504136	0,0683116673	0,0182200843	0,0513457978
194	0,0331351067	0,0680094072	0,0182125155	0,0511858124
195	0,0331200308	0,0677095234	0,0182050858	0,0510261363
196	0,0331051806	0,0674120192	0,0181977912	0,0508667899
197	0,0330905512	0,0671168975	0,0181906277	0,0507077942
198	0,0330761376	0,0668241602	0,0181835916	0,0505491705
199	0,0330619353	0,0665338089	0,0181766794	0,0503909409
200	0,0330479395	0,0662458445	0,0181698875	0,0502331276
201	0,0330341457	0,0659602672	0,0181632128	0,0500757536

Continued on next page

Table A.1 – *Continued from previous page*

Epoch Number	**Zombie Walk Loss (1)**	**Zombie Run Loss (2)**	**Zombie Walk Loss (3)**	**Zombie Run Loss (4)**
202	0,0330205495	0,0656770767	0,0181566522	0,0499188420
203	0,0330071468	0,0653962723	0,0181502024	0,0497624162
204	0,0329939332	0,0651178524	0,0181438608	0,0496065001
205	0,0329809048	0,0648418152	0,0181376245	0,0494511178
206	0,0329680576	0,0645681581	0,0181314908	0,0492962935
207	0,0329553876	0,0642968782	0,0181254571	0,0491420518
208	0,0329428912	0,0640279720	0,0181195210	0,0489884173
209	0,0329305645	0,0637614354	0,0181136801	0,0488354149
210	0,0329184041	0,0634972641	0,0181079322	0,0486830694
211	0,0329064064	0,0632354531	0,0181022750	0,0485314057
212	0,0328945680	0,0629759970	0,0180967063	0,0483804490
213	0,0328828855	0,0627188901	0,0180912242	0,0482302241
214	0,0328713557	0,0624641261	0,0180858267	0,0480807559
215	0,0328599753	0,0622116984	0,0180805119	0,0479320693
216	0,0328487412	0,0619615999	0,0180752779	0,0477841890
217	0,0328376504	0,0617138232	0,0180701230	0,0476371395
218	0,0328266999	0,0614683606	0,0180650454	0,0474909453
219	0,0328158867	0,0612252040	0,0180600435	0,0473456304
220	0,0328052082	0,0609843447	0,0180551157	0,0472012188
221	0,0327946614	0,0607457742	0,0180502604	0,0470577341
222	0,0327842437	0,0605094831	0,0180454761	0,0469151995
223	0,0327739523	0,0602754622	0,0180407614	0,0467736381
224	0,0327637848	0,0600437018	0,0180361149	0,0466330723
225	0,0327537386	0,0598141918	0,0180315351	0,0464935243
226	0,0327438113	0,0595869220	0,0180270208	0,0463550158
227	0,0327340003	0,0593618819	0,0180225706	0,0462175681
228	0,0327243034	0,0591390609	0,0180181834	0,0460812018

Continued on next page

Table A.1 – *Continued from previous page*

Epoch Number	**Zombie Walk Loss (1)**	**Zombie Run Loss (2)**	**Zombie Walk Loss (3)**	**Zombie Run Loss (4)**
229	0,0327147182	0,0589184479	0,0180138578	0,0459459371
230	0,0327052425	0,0587000317	0,0180095927	0,0458117938
231	0,0326958741	0,0584838010	0,0180053869	0,0456787909
232	0,0326866108	0,0582697441	0,0180012394	0,0455469469
233	0,0326774506	0,0580578492	0,0179971490	0,0454162798
234	0,0326683914	0,0578481044	0,0179931146	0,0452868068
235	0,0326594311	0,0576404974	0,0179891353	0,0451585446
236	0,0326505679	0,0574350158	0,0179852100	0,0450315090
237	0,0326417998	0,0572316471	0,0179813377	0,0449057154
238	0,0326331249	0,0570303787	0,0179775176	0,0447811785
239	0,0326245414	0,0568311976	0,0179737485	0,0446579120
240	0,0326160475	0,0566340908	0,0179700298	0,0445359293
241	0,0326076415	0,0564390451	0,0179663603	0,0444152427
242	0,0325993216	0,0562460472	0,0179627394	0,0442958640
243	0,0325910862	0,0560550836	0,0179591660	0,0441778042
244	0,0325829337	0,0558661408	0,0179556395	0,0440610736
245	0,0325748625	0,0556792048	0,0179521590	0,0439456817
246	0,0325668709	0,0554942620	0,0179487237	0,0438316373
247	0,0325589575	0,0553112981	0,0179453328	0,0437189485
248	0,0325511208	0,0551302992	0,0179419856	0,0436076224
249	0,0325433594	0,0549512509	0,0179386814	0,0434976656
250	0,0325356717	0,0547741388	0,0179354195	0,0433890839

Table A.1: The total loss generated at each epoch during the training process.

Appendix B

Euclidean Distance between Frames for the Zombie Style

The table below illustrates the total distance travelled by a character either walking or running in the zombie style. The distance is calculated using the euclidean distance for successive frames per joint. The path travelled by the character is calculated by summing the euclidean distance between frames for each specific joint. Table B.1 presents a character either walking or running with two different sequences provided.

Joints	Distance (Walking 1)	Distance (Walking 2)	Distance (Running 3)	Distance (Running 4)
Central Hips	5.28	5.58	21.05	12.49
Right Hip	5.32	6.19	19.33	13.14
Right Knee	20.23	23.17	53.93	43.62
Right Heel	38.27	43.24	99.58	87.14
Right Toe	42.49	47.50	111.92	97.29
Left Hip	7.34	8.47	24.00	17.71
Left Knee	23.12	30.96	67.99	58.52
Left Heel	46.00	53.33	117.71	99.51
Left Toe	51.64	59.62	129.34	108.36
Lower Spine	5.59	9.62	19.50	12.83

Continued on next page

Joints	Distance (Walking 1)	Distance (Walking 2)	Distance (Running 3)	Distance (Running 4)
Upper Spine	9.09	16.86	22.26	16.72
Neck	12.20	22.99	25.76	19.61
Head	16.38	28.65	31.57	22.86
Right Shoulder	18.12	24.53	27.40	20.86
Right Elbow	26.48	26.73	64.60	27.25
Right Palm base	39.52	41.77	106.01	38.15
Right Index Finger	43.15	45.75	115.69	41.11
Left Shoulder	14.78	25.43	30.27	23.27
Left Elbow	26.39	19.81	43.59	27.06
Left Palm base	37.34	29.61	63.47	35.34
Left hand index finger	40.09	32.41	68.09	37.24