STEP: Spatial Temporal Graph Convolutional Networks for Emotion Perception from Gaits

Uttaran Bhattacharya,[1] **Trisha Mittal,**[1] **Rohan Chandra,**[1] **Tanmay Randhavane,**[2]
Aniket Bera,[1] **Dinesh Manocha**[1]

[1]Department of Computer Science, University of Maryland, College Park, USA
[2]Department of Computer Science, University of North Carolina, Chapel Hill, USA
{uttaranb, trisha, rohan, ab, dm}@cs.umd.edu, tanmay@cs.unc.edu
Project URL: http://www.gamma.umd.edu/step

Abstract

We present a novel classifier network called STEP, to classify perceived human emotion from gaits, based on a Spatial Temporal Graph Convolutional Network (ST-GCN) architecture. Given an RGB video of an individual walking, our formulation implicitly exploits the gait features to classify the perceived emotion of the human into one of four emotions: happy, sad, angry, or neutral. We train STEP on annotated real-world gait videos, augmented with annotated synthetic gaits generated using a novel generative network called STEP-Gen, built on an ST-GCN based Conditional Variational Autoencoder (CVAE). We incorporate a novel push-pull regularization loss in the CVAE formulation of STEP-Gen to generate realistic gaits and improve the classification accuracy of STEP. We also release a novel dataset (E-Gait), which consists of 4,227 human gaits annotated with perceived emotions along with thousands of synthetic gaits. In practice, STEP can learn the affective features and exhibits classification accuracy of 88% on E-Gait, which is 14–30% more accurate over prior methods.

1 Introduction

Human emotion recognition using intelligent systems is an important socio-behavioral task that arises in various applications, including behavior prediction (Denham et al. 2000), surveillance (Arunnehru and Geetha 2017), robotics (Bauer and others 2009), affective computing (Yates et al. 2017; Atcheson, Sethu, and Epps 2017), etc. Current research in perceiving human emotion predominantly uses facial cues (Fan et al. 2016), speech (Jacob and Mythili 2015), or physiological signals such as heartbeats and respiration rates (Zhao, Adib, and Katabi 2016). These techniques have been used to identify and classify broad emotions including happiness, sadness, anger, disgust, fear and other combinations (Ekman and Friesen 1967).

Understanding the perceived emotions of individuals using non-verbal cues, such as face expressions or body movement, is regarded as an important and challenging problem in both AI and psychology, especially when self-reported emotions are unreliable or misleading (Quigley, Lindquist,
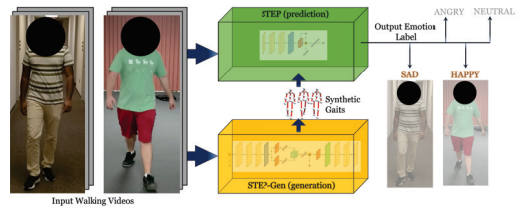
Figure 1: STEP and STEP-Gen: We present a novel classifier network (STEP) to predict perceived emotions from gaits extracted from walking videos. We also present a generator network (STEP-Gen) to generate annotated synthetic gaits to improve the accuracy of STEP.

and Barrett 2014). Most prior work has focused on facial expressions, due to the availability of large datasets (Fabian Benitez-Quiroz, Srinivasan, and Martinez 2016). However, facial emotions can be unreliable in contexts such as referential expressions (Ekman 1993) or the presence or absence of an audience (Fernández-Dols and Ruiz-Belda 1995). Thus, we need techniques that can utilize other non-verbal cues.

In this paper, we focus on using movement features corresponding to gaits in a walking video for emotion perception. A gait as an ordered temporal sequence of body joint transformations (predominantly translations and rotations) during the course of a single walk cycle. Simply stated, a person's gait is the way the person walks. Prior works in psychology literature have reported that participants were able to identify sadness, anger, happiness, and pride by observing affective features corresponding to arm swinging, long strides, erect posture, collapsed upper body, etc. (Michalak et al. 2009; Kleinsmith and Bianchi-Berthouze 2013).

There is considerable recent work on pose or gait extraction from a walking video using deep convolutional network architectures and intricately designed loss functions (Dabral et al. 2018; Girdhar et al. 2017). Gaits have also been used for a variety of applications including action recognition (Wang et al. 2003; Yan, Xiong, and Lin 2018) and person identification (Zhang and Troje 2005). However, the use of gaits for automatic emotion perceptions has been fairly limited, primarily due to a lack of gait data or videos anno-

tated with emotions (Chiu, Shu, and Hui 2018). It is difficult and challenging to generate a large dataset with many thousands of annotated real-world gait videos to train a network.

Main Contributions: We present a learning-based approach to classify perceived emotions of an individual walking in a video. Our formulation consists of a novel classifier and a generative network as well as an annotated gait video dataset. The main contributions include:

1. A novel end-to-end Spatial Temporal Graph Convolution-Based Network (STEP), which implicitly extracts a person's gait from a walking video to predict their emotion. STEP combines deeply learned features with affective features to learn hybrid features.

2. A Conditional Variational Autoencoder (CVAE) called STEP-Gen, which is trained on a sparse real-world annotated gait set and can easily generate thousands of annotated synthetic gaits. We enforce the temporal constraints (*e.g.,* gait drift and gait collapse) inherent in gaits directly into the loss function of the CVAE, along with a novel push-pull regularization loss term. Our formulation helps to avoid over-fitting by generating more realistic gaits. These synthetic gaits improve the accuracy of STEP by 6% in our benchmarks.

3. We present a new dataset of human gaits annotated with emotion labels, called Emotion-Gait (E-Gait). It currently consists of $4,227$ real-world gait videos annotated with the emotion labels happy, sad, angry, and neutral.

We have evaluated the performance of STEP on E-Gait. The gaits in this dataset were extracted from videos of humans walking in both indoor and outdoor settings. In practice, STEP results in classification accuracy of 88% on E-Gait. We have compared it with prior methods and observe:

• An accuracy increase of 14% over prior learning-based method (Randhavane et al. 2019). This method uses LSTMs for modeling their input, but for an action recognition task.

• Accuracy improvement of $21 - 30\%$ on the absolute over prior gait-based emotion recognition methods reported in the psychology literature that use affective features.

2 Related Work

We provide a brief overview of prior work in emotion perception and generative models for gait-like datasets.

Emotion Perception. Face and speech data have been widely used to perceive human emotions. Prior methods that use faces as input commonly track action units on the face such as points on the eyebrow, cheeks and lips (Fabian Benitez-Quiroz, Srinivasan, and Martinez 2016), or track eye movements (Schurgin et al. 2014) and facial expressions (Majumder, Behera, and Subramanian 2014). Speech-based emotion perception methods use either spectral features or prosodic features like loudness of voice, difference in tones and changes in pitch (Jacob and Mythili 2015). With the rising popularity of deep learning, there is considerable work on developing learned features for emotion detection from large-scale databases of faces (Yang, Ciftci, and Yin 2018; Zhang et al. 2018) and speech signals (Deng

et al. 2018). Recent methods have also looked at the cross-modality of combined face and speech data to perform emotion recognition (Albanie et al. 2018). In addition to faces and speech, physiological signals such as heartbeats and respiration rates (Zhao, Adib, and Katabi 2016) have also been used to increase the accuracy of emotion perception. Our approach for emotion perception from walking videos and gaits is complimentary to these methods and can be combined.

Different methods have also been proposed to perceive emotions from gaits. Karg et al. (2010) use PCA-based classifiers, and Crenn et al. (2016a) use SVMs on affective features. Venture et al. (2014) use autocorrelation matrices between joint angles to perform similarity-based classification. Daoudi et al. (2017) represent joint movements as symmetric positive definite matrices and perform nearest neighbor classification.

Gaits have also been widely used in the related problem of action recognition (Ji et al. 2013; Feichtenhofer, Pinz, and Wildes 2016; Feichtenhofer, Pinz, and Zisserman 2016; Lea et al. 2017; Yan, Xiong, and Lin 2018). In our approach, we take motivation from prior works on both, emotion perception and action recognition from gaits.

Gait Generation. Collecting and compiling a large dataset of annotated gait videos is indeed a challenging task. As a result, it is important to develop generative algorithms for gaits conditioned on emotion labels. Current learning-based generation models are primarily based on Generative Adverserial Networks (GANs) or Variational Autoencoders (VAEs). MoCoGAN (Tulyakov et al. 2018) uses a GAN-based model, the latent space of which is divided into motion space (for generating temporal features) and content space (for generating spatial features). It can generate tiny videos of facial expressions corresponding to various emotions. vid2vid (Wang et al. 2018) is a state-of-the-art GAN-based network that uses a combined spatial temporal adversarial objective to generate high-resolution videos, including videos of human poses and gaits when trained on relevant real data. Other generative methods for gaits learn the initial poses and the intermediate transformations between frames in separate networks, and then combine the generated samples from both networks to develop realistic gaits (Yang et al. 2018; Cai et al. 2018). In this work, we model gaits as skeletal graphs and use spatial-temporal graph convolutions (Yan, Xiong, and Lin 2018) inside a VAE to generate synthetic gaits.

3 Background

In this section, we give a brief overview of Spatial Temporal Graph Convolutional Networks (ST-GCNs) and Conditional Variational Autoencoders (CVAE).

3.1 GCN and ST-GCN

The Graph Convolutional Network (GCN) was first introduced in (Bruna et al. 2013) to apply convolutional filters to arbitrarily structured graph data. Consider a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with $N = |\mathcal{V}|$ nodes. Also consider a feature matrix $X \in \mathbb{R}^{N \times F}$, where row $x_i \in \mathbb{R}^F$ corresponds to a

feature for vertex i. The propagation rule of a GCN is given as

$$Z^{(l+1)} = \sigma(A Z^{(l)} W^{(l)}), \qquad (1)$$

where $Z^{(l)}$ and $Z^{(l+1)}$ are the inputs to the l-th and the $(l+1)$-th layers of the network, respectively. $Z^{(0)} = X$, $W^{(l)}$ is the weight matrix between the l-th and the $(l+1)$-th layers, A is the $N \times N$ adjacency matrix associated with the graph $\mathcal{G}$ and $\sigma(\cdot)$ is a non-linear activation function (*e.g.*, ReLU). Thus, a GCN takes in a feature matrix X as an input and generates another feature matrix $Z^{(L)}$ as the output, L being the number of layers in the network. In practice, each weight matrix W in a GCN represents a convolutional kernel. Multiple such kernels can be applied to the input of a particular layer to get a feature tensor as output, similar to a conventional Convolutional Neural Network (CNN). For example, if K kernels, each of dimension $F \times D$ are applied to the input X, then the output of the first layer will be an $N \times D \times K$ feature tensor.

Yan et al. (2018) extended GCNs to develop the spatial temporal GCN (ST-GCN), which can be used for action recognition from human skeletal graphs. The graph in their case is the skeletal model of a human extracted from videos. Since they extract poses from each frame of a video, their input is a temporal sequence of such skeletal models. "Spatial" refers to the spatial edges in the skeletal model, which are the limbs connecting the body joints. "Temporal" refers to temporal edges which connect the positions of each joint across different time steps. Such a representation enables the gait video to be expressed as a single graph with a fixed adjacency matrix, and thus can be passed through a GCN network. The feature per vertex in their case is the 3D position of the joint represented by that vertex. In our work, we use the same representation for gaits, described later in Section 4.1.

3.2 Conditional Variational Autoencoder

The variational autoencoder (Kingma and Welling 2013) is an encoder-decoder architecture that is used for data generation based on Bayesian inference. The encoder transforms the training data into a latent lower-dimensional distribution space. The decoder draws random samples from that distribution and generates synthetic data that are as similar to the training data as possible.

In conditional VAE (Sohn, Lee, and Yan 2015), instead of generating from a single distribution space learned by the encoder, it learns separate distributions for the separate classes in the training data. Thus, given a class, the decoder produces random samples from the conditional distribution of that class, and generates synthetic data of that class from those samples. Furthermore, if we assume that the decoder generates Gaussian variables for every class, then the negative log likelihood for each class is given by the MSE loss

$$\mathcal{L}_o = \|x - f_{\theta_c}(z)\|^2 \qquad (2)$$

where $f_{\theta_c}(\cdot)$ denotes the decoder function for class c, x represents the training data, and z the latent random variable. We incorporate a novel push-pull regularization loss on top of this standard CVAE loss, as described in Section 5.3.

4 STEP and STEP-Gen

Our objective is to perform emotion perception from gaits. Based on prior work (Kleinsmith and Bianchi-Berthouze 2013; Karg, Kuhnlenz, and Buss 2010; Crenn et al. 2016a), we assume that emotional cues are largely determined by localized variances in gaits, such as swinging speed of the arm (movement of 3 adjacent joints: shoulder, elbow and hand), stride length and speed (movement of 3 adjacent joints: hip, knee and foot), relative position of the spine joint w.r.t. the adjacent root and neck joints and so on. Convolutional kernels are known to capture such local variances and encode them into meaningful feature representations for learning-based algorithms (Krizhevsky, Sutskever, and Hinton 2012). Additionally, since we treat gaits as a periodic motion that consists of a sequence of localized joint movements in 3D, we therefore use GCNs for our generation and classification networks to capture these local variances efficiently. In particular, we use Spatial Temporal GCNs (ST-GCNs) developed by Yan, Xiong, and Lin to build both our generation and classification networks. We now elaborate our entire approach in detail.

4.1 Extracting Gaits from Videos

Naturally collected human gait videos contain a wide variety of extraneous information such as attire, items carried (*e.g.*, bags or cases), background clutter, etc. We use a state-of-the art pose estimation method (Dabral et al. 2018) to extract clean, 3D skeletal representations of the gaits from videos. Moreover, gaits in our dataset are collected from varying viewpoints and scales. To ensure that the generative network does not end up generating an extrinsic mean of the input gaits, we perform view normalization. Specifically, we transform all gaits to a common point of view in the world coordinates using the Umeyama method (1991). Thus, a gait in our case is a temporal sequence of view normalized skeletal graphs extracted per frame from a video. We now provide a formal definition for gait.

A gait is represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ denotes the set of vertices and $\mathcal{E}$ denotes the set of edges, such that

- $v_i^t \in \mathcal{V}$, $i \in \{1, \dots V\}$ represents the 3D position of the i-th joint in the skeleton at time step t and V is the total number of joints in the skeleton.

- $\mathcal{A}_i^t \subseteq \mathcal{V}$ is the set of all nodes that are adjacent to v_i^t as per the skeletal graph at time step t,

- $v_i := \{v_i^t\}_{t \in \{1, \dots, T\}}$ denotes the set of positions of of the i-th joint across all time steps $1 \dots T$,

- $(v_i^t, v_j^t) \in \mathcal{E}$, $\forall v_j^t \in \mathcal{A}_i^t \cup v_i$, $\forall t \in \{1, \dots, T\}$, $\forall i \in \{1, \dots, V\}$.

A key pre-requisite for using GCNs is to define the adjacency between the nodes in the graph (Bruna et al. 2013; Kipf and Welling 2016; Yan, Xiong, and Lin 2018). Note that as per our definition of gait, given fixed T and V, any pair of gaits $\mathcal{G}_x$ and $\mathcal{G}_y$ can have different sets of vertices, $\mathcal{V}_x$ and $\mathcal{V}_y$ respectively, but necessarily have the same edge set $\mathcal{E}$ and hence the same adjacency matrix A. This useful property of the definition allows us to maintain a unique notion
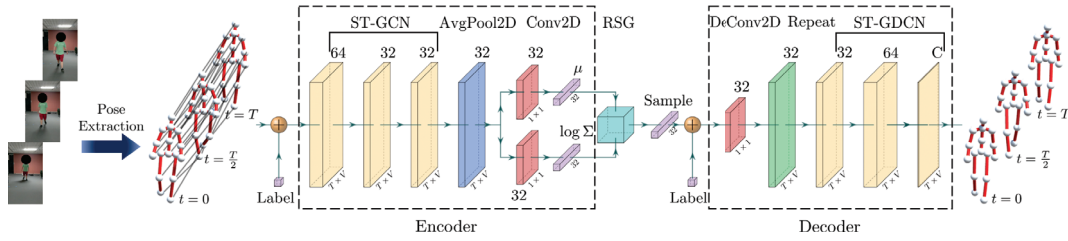
Figure 2: Our Generation Network (STEP-Gen): The encoder consists of ST-GCN, Average Pool and Conv2D layers. The decoder consists of DeConv2D, Repeat and ST-GDCN layers. RSG (Random Sample Generator) is used to generate random samples from the latent space. $+$ denotes appending; T: number of time steps (75 in our dataset); V: number of nodes (16 in our dataset); C: dimension of each node (3 in our dataset). Input: Human gaits processed from walking videos and corresponding emotion label. Spheres are nodes, thick red lines are spatial edges and thin gray lines are temporal edges. Output: Human gaits corresponding to the input label, with same T, V, and C.

of adjacency for all the gaits in a dataset, and thus develop ST-GCN-based networks for the dataset.

4.2 STEP-Gen: The Generation Network

We show our generative network in Figure 2. Our network architecture is based on the Conditional Variational Autoencoder (CVAE) (Sohn, Lee, and Yan 2015).

In the encoder, each $C \times T \times V$ dimensional input gait, pre-processed from a video (as per Section 4.1), is appended with the corresponding label, and passed through a set of 3 ST-GCN layers (yellow boxes). C=3 is the feature dimension of each node in the gait, representing the 3D position of the corresponding joint. The first ST-GCN layer has 64 kernels and the next two have 32 kernels each. The output from the last ST-GCN layer is average pooled along both the temporal and joint dimensions (blue box). Thus, the output of the pooling layer is a $32 \times 1 \times 1$ tensor. This tensor is passed through two 1×1 convolutional layers in parallel (red boxes). The outputs of the two convolutional layers are 32 dimensional vectors, which are the mean and the log-variance of the latent space respectively (purple boxes). All ST-GCN layers are followed by the ReLU nonlinearity, and all the layers are followed by a BatchNorm layer (not shown separately in Figure 2).

In the decoder, we generate random samples from the 32 dimensional latent space and append them with the same label provided with the input. As commonly performed in VAEs, we use the reparametrization trick (Kingma and Welling 2013) to make the overall network differentiable. The random sample is passed through a 1×1 deconvolutional layer (red box), and the output feature is repeated ("un-pooled") along both the temporal and the joint dimension (green box) to produce a $32 \times T \times V$ dimensional tensor. This tensor is then passed through 3 spatial temporal graph deconvolutional layers (ST-GDCNs) (yellow boxes). The first ST-GDCN layer has 32 kernels, the second one has 64 channels, and the last one has C=3 channels. Hence, we finally get a $C \times T \times V$ dimensional tensor at the output, which is a synthetic gait for the provided label. As in the encoder part, all ST-GDCN layers are followed by a ReLU nonlinearity, and all layers are followed by a BatchNorm layer (not shown separately in Figure 2).

Once the network is trained, we can generate new synthetic gaits by drawing random samples from the 32 dimensional latent distribution space parametrized by the learned μ and Σ.

The original CVAE loss $\mathcal{L}_o$ is given by:

$$\mathcal{L}_o = \sum_{t=1}^{T} \|v_R^t - v_S^t\|^2, \tag{3}$$

where $v^t = \begin{bmatrix} v_1^t & \dots & v_V^t \end{bmatrix}^\top$, where each v_i^t is assumed to be a row vector consisting of the 3D position of the joint i at frame t. The subscripts R and S stand for real and synthetic data respectively.

Each gait corresponds to a temporal sequence. Therefore, for any gait representation, it is essential to incorporate such temporal information. This is even more important as temporal changes in a gait provide significant cues for emotion perception (Kleinsmith and Bianchi-Berthouze 2013; Karg, Kuhnlenz, and Buss 2010; Crenn et al. 2016a). But, the baseline-CVAE architecture does not take into account the temporal nature of the gaits. We therefore modify the original reconstruction loss of the CVAE by adding regularization terms that enforce the desired temporal constraints (Equation 8).

We propose a novel "push-pull" regularization scheme. We first make sure that sufficient movement occurs in a generated gait across the frames so that the joint configurations at different time frames do not collapse into a single configuration. This is the "push" scheme. Simultaneously, we make sure that the generated gaits do not drift too far from the real gaits over time due to excessive movement. This is the "pull" scheme.

- *Push*: We require the synthetic data to resemble the joint velocities and accelerations of the real data as closely as possible. The velocity vel_i^t of a node i at a frame t can be approximated as the difference between the positions of the node at frames t and $t-1$, *i.e.*,

$$vel_i^t = v_i^t - v_i^{t-1} \tag{4}$$

Similarly, acceleration acc_i^t of a node i at a frame t can be approximated as the difference between the velocities of

the node at frame t and $t-1$, *i.e.*,

$$acc_i^t = vel_i^t - vel_i^{t-1} = v_i^t - 2v_i^{t-1} + v_i^{t-2} \quad (5)$$

We use the following loss for gait collapse:

$$\mathcal{L}_c = \sum_{t=2}^{T} \|vel_R^t - vel_S^t\|^2 + \sum_{t=3}^{T} \|acc_R^t - acc_S^t\|^2 \quad (6)$$

where $vel^t = \begin{bmatrix} vel_1^t & \dots & vel_V^t \end{bmatrix}^\top$ and $acc^t = \begin{bmatrix} acc_1^t & \dots & acc_V^t \end{bmatrix}^\top$.

- *Pull*: When the synthetic gait nodes are enforced to have non-zero velocity and acceleration between the frames, the difference between the synthetic node positions and the corresponding real node positions tends to increase as the number of frames increases. This is commonly known as the drift error. In order to constrain this error, we use the notion of anchor frames. At the anchor frames, we impose additional penalty on the loss between the real and synthetic gaits. In order to be effective, we need to ensure that there are a high number of anchor frames and they are as far apart as possible. Based on this trade off, we choose 3 anchor frames in the temporal sequence — the first frame, the middle frame and the last frame of the gait. We use the following loss function for gait drift:

$$\mathcal{L}_d = \sum_{t=1}^{T} \sum_{\omega \in \Omega} \|v_R^t - v_R^\omega - (v_S^t - v_S^\omega)\|^2 \quad (7)$$

where Ω denotes the set of anchor frames.

Finally, our modified reconstruction loss $\mathcal{L}_r$ of the CVAE is given by

$$\mathcal{L}_r = \mathcal{L}_o + \lambda_c \mathcal{L}_c + \lambda_d \mathcal{L}_d \quad (8)$$

where λ_c and λ_d are the regularization weights. Note that this modified loss function still satisfies the ELBO bound (Kingma and Welling 2013), if we assume that the decoder generates variables from a mixture of Gaussian distributions for every class, with the original loss, the push loss ad the pull loss representing the 3 Gaussian distributions in the mixture.

4.3 STEP: The Classification Network

We show out classifier network in Figure 3. In the base network, each input gait is passed through a set of 3 ST-GCN layers (yellow boxes). The first ST-GCN layer has 32 kernels and the next two have 64 kernels each. The output from the last ST-GCN layer is average pooled (blue box) in both the temporal and joint dimensions and passed through a 1×1 convolutional layer (red box). The output of the convolutional layer is passed through a fully connected layer of dimension 4 (corresponding to the 4 emotion labels that we have), followed by a softmax operation to generate the class labels. All the ST-GCN layers are followed by the ReLU nonlinearity and all layers except the fully connected layer are followed by a BatchNorm layer (not shown separately in Figure 3). We refer to this version of the network as the *Baseline-STEP*.
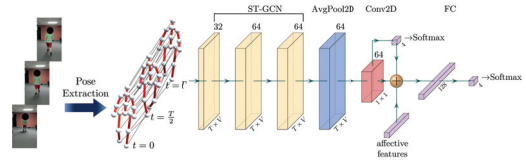

Figure 3: Our Classifier Network (STEP): It consists of ST-GCN, Average Pool, Conv2D and fully connected (FC) layers. + denotes appending. T: number of time steps (75 in our dataset); V: number of nodes (16 in our dataset); C: dimension of each node (3 in our dataset). Input: Human gaits processed from walking videos. Spheres are nodes, thick red lines are spatial edges and thin gray lines are temporal edges. Output: Predicted label after Softmax. The first Softmax from the left gives the output of Baseline-SETP, and the second Softmax gives the output of STEP.

Prior work in gait analysis has shown that affective features for gaits provide important information for emotion perception (Kleinsmith and Bianchi-Berthouze 2013; Karg, Kuhnlenz, and Buss 2010; Crenn et al. 2016a). Affective features are comprised of two types of features:

- Posture features. These include angle and distance between the joints, area of different parts of the body (*e.g.*, area of the triangle formed by the neck, the right hand and the left hand), and the bounding volume of the body.

- Movement features. These include the velocity and acceleration of individual joints in the gait.

We exploit the affective feature formulation (Kleinsmith and Bianchi-Berthouze 2013; Crenn et al. 2016b) in our final network. We append the 29 dimensional affective feature (purple box) to the final layer feature vector learned by our Baseline-STEP network, thus generating hybrid feature vectors. These hybrid feature vectors are passed through two fully connected layers of dimensions 128 and 4 respectively, followed by a softmax operation to generate the final class labels. We call this combined network STEP.

5 Experiments and Results

We list all the parameters and hardware used in training both our generation and classification networks in Section 5.1. In Section 5.2, we give details of our new dataset. In Sections 5.3, we list the standard metrics used to compare generative models and classification networks and in Section 5.4, we list the state-of-the-art methods against which we compare our algorithms. In Section 5.5, we present the evaluation results. Finally, in Section 5.6, we analyse the robustness of our system and show that both STEP and STEP-Gen do not overfit on the E-Gait Dataset.

5.1 Training Parameters

For training STEP-Gen, we use a batch size of 8 and train for 150 epochs. We use the Adam optimizer (Kingma and Ba 2014) with an initial learning rate of 0.1, which decreases to $\frac{1}{10}$-th of its current value after 75, 113 and 132 epochs.

We also use a momentum of 0.9 and and weight-decay of 5×10^{-4}.

For training STEP, we use a split of $7 : 2 : 1$ for training, validation and testing sets. We use a batch size of 8 and train for 500 epochs using the Adam optimizer (Kingma and Ba 2014) with an initial learning rate of 0.1. The learning rate decreases to $\frac{1}{10}$-th of its current value after 250, 375 and 438 epochs. We also use a momentum of 0.9 and and weight-decay of 5×10^{-4}. All our results were generated on an NVIDIA GeForce GTX 1080 Ti GPU.

5.2 Dataset: Emotion-Gait

Emotion-Gait (E-Gait) consists of $4,227$ real gaits and $1,000$ synthetic gaits each of the 4 emotion classes generated by STEP-Gen, for a total for $5,227$ gaits. We obtained the videos for the real gaits from various sources, including BML (Ma, Paterson, and Pollick 2006), Human3.6M (Ionescu et al. 2013), ICT (Narang et al. 2017), CMU-MOCAP (2018) and ELMD (Habibie et al. 2017), and converted all the input gaits to 21-joint skeletons following the procedure of Habibie et al. Each gait in the consolidated dataset was annotated by the same 10 annotators. The annotators were between 20 and 28 years old, with a median age of 23 years. 4 of the annotators were female and 6 were male. There was also an even mix of annotators belonging to the same culture as the subjects in the dataset as well as annotators from other, different cultures.

Based on the annotations, we found that the agreement score due to chance was 0.29, and the corresponding Fleiss' Kappa (FK) score (Fleiss 1971) was 0.45. Moreover, the FK score between only the sad and the neutral labels was -0.01 (< 0) and that between only the happy and the neutral labels was -0.03 (< 0), indicating a lack of disagreement between these two pairs of labels. This is expected, as these pairs of labels contain many marginal cases where the subjectivity of emotion perception is the most prominent. In our case, we picked the class with the maximum number of votes as the emotion label for the corresponding gait. If, however, the total number of votes in the top k classes were within 20% of each other, then we add the corresponding gait as an instance of all the top k classes. For example, if a gait was annotated as happy by 40% of the annotators and neutral by 60% of the annotators, we marked that gait as an instance of both happy and neutral.

5.3 Evaluation Metrics

Generation: For generative models, we compute the Fréchet Inception Distance (FID) score (Heusel et al. 2017) that measures how close the generated samples are to the real inputs while maintaining diversity among the generated samples. The FID score is computed using the following formula:

$$FID = \|\mu_x - \mu_g\|_2^2 + Tr(\Sigma_x + \Sigma_g + 2(\Sigma_x \Sigma_g)^{\frac{1}{2}}) \quad (9)$$

Classification: For classifier models, we report the classification accuracy given by $Accuracy = (TP + TN)/TD$, where TP, TN, TD are the number of true positives, true negatives, and total data, respectively.
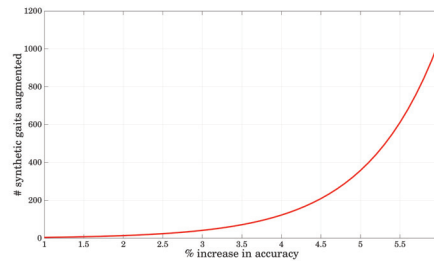

Figure 4: Data Augmentation: Effect of augmenting synthetically generated data to the train and test sets of STEP+Aug. For every percent improvement in accuracy, an exponentially larger amount of data needs to be augmented.

5.4 Evaluation Methods

Generation: We compare our generative network with both GAN- and VAE-based generative networks, as listed below.

- vid2vid (GAN-based) (Wang et al. 2018): This is the state-of-the-art video generation method. It can take human motion videos as input and generate high-resolution videos of the same motion.

- Baseline CVAE (VAE-based): We use a CVAE with the same network architecture as STEP-Gen, but with only the original CVAE loss given in Equation 3.

Classification: We compare our classifier network with both prior methods for emotion recognition from gaits, and prior methods for action recognition from gaits, as listed below.

- Emotion Recognition: We compare with the current state-of-the-art classifiers of (Karg, Kuhnlenz, and Buss 2010; Venture et al. 2014; Crenn et al. 2016a; Wang, Enescu, and Sahli 2016; Daoudi et al. 2017).

- Action Recognition: We compare with the state-of-the-art methods using both GCNs (Yan, Xiong, and Lin 2018) and LSTMs (Randhavane et al. 2019). The networks of both these methods were trained on our dataset before comparing the performance.

We also perform the following ablation experiments with our classifier network:

- Baseline-STEP: It predicts emotions based only on the network-learned features from gaits. This network is trained on the $4,227$ real gaits in E-Gait.

- STEP: This is our hybrid network combining affective features (Kleinsmith and Bianchi-Berthouze 2013; Crenn et al. 2016b) with the network-learned features of Baseline-STEP. This network is also trained on the $4,227$ real gaits in E-Gait.

- STEP+Aug: This is same as STEP, but trained on both the real and the synthetic gaits in E-Gait.

5.5 Results on E-Gait

Generation: All the generative networks are trained on the $4,227$ real data in E-Gait. We report an FID score of 11.11, while the FID score of Baseline-CVAE is 11.33. Lower FID
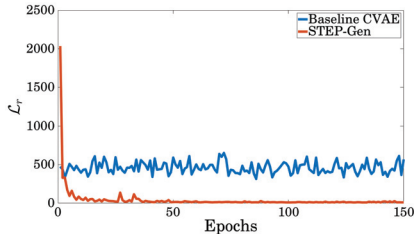
Figure 5: Training Loss Convergence: Our "Push-Pull" regularization loss (Equation 8) as a function of training epochs, as produced by the baseline-CVAE and our STEP-Gen. The baseline-CVAE fails to converge even after 150 epochs, while STEP-Gen converges in around 28 epochs.
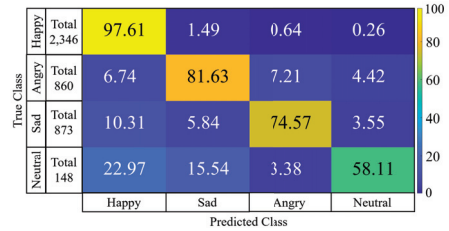

Figure 6: Accuracy Analysis of STEP+Aug on E-Gait dataset: Per class classification results over the 4, 227 gaits. We obtain a balanced accuracy score of 77.98%.

indicates higher fidelity to the real data. However, we also note that vid2vid (Wang et al. 2018) completely memorizes the dataset and thus gives an FID score of 0. This is undesirable for our task since we require the generative network to be able to produce diverse data that can be augmented to the training set of the classifier network.

Additionally, to show that our novel "Push-Pull" regularization loss function (Equation 8) generates gaits with joint movements, we measure the decay of the value of the loss function for the baseline-CVAE and STEP-Gen with time (Figure 5). We add the $\mathcal{L}_c$ and $\mathcal{L}_d$ terms from equation 8 (without optimizing them) to the baseline-CVAE loss function (Equation 3). We observe that STEP-Gen converges extremely quickly to a smaller loss value in around 28 epochs. On the other hand, the base-line CVAE produces oscillations and fails to converge as it does not optimize $\mathcal{L}_c$ and $\mathcal{L}_d$.

We also perform qualitative tests of gait generated by all the methods. vid2vid (Wang et al. 2018) uses GANs to produce high-quality videos. However, in our experiments, vid2vid memorizes the dataset and does not produce diverse samples. Baseline-CVAE produces static gaits that do not move in time. Finally, our gaits are both diverse (different from input) and realistic (successfully mimics walking motion). We show all these results in our project page.

Classification: In Table 1, we report the mean classification accuracies of all the methods using the formula in Section 5.3. We observe that most of the prior methods for emotion recognition from gaits have less than 60% accuracy on E-Gait. Only Crenn et al. (2016a), where the authors manually compute the same features we use in our novel "push-pull" regularization loss function (enforce *i.e.* distances between joints across time) has greater than 65% accuracy. The two prior action recognition from gait methods we compare

with have 65% and 75% accuracy respectively. By comparison, our Baseline-STEP has an accuracy of 78%. Combining network-learned and affective features in STEP gives an accuracy of 83%. Finally, augmenting synthetic gaits generated by STEP-Gen in STEP+Aug gives an accuracy of 88%.

To verify that our classification accuracy is statistically significant and not due to random chance, we perform two statistical tests:

- *Hypotheses Testing:* Classification as a task, depends largely on the test sample to be classified. To ensure that the classification accuracy of STEP is not achieved due to random positive examples, we determine the statistical likelihood of our results. Note that we do not test on STEP+Aug as accuracy of STEP+Aug is also dependent on the augmentation size. We generate a population of size 10, 000 accuracy values of STEP with mean 83.15 and standard deviation 6.9. We set $\mu = 83.15$, *i.e.* the reported mean accuracy of STEP as the null hypothesis, H_0. To accept our null hypothesis, we require the p-value to be greater than 0.50. We compute the p-value of this population as 0.78 > 0.50. Therefore, we fail to reject the null hypothesis, thus corroborating our classification accuracy statistically.

- *Confidence Intervals:* This metric determines the likelihood of a value residing in an interval. For a result to be statistically significant, we require a tight interval with high probability. With a 95% likelihood, we report a confidence interval of [81.19, 85.33] with a standard deviation of 1.96. Simply put, our classification accuracy will lie between 81.19 and 85.33 with a probability of 0.95.

Finally, we show the discriminatory capability of our classifier through a confusion matrix in Figure 6.

5.6 Overfitting Analysis

Effect of Generated Data on Classification: We show in Figure 4 that the synthetic data generated by STEP-Gen increases the classification accuracy of STEP+Aug. This, in turn, shows that STEP-Gen does not memorize the training dataset, but can produce useful diverse samples. Nevertheless, we see that to achieve every percent improvement in the accuracy of STEP+Aug, we need to generate an exponentially larger number of synthetic samples as training saturation sets in.
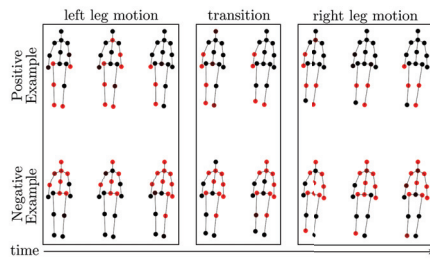
Venture et al. (2014)	Karg et al. (2010)	Daoudi et al. (2017)	Wang et al. (2016)	Crenn et al. (2016a)	ST-GCN (2018)	LSTM (2019)	Base-STEP	STEP	**STEP + Aug**
30.83	39.58	42.52	53.73	66.22	65.62	74.10	78.24	82.15	**88.22**

Table 1: Classification Accuracy Comparison: Accuracies are computed using the formula in Section 5.3 and shown in increasing order. We choose methods from both psychology and computer vision literature. Base-STEP and STEP+Aug are variations of STEP.

Figure 7: Saliency Map: Saliency map showing the magnitude of the network gradient along the z-axis (in and out of the paper) generated by our trained network, which is the direction of walking in both the examples shown. The examples are for the 'happy' emotion. In the positive example, the network focuses on the movements of the arms and the legs. These movements contain important emotional cues, thereby confirming that our classifier is learning meaningful features to recognize emotions accurately.

Saliency Maps: We show that STEP does not memorize the training dataset, but learns meaningful features, using saliency maps obtained via guided backpropagation on the learned network (Simonyan and Zisserman 2014; Springenberg et al. 2014). Saliency maps determine how the loss function output changes with respect to a small change in the input. In our case, the input consists of 3D joint positions over time, therefore, the corresponding saliency map highlights the joints that most influence the output. Intuitively, we expect the saliency map for a positively classified example to capture the joint movements that are most important for predicting the perceived emotion from a psychological point of view (Crenn et al. 2016a). We show the saliency map given by our trained network for both a positively classified and a negatively classified example for the label 'happy' in Figure 7. The saliency map only shows magnitude of the gradient along the z-axis (in and out of the plane of the paper), which is the direction of walking in both the examples. Black represents zero magnitude, and bright red represents a high magnitude. In the positive example, we see that the network focuses on the movements of the arms and the legs. This is the expected behavior, as the movement of hands and the stride length and speed are important cues for emotion perception (Crenn et al. 2016a). By the contrast, there is no intuitive pattern to the detected movements in the saliency map for the negative example.

6 Limitations and Future Work

Our generative model is currently limited to generating gait sequences of a single person. The accuracy of the classification algorithm is also governed by the quality of the video and the pose extraction algorithm. There are many avenues for future work as well. We would like to extend the approach to deal with multi-person or crowd videos. Given the complexity of generating annotated real-world videos, we need better generators to improve the accuracy of classification algorithm. To this end, coupling the generator and the classifier in a semi-supervised manner, similar to that of Bhattacharya et al., is a useful future direction.

7 Acknowledgements

References

Albanie, S.; Nagrani, A.; Vedaldi, A.; and Zisserman, A. 2018. Emotion recognition in speech using cross-modal transfer in the wild. *arXiv:1808.05561*.

Arunnehru, J., and Geetha, M. K. 2017. Automatic human emotion recognition in surveillance video. In *ITSPMS*. Springer. 321–342.

Atcheson, M.; Sethu, V.; and Epps, J. 2017. Gaussian process regression for continuous emotion recognition with global temporal invariance. In *IJCAI-W*, 34–44.

Bauer, A., et al. 2009. The autonomous city explorer: Towards natural human-robot interaction in urban environments. *IJSR* 1(2):127–140.

Bhattacharya, U.; Roncal, C.; Mittal, T.; Chandra, R.; Bera, A.; and Manocha, D. 2019. Take an emotion walk: Perceiving emotions from gaits using hierarchical attention pooling and affective mapping. *arXiv preprint arXiv:1911.08708*.

Bruna, J.; Zaremba, W.; Szlam, A.; and LeCun, Y. 2013. Spectral networks and locally connected networks on graphs. *arXiv:1312.6203*.

Cai, H.; Bai, C.; Tai, Y.-W.; and Tang, C.-K. 2018. Deep video generation, prediction and completion of human action sequences. In *ECCV*, 366–382.

Chiu, M.; Shu, J.; and Hui, P. 2018. Emotion recognition through gait on mobile devices. In *PerCom Workshops*, 800–805. IEEE.

CMU-MOCAP. 2018. Cmu graphics lab motion capture database. *http://mocap.cs.cmu.edu/*.

Crenn, A.; Khan, R. A.; Meyer, A.; and Bouakaz, S. 2016a. Body expression recognition from animated 3d skeleton. In *IC3D*, 1–7. IEEE.

Crenn, A.; Khan, R. A.; Meyer, A.; and Bouakaz, S. 2016b. Body expression recognition from animated 3d skeleton. In *IC3D*, 1–7. IEEE.

Dabral, R.; Mundhada, A.; Kusupati, U.; Afaque, S.; Sharma, A.; and Jain, A. 2018. *Learning 3D Human Pose from Structure and Motion*. Lecture Notes in Computer Science. Springer International Publishing.

Daoudi, M.; Berretti, S.; Pala, P.; Delevoye, Y.; and Del Bimbo, A. 2017. Emotion recognition by body movement representation on the manifold of symmetric positive definite matrices. In *ICIAP*, 550–560. Springer.

Deng, J.; Xu, X.; Zhang, Z.; Frühholz, S.; and Schuller, B. 2018. Semisupervised autoencoders for speech emotion recognition. *IEEE/ACM ASLP* 26(1):31–43.

Denham, S. A.; Workman, E.; Cole, P. M.; Weissbrod, C.; Kendziora, K. T.; and ZAHN-WAXLER, C. 2000. Prediction of externalizing behavior problems from early to middle childhood. *Development and Psychopathology* 12(1):23–45.

Ekman, P., and Friesen, W. V. 1967. Head and body cues in the judgment of emotion: A reformulation. *Perceptual and motor skills*.

Ekman, P. 1993. Facial expression and emotion. *American psychologist* 48(4):384.

Fabian Benitez-Quiroz, C.; Srinivasan, R.; and Martinez, A. M. 2016. Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. In *CVPR*.

Fan, Y.; Lu, X.; Li, D.; and Liu, Y. 2016. Video-based emotion recognition using cnn-rnn and c3d hybrid networks. In *ICMI*, 445–450. ACM.

Feichtenhofer, C.; Pinz, A.; and Wildes, R. 2016. Spatiotemporal residual networks for video action recognition. In *NIPS*, 3468–3476.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2016. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 1933–1941.

Fernández-Dols, J.-M., and Ruiz-Belda, M.-A. 1995. Expression of emotion versus expressions of emotions. In *Everyday conceptions of emotion*. Springer. 505–522.

Fleiss, J. L. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin* 76(5):378.

Girdhar, R.; Gkioxari, G.; Torresani, L.; Paluri, M.; and Tran, D. 2017. Detect-and-track: Efficient pose estimation in videos. *CoRR* abs/1712.09184.

Habibie, I.; Holden, D.; Schwarz, J.; Yearsley, J.; and Komura, T. 2017. A recurrent variational autoencoder for human motion synthesis. In *BMVC*.

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 6626–6637.

Ionescu, C.; Papava, D.; Olaru, V.; and Sminchisescu, C. 2013. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *PAMI* 36(7):1325–1339.

Jacob, A., and Mythili, P. 2015. Prosodic feature based speech emotion recognition at segmental and supra segmental levels. In *SPICES*, 1–5. IEEE.

Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2013. 3d convolutional neural networks for human action recognition. *PAMI* 35(1):221–231.

Karg, M.; Kuhnlenz, K.; and Buss, M. 2010. Recognition of affect based on gait patterns. *Cybernetics* 40(4):1050–1061.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv:1412.6980*.

Kingma, D. P., and Welling, M. 2013. Auto-encoding variational bayes. *arXiv:1312.6114*.

Kipf, T. N., and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv:1609.02907*.

Kleinsmith, A., and Bianchi-Berthouze, N. 2013. Affective body expression perception and recognition: A survey. *IEEE Transactions on Affective Computing* 4(1):15–33.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*, 1097–1105.

Lea, C.; Flynn, M. D.; Vidal, R.; Reiter, A.; and Hager, G. D. 2017. Temporal convolutional networks for action segmentation and detection. In *CVPR*, 156–165.

Ma, Y.; Paterson, H. M.; and Pollick, F. E. 2006. A motion capture library for the study of identity, gender, and emotion perception from biological motion. *Behavior Research Methods* 38(1):134–141.

Majumder, A.; Behera, L.; and Subramanian, V. K. 2014. Emotion recognition from geometric facial features using self-organizing map. *Pattern Recognition* 47(3):1282–1293.

Michalak, J.; Troje, N. F.; Fischer, J.; Vollmar, P.; Heidenreich, T.; and Schulte, D. 2009. Embodiment of sadness and depression—gait patterns associated with dysphoric mood. *Psychosomatic Medicine* 71(5):580–587.

Narang, S.; Best, A.; Feng, A.; Kang, S.-h.; Manocha, D.; and Shapiro, A. 2017. Motion recognition of self and others on realistic 3d avatars. *Computer Animation and Virtual Worlds* 28(3-4):e1762.

Quigley, K. S.; Lindquist, K. A.; and Barrett, L. F. 2014. *Inducing and measuring emotion and affect: Tips, tricks, and secrets.* Cambridge University Press.

Randhavane, T.; Bera, A.; Kapsaskis, K.; Bhattacharya, U.; Gray, K.; and Manocha, D. 2019. Identifying emotions from walking using affective and deep features. *arXiv:1906.11884*.

Schurgin, M.; Nelson, J.; Iida, S.; Ohira, H.; Chiao, J.; and Franconeri, S. 2014. Eye movements during emotion recognition in faces. *Journal of vision* 14(13):14–14.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*.

Sohn, K.; Lee, H.; and Yan, X. 2015. Learning structured output representation using deep conditional generative models. In *NIPS*, 3483–3491.

Springenberg, J. T.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. 2014. Striving for simplicity: The all convolutional net. *arXiv:1412.6806*.

Tulyakov, S.; Liu, M.-Y.; Yang, X.; and Kautz, J. 2018. Mocogan: Decomposing motion and content for video generation. In *CVPR*, 1526–1535.

Umeyama, S. 1991. Least-squares estimation of transformation parameters between two point patterns. *TPAMI* 376–380.

Venture, G.; Kadone, H.; Zhang, T.; Grèzes, J.; Berthoz, A.; and Hicheur, H. 2014. Recognizing emotions conveyed by human gait. *IJSR* 6(4):621–632.

Wang, L.; Tan, T.; Hu, W.; and Ning, H. 2003. Automatic gait recognition based on statistical shape analysis. *TIP* 12(9):1120–1131.

Wang, T.-C.; Liu, M.-Y.; Zhu, J.-Y.; Liu, G.; Tao, A.; Kautz, J.; and Catanzaro, B. 2018. Video-to-video synthesis. In *NeurIPS*.

Wang, W.; Enescu, V.; and Sahli, H. 2016. Adaptive real-time emotion recognition from body movements. *TiiS* 5(4):18.

Yan, S.; Xiong, Y.; and Lin, D. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*.

Yang, C.; Wang, Z.; Zhu, X.; Huang, C.; Shi, J.; and Lin, D. 2018. Pose guided human video generation. In *ECCV*, 201–216.

Yang, H.; Ciftci, U.; and Yin, L. 2018. Facial expression recognition by de-expression residue learning. In *CVPR*, 2168–2177.

Yates, H.; Chamberlain, B.; Norman, G.; and Hsu, W. H. 2017. Arousal detection for biometric data in built environments using machine learning. In *IJCAI-W*, 58–72.

Zhang, Z., and Troje, N. F. 2005. View-independent person identification from human gait. *Neurocomputing* 69(1-3):250–256.

Zhang, F.; Zhang, T.; Mao, Q.; and Xu, C. 2018. Joint pose and expression modeling for facial expression recognition. In *CVPR*, 3359–3368.

Zhao, M.; Adib, F.; and Katabi, D. 2016. Emotion recognition using wireless signals. In *ICMCN*, 95–108. ACM.