# Skeleton-Based Typing Style Learning For Person Identification

Lior Gelberg
Tel-Aviv University
Tel-Aviv, Israel
liorgelberg@mail.tau.ac.il

David Mendlovic
Tel-Aviv University
Tel-Aviv, Israel
mend@eng.tau.ac.il

Dan Raviv
Tel-Aviv University
Tel-Aviv, Israel
darav@tauex.tau.ac.il

## Abstract

*We present a novel approach for person identification based on typing-style, using a novel architecture constructed of adaptive non-local spatio-temporal graph convolutional network. Since type style dynamics convey meaningful information that can be useful for person identification, we extract the joints positions and then learn their movements' dynamics. Our non-local approach increases our model's robustness to noisy input data while analyzing joints locations instead of RGB data provides remarkable robustness to alternating environmental conditions,* e.g.*, lighting, noise,* etc.*. We further present two new datasets for typing style based person identification task and extensive evaluation that displays our model's superior discriminative and generalization abilities, when compared with state-of-the-art skeleton-based models.*

## 1. Motivation

User identification and continuous user identification are some of the most challenging open problems we face today more than ever in the working-from-home lifestyle due to the COVID-19 pandemic. The ability to learn a style instead of a secret passphrase opens up a hatch towards the next level of person identification, as style is constructed from a person's set of motions and their relations, which are typically indifferent to most scene properties. Therefore, analyzing style, rather than relying on appearance (or some other easily fooled characteristic), can increase the level of security in numerous real-world applications, *e.g.*, VPN, online education, finance, *etc.*.

We focus on a typical daily task - typing, as a method for identification, and present a substantial amount of experiments supporting typing style as a strong indicator of a person's identity, as appeared in fig. 1. Now, typing someone's password is insufficient, but typing it in a similar style is needed. Therefore, our typing style-based identification approach can offer an elegant and natural solution for both identification and continuous identification tasks.

## 2. Introduction

Biometrics are the physical and behavioral characteristics that make each one of us unique. Therefore, this kind of character is a natural choice for a person identity verification. Unlike passwords or keys, biometrics cannot be lost or stolen, and in the absence of physical damage, it offers a reliable way to verify someone's identity. Physiological biometrics involves biological input or measurement of other unique characteristics of the body. Such methods are fingerprint [11], and face geometry [1, 24]. Unlike physiological characteristics, behavioral characteristics encompass both physiological and psychological states. Human behavior is revealed as motion patterns in which their analysis forms the basis for dynamic biometric.

Motion analysis is drawing increasing attention due to a substantial improvement in performance it provides in a variety of tasks [26],[9], [34], [16], [28]. Motion patterns convey meaningful information relevant to several applications such as surveillance, gesture recognition, action recognition, and many more. These patterns can indicate the type of action within these frames, even manifesting a person's mood, intention, or identity.

Deep learning methods are the main contributors to the performance gain in analyzing and understanding motion that we have witnessed during recent years. Specifically, spatio-temporal convolutional neural networks that can learn to detect motion and extract high-level features from these patterns become common approaches in various tasks. Among them, video action classification (VAC), in which given a video of a person performing some action, the model needs to predict the type of action in the video. In this work, we take VAC one step further. Instead of predicting the action in the input video, we eliminate all action classes and introduce a single action - typing. Now, given a set of videos containing hands typing a sentence, we classify the videos according to the person typing the sentence.

Over time, researchers in VAC's field presented various approaches, where some use RGB-based 2D or 3D convolutions [26, 3, 6] while others focus on skeleton-based spatio-temporal analysis [30, 7, 18]. The skeleton-based approach

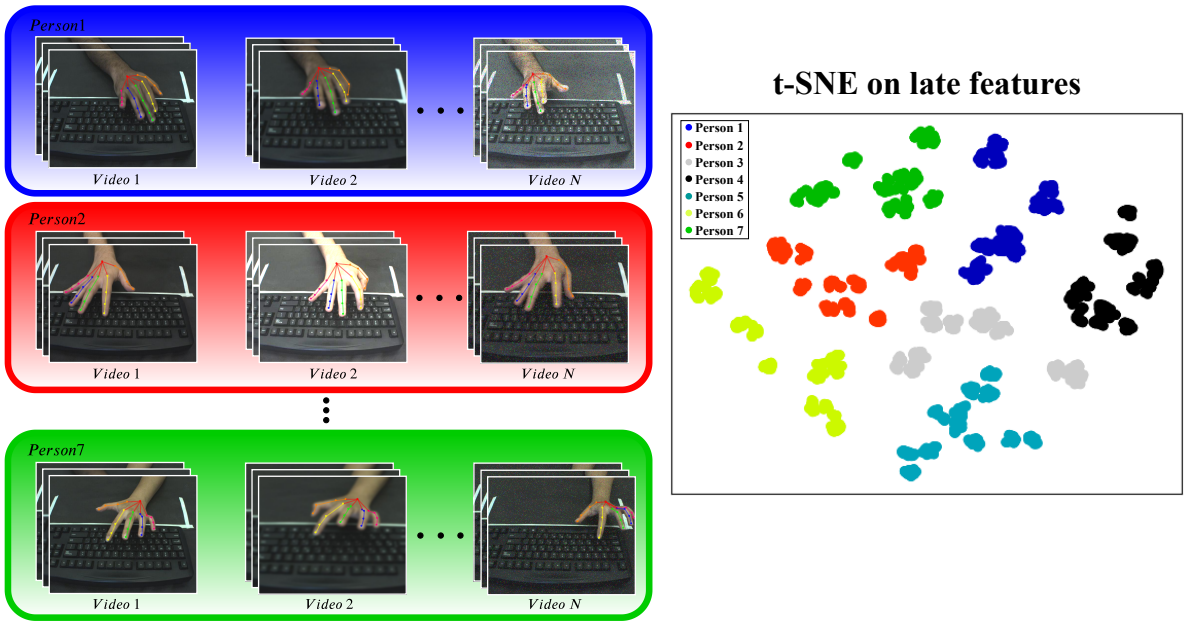



Figure 1. t-SNE on late features of 7 out of 60 people appears in *60Typing10* dataset, where some videos went through data augmentation to simulate changing environmental conditions. Given a video of a person typing a sentence, our model can classify the person according to its unique dynamic, *i.e.*, typing style, with high accuracy, regardless of scene properties (*e.g.*, lighting, noise, *etc.*). The model generalizes the typing style to other sentences, which it never saw during training even when it trains on one sentence type alone, while our non-local approach provides remarkable robustness to noisy data resulting from joints detector failures. Best viewed in color.

proved its efficiency in cases where the videos are taken under uncontrolled scene properties or in the presence of a background that changes frequently. The skeleton data is captured by either using a depth sensor that provides joint $(x, y, z)$ location or by using a pose estimator such as [2], that extracts the skeleton data from the RGB frames. The joint locations are then forwarded to the model that performs the action classification.

Recent works in the field of skeleton-based VAC uses architectures of *Spatio Temporal Graph Convolutional Network* (GCN) as graph-based networks are the most suitable for skeleton analysis since GCN can learn the dependencies between correlated joints. Since Kipf and Welling introduced GCN in their work [15], other works such as [35] presented adapted versions of GCN that applied for action classification. These adaptations include spatio-temporal GCN that performs an analysis in space and time and adaptive graphs that use a data-driven learnable adjacency matrix. Recently, a two-stream approach [25, 31] that uses both joints and bones data is gaining attention. Bones data is a differential version of the joints locations data since it is constructed from subtractions between linked joints. The bones vector contains each bone's length and direction, so analyzing this data is somewhat similar to how a human is analyzing motion. Furthermore, bones can offer new correlated yet complementary data to the joints locations. When combining both joints and bones, the model is provided with much more informative input data, enabling it to learn meaningful information that could not be achieved with a one-stream approach alone.

Even though VAC is a highly correlated task to ours, there are some critical differences. The full-body skeleton is a large structure. Its long-ranged joints relations are less distinct than those that appear in a human hand, which has strong dependencies between the different joints due to its biomechanical structure. These dependencies cause each joint's movement to affect other joints as well, even those on other fingers. Thus, when using a GCN containing a fixed adjacency matrix, we limit our model to a set of pre-defined connections and do not allow it to learn the relations between joints which are not directly connected. Furthermore, the hand's long-ranged dependencies that convey meaningful information tend to be weaker than the close-range ones, and unless these connections are amplified, we lose essential information. Our constructed modules are designed to increase vertices and edges inter (non-local) connections, allowing our model to learn non-trivial dependencies and to extract motion patterns from several scales in time, which we refer to as style.

In practice, we use a learnable additive adjacency matrix and a non-local operation that increases the long-range dependencies in the layer's unique graph. The spatial non-

local operation enables the GCN unit to permute forward better spatial features, and the temporal non-local operation provides the model with a new order of information by generating the inter joints relation in time. Now, each joint interacts with all other joints from different times as well. These dependencies in time help the model gain information regarding the hand and finger posture along time and the typing division among the different fingers. We further apply a downsampler learnable unit to sum each channel information into a single value while causing minimal information loss. As a result, the refined features resulting from the long-ranged dependencies can be reflected as much as possible in the model's final prediction layer. Also, we follow the two-stream approach and apply bones data to a second stream of our model. We train both streams jointly and let the data dictate the relationship between both streams, *i.e.*, we apply learnable scalars that set each stream's contribution.

The final model is evaluated on two newly presented datasets gathered for typing style learning for person identification (person-id) task. Since this work offers a new task, we present comprehensive comparisons with state-of-the-art skeleton-based VAC models to prove our model's superiority. The main contributions of our work are in four folds:

1. Develop a Spatio-Temporal GCN (*StyleNet*) for the task of typing style learning which outperforms all compared models in all experiments performed under controlled environmental conditions.
2. Present substantially better robustness to challenging environmental conditions and noisy input data than all compared state-of-the-art VAC models.
3. Introduce two new datasets for *typing style learning for person-id* task, which will become publicly available.
4. Introduce an innovative perspective for person-id based on joints locations while typing a sentence.

## 3. Background

AI methods entering the game allow for higher accuracy in various tasks, moving for axiomatic methods towards data-driven approaches. These models focus on the detection of minor changes that were missed earlier by examining dramatically more data. The hardware improvement allowed us to train deeper networks in a reasonable time and classify in real-time using these complex models. This paper's related works can refer to biometric-based person identification and skeleton-based action recognition.

Biometrics-based person identification methods using different techniques and inputs were presented over the years. [14, 20, 21] presented an approach for person identification that uses Keystroke dynamics as an indicator for discriminative purposes. Fong *et al.* [10], suggested identifying a person by geometric measurements of the user's stationary hand gesture of hand sign language. Roth *et al.* [23] presented an online user verification based on hand geometry and angle through time. Unlike [23], our method does not treat the hand as one segment but as a deformable part model by analyzing each of the hand joints relations in space and time. Furthermore, our method is more flexible since it is not based on handcrafted features and does not require a gallery video to calculate a distance for its decision.

Skeleton-based action recognition methods are going through a significant paradigm shift in recent years. This shift involves moving from hand-designed features [4, 17, 32, 13, 22, 12] to deep neural network approaches that learn features and classify them in an end-to-end manner. Most skeleton-based method uses GCN architectures as well as joints locations as input instead of the RGB video. Yan *et al.* [35] presented their spatio-temporal graph convolutional network that directly models the skeleton data as the graph structure. Shi *et al.* [25] presented their adaptive graph two-stream model that uses both joints coordinates and bones vectors for action classification and based on the work of [19] that introduced adaptive graph learning.

Inspired by the works presented above, this work follows skeleton-based methods for person-id task based on his typing style. Unlike full-body analysis, hand typing style analysis has higher discriminating requirements, which can be fulfilled by better analysis of the hand's global features such as the hand's posture and the fingers intra-relationships and inter-relationships in space-time. We claim that all skeleton-based methods presented earlier in this section fail to fulfill these discriminative requirements fully. Therefore, we propose a new architecture that aggregates non-locality with spatio-temporal graph convolution layers. Overall, we explored person-id on seen and unseen sentences under different scenarios.

## 4. StyleNet

The human hand is made from joints and bones that dictate its movements. Therefore, to analyze the hand's movements, a Graph Convolutional Network (GCN) is the preferred choice for deep neural network architecture in that case. GCN can implement the essential joints links, sustain the hand's joints hierarchy, and ignore links that do not exist.

### 4.1. Spatial Domain

Motivated by [35], we first formulate the graph convolutional operation on vertex $v_i$ as

$$f^S_{out}(v_i) = \sum_{v_j \in \mathbb{B}_i} \frac{1}{Z_{ij}} f^S_{in}(v_j) \cdot w(l_i(v_j)), \qquad (1)$$

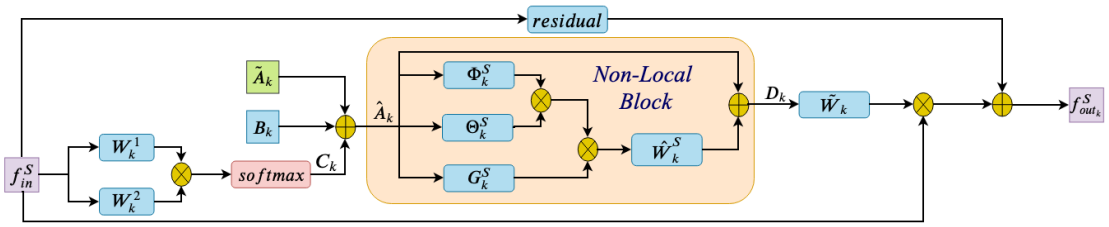



Figure 2. Diagram of our spatial Non-Local GCN unit. Blue rectangles are for trainable parameters. $\bigotimes$ denotes matrix multiplication and $\bigoplus$ denotes element-wise summation. $residual$ block exist only when the unit's $Ch_{in} \neq Ch_{out}$. This unit repeated $K_v$ times according to the number of subsets, Therefore, $F^S_{out} = \sum_{k=1}^{K_v} f^S_{out_k}$.

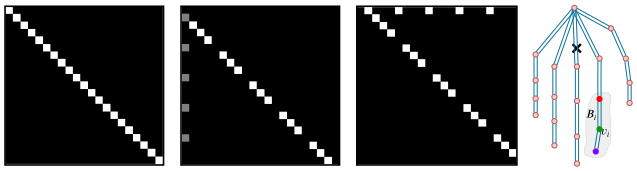


Figure 3. Left to right - adjacency matrix of the 1st, 2nd, and 3rd subset, respectively. Right - The hand as a graph. Each circle denotes a joint, and each blue line is a bone connecting two linked joints, i.e., each joint is a vertex, and bones are links in the graph. Black X marks the center of gravity. Gray blob is the subset $B_i$ of joint $v_i$ and its immediate neighbors. The green joint is $v_i$, the joint in red is the immediate neighbor of $v_i$ that is closer to the center of gravity, and the joint in purple is the immediate neighbor of $v_i$ that is farther from the center of gravity.

where $f^S_{in}$ is the input feature map and superscript $S$ refers to the spatial domain. $v$ is a vertex in the graph and $B_i$ is the convolution field of view which include all immediate neighbor $v_j$ to the target vertex $v_i$. $w$ is a weighting function operates according to a mapping function $l_i$. We followed the partition strategy introduced in [15] and construct the mapping function $l_i$ as follows: given a hand center of gravity (shown in Figure 3), for each vertex $v_i$ we define a set $B_i$ that include all immediate neighbors $v_j$ to $v_i$. $B_i$ is divided to 3 subsets, where $B^1_i$ is the target vertex $v_i$, $B^2_i$ is the subset of vertices in $B_i$ that are closer to the center of gravity and $B^3_i$ is the subset that contains all vertices in $B_i$ that are farther from the center of gravity. According to this partition strategy, each $v_j \in B_i$, is mapped by $l_i$ to its matching subset. $Z_{ij}$ is the cardinality of the subset $B^k_i$ that contains $v_j$. We follow [5, 15] method for graph convolution using polynomial parametrization and define a normalized adjacency matrix $A$ of the hand's joints by

$$\tilde{A} = \Lambda^{-\frac{1}{2}}(A + I)\Lambda^{-\frac{1}{2}}, \tag{2}$$

where $I$ is the identity matrix representing self connections, $A$ is the adjacency matrix representing the connections between joints, and $\Lambda$ is the normalization matrix, where $\Lambda_{ii} = \sum_j A_{ij}$. Therefore, $\tilde{A}$ is the normalized adjacency matrix, where its non diagonal elements, *i.e.*, $\tilde{A}_{ij}$ where $i \neq j$ indicate whether the vertex $v_j$ is connected to vertex $v_i$. Using eq.1 and eq. 2 we define our spatial non-local graph convolutional (Figure 2) operation as

$$F^S_{out} = \sum_k^{K_v} W_k f^S_{in} D_k, \tag{3}$$

where $K_v$ is the total number of subsets and is equal to 3 in our case. $W_k$ is a set of learned parameters, and $f^S_{in}$ is the input feature map. Inspired by [33], we construct $D_k$ by

$$D_k = \hat{W}^S_k((\Theta^S_k(\hat{A}_k))^T \cdot \Phi^S_k(\hat{A}_k))G^S_k(\hat{A}_k)) + \hat{A}_k, \tag{4}$$

where superscript $S$ denotes spatial domain. $\Phi^S_k, \Theta^S_k$, and $G^S_k$ are trainable 1D convolutions. These convolutions operate on the graph and embed their input into a lower-dimensional space, where an affinity between every two features is calculated. $\hat{W}^S_k$ is a trainable 1D convolution used to re-project the features to the higher dimensional space of $\hat{A}_k$. We use eq. 4 to apply self-attention on the input signal to enhances the meaningful connections between the features of its input $\hat{A}_k$, especially the long-range ones. To construct the input signal $\hat{A}_k$, we adopt a similar approach to [25] and define $\hat{A}_k$ to be

$$\hat{A}_k = \tilde{A}_k + B_k + C_k, \tag{5}$$

where $\tilde{A}_k$ is the normalized adjacency matrix of subset k according to eq. 2. This matrix is used for extracting only the vertices directly connected in a certain subset of the graph. $B_k$ is an adjacency matrix with the same size as $\tilde{A}$ initialized to zeros. Unlike $\tilde{A}_k$, $B_k$ is learnable and optimized along with all other trainable parameters of the model. $B_k$ is dictated by the training data, and therefore, it can increase the model's flexibility and make it more suitable for a specific given task. $C_k$ is the sample's unique graph constructed by the normalized embedded Gaussian that calculates the similarity between all vertices pairs according to

$$C_k = softmax((W^1_k f^S_{in})^T W^2_k f^S_{in}), \tag{6}$$

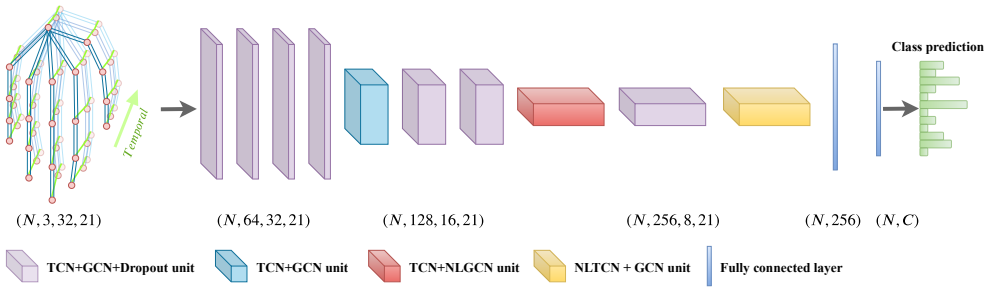



Figure 4. Single stream *StyleNet* architecture. Input is consists of the 21 coordinates of the hand's joints, while for each joint, we provide a 2D location and a confidence level of its location per frame. The blue lines represent the joints' spatial connections, while the green lines represent the joints' temporal connections. (N,Ch,T,V) Placed under the layers denote Batch size, the number of channels, temporal domain length, and V denotes the joint's index and represents a vertex in the graph, respectively. As for the fully connected layers, N denotes the batch size, and C is the dataset's number of classes.

where $W_k^1$ and $W_k^2$ are trainable parameters that embed the input features to a lower-dimensional space, $softmax$ used for normalizing the similarity operation's output and superscript $S$ denotes spatial domain. $C_k$ is somewhat related to $D_k$ in the way they are both constructed. The main difference is that $C_k$ is generated by the input features alone, while $D_k$ is generated using the input features, the learned adjacency matrix $B_k$, and the normalized adjacency matrix $\tilde{A}_k$. We use the non-local operation on the addition of $\tilde{A}_k$, $B_k$ and $C_k$ to exploit the information from all three matrices. This information enables the spatial block to permute more meaningful information forward, which contributes to the model's discriminative ability.

### 4.2. Temporal Domain

To better exploit the time domain, we place a temporal unit after each spatial GCN unit for better processing longitudinal information. We define $X$ to be $X = Conv(F_{out}^S)$, where $Conv$ is 2D convolution with kernel size of $9 \times 1$ and $F_{out}^S$ is the spatial unit output. A temporal non-local operation is applied on $X$ according to

$$F_{out}^{\tilde{T}} = W_{\tilde{T}}((\Theta_{\tilde{T}}(X)^T \cdot \Phi_{\tilde{T}}(X)) \cdot G_{\tilde{T}}(X)) + X, \quad (7)$$

where $\tilde{T}$ denotes the temporal domain. Unlike the spatial non-local operation, here $\Phi_{\tilde{T}}$, $\Theta_{\tilde{T}}$, and $G_{\tilde{T}}$ are trainable 2D convolutions, since they process the temporal domain and not part of the graph. These convolutions are used to embed their input into a lower-dimensional space. Similarly, $W_{\tilde{T}}$ is a trainable 2D convolution used to re-project the features to the higher dimensional space of $X$. The temporal non-local operation is used for two reasons: First, to better utilize the temporal information regarding the same joint in different places in time. Second, to construct the temporal relations between the different joints through the temporal domain.

### 4.3. Downsampling Unit

We further apply a downsampling unit before the classification layer. This unit receives the last temporal unit's output and downsamples each channel into a single value instead of using max or mean pooling. It is constructed from *[fully-connected,batch-normalization,fully-connected]* layers and shared among all channels. The benefit of using this sampling method is that it enables our model to learn summarizing each channel into a single value while minimizing the loss of essential features. *StyleNet* architecture for one stream is presented in fig. 4.

### 4.4. Joint decision

Encouraged by the work of [25], we adopt their two-stream approach and introduce *StyleNet*. This ensemble model consists of one stream that operates on the joints location, and the other that operates on the bone vectors. The final prediction is constructed according to

$$prediction = \alpha \cdot Output_{Joints} + \beta \cdot Output_{Bones}, \quad (8)$$

where both $\alpha$ and $\beta$ are trainable parameters that decide on each stream weight for the final prediction. This weighting method increases the model's flexibility since the training data itself determines the weight of each stream. We ensemble the bones data by subtracting pairs of joints coordinates that are tied by a connection in the graph. Therefore, the bones data is a differential version of the joints data, *i.e.*, the high frequencies of the joints data.

## 5. Experiments

Since there is no dataset for the suggested task, we created *80Typing2* and *60Typing10* datasets for the evaluation of our model. We compared our model with other skeleton-based action classification models using these datasets under various test cases, simulating user identification, and continuous user identification tasks. In 5.1 we present our

| [4,2,4] | |
|---|---|
| Model | Acc(%) |
| HCN [18] | 91.98 |
| STGCN [35] | 97.09 |
| 3sARGCN [27] | 95.8 |
| PBGCN [29] | 98.9 |
| 2sAGCN [25] | 99.04 |
| StyleNet | **99.84** |

| [3,2,5] | |
|---|---|
| Model | Acc(%) |
| HCN [18] | 84.16 |
| STGCN [35] | 97.21 |
| 3sARGCN [27] | 93.6 |
| PBGCN [29] | 98.6 |
| 2sAGCN [25] | 98.82 |
| StyleNet | **99.77** |

| [2,2,6] | |
|---|---|
| Model | Acc(%) |
| HCN [18] | 79.53 |
| STGCN [35] | 94.94 |
| 3sARGCN [27] | 91.35 |
| PBGCN [29] | 96.94 |
| 2sAGCN [25] | 97.97 |
| StyleNet | **99.5** |

Table 1. Test accuracy of user classification on unseen sentences on *60Typing10*. $[\alpha, \beta, \gamma]$ denotes number of sentences for train, validation and test, respectively

new datasets and our main experiments results presented in 5.2 and 5.3. We evaluate our model under challenging scenarios such as noisy input data 5.4 and presents the skeleton-based approach superiority over RGB modality in 5.5. In 5.6, we provide an additional evaluation using 3D input data taken from *How We Type* dataset [8]. In the supplementary material, we provide our implementation and training details, and present an ablation study which analyzes the contribution of each of our components.

In all experiments, we split our data between train, validation, and test sets randomly according to the experiment's settings for an accurate evaluation of the models. Each input video consists of 32 sampled frames from the entire video. We tested each trained model for tens of times and set its accuracy according to all tests' mean accuracy. It is crucial to evaluate each trained model several times since we sample only 32 frames and not use the entire video.

### 5.1. 80Typing2 and 60Typing10 datasets

We present two new datasets created for *typing style learning for person identification* task. The datasets recorded using a simple RGB camera with 100 fps for *80Typing2* and 80 fps for *60Typing10*. No special lighting was used, and the camera's position remained fixed through all videos. No jewelry or any other unique clues appear in the videos. Both men and women, as well as right and left-handed, appear in the dataset. All participants were asked to type the sentences with their dominant hand only. We chose sentences that use a large variety of keyboards letters to encourage hand movement.

*80Typing2* dataset consists of 1600 videos of 80 participants. Each participant typed two different sentences, and each sentence was repeated ten times. This setting's main purpose is simulating a scenario where a small number of different sentences and many repetitions from each sentence are provided. As each person encounters a changing level of concentration, typing mistakes, distractions, and accumulated fatigue, the variety in the typing style of each participant is revealed among a large number of repetitions of each sentence. Therefore, this dataset deals with classifying a person under intra-sentence varying typing style, *i.e.*, changing motion patterns of the same sentence, and inter-person changing level of typing consistency. Additionally, this dataset can suggest a scenario in which a model learns on one sentence and needs to infer to another sentence it never saw during training.

*60Typing10* dataset consists of 1800 videos of 60 participants. Each participant typed ten different sentences, while each sentence was repeated three times. Unlike *80Typing2*, *60Typing10* setting's purpose is simulating a scenario where a large number of different sentences, as well as a small number of repetitions from each sentence, are provided. The large abundance of different sentences, *i.e.*, different motion patterns, reveals each participant's unique typing style. At the same time, the small amount of repetitions supports each participant's variance in the typing style. Therefore, this dataset deals with the classification of a person under inter-sentence varying motion patterns, and for the model to generalize well to sentences it never saw during training, it must learn to classify each person by his unique typing style, *i.e.*, learn to classify the different people according to their unique typing style.

### 5.2. User classification on unseen sentences

In this experiment, we simulate a test case of continuous user identification by testing our model's ability to infer on unseen sentences, *i.e.*, different motion patterns. We split our data by sentence type and let the model train on a certain set of sentences while testing performed on a different set of sentences which the model never saw during training, *i.e.*, different types of sentences the user typed. Therefore, to perform well, the model must learn the unique motion style of each person.

The experiment performed on *60Typing10* as follows, we split our data in three ways, wherein each split a different number of sentences is given for training. We randomly split our data by sentences to train, validation, and test sets according to the split settings. We applied the same division to all other models for legitimate comparison. For *80Typing2*, we randomized the train sentence, and the other sentence divided between validation and test where two repetitions were used for validation and eight for test.

Results for this experiment on *60Typing10* and *80Typing2* appears in table 1 and 2, respectively. Our model out-

| Model | Acc(%) on unseen | Acc(%) on seen |
|---|---|---|
| HCN [18] | 94.18 | 99.66 |
| STGCN [35] | 93.59 | 99.64 |
| 3sARGCN [27] | 91.08 | 99.44 |
| PBGCN [29] | 95.98 | 99.84 |
| 2sAGCN [25] | 96.88 | 99.85 |
| StyleNet | **99.57** | **99.98** |

Table 2. *80Typing2* test accuracy of user classification on **unseen** and **seen** sentences.

performs all other compared models by an increasing margin as less training sentences are provided, which indicates our model's superior generalization ability.

### 5.3. User classification on seen sentences

In this experiment, we simulate a test case of user identification (access control by password sentence) by testing our model's ability to infer the same movement patterns, *i.e.*, sentences, he saw during training and other repetitions of these patterns. We use a large number of sentence repetitions to test the robustness to the variance in the typing style by simulating a scenario where a small amount of different motion patterns, *i.e.*, sentence type, is given along with a substantial variance in these patterns resulting from a large number of repetitions.

This experiment is performed by dividing *80Typing2*'s ten repetitions of each sentence as follows: five for train, one for validation, and four for test. We trained each model on the train set and tested its accuracy on the seen sentences but unseen repetitions.

According to the experiment's results, which appears in table 2, it is clear that this specific task is not complex and can be addressed by other methods. However, it proves that our models' extra complexity does not harm the performance in the simpler "password sentence" use cases.

### 5.4. Noisy data

The skeleton-based approach is dependent on a reliable joints detector that extracts the joint's location from each input frame. To challenge our model, we experimented with a scenario similar to 5.2 (the more challenging task simulating continuous user identification), where during inference, the joints detector is randomly failing and providing noisy data, *i.e.*, incorrect joints location.

We trained all models as usual, while during test time, we randomly zeroed $(x, y, score)$ data of a joint. The amount of joints that zeroed is drawn uniformly among [0,1,2], while the decision of which joint values to zero is random, but weighted by each joint tendency to be occluded, *e.g.*, the tip of the thumb's joint has a higher probability of being drawn than any of the ring fingers which tend less to be occluded while typing.

| Model | [4,2,4] Acc(%) | [3,2,5] Acc(%) | [2,2,6] Acc(%) |
|---|---|---|---|
| HCN [18] | 57.87 | 53.46 | 45.06 |
| STGCN [35] | 70.03 | 68.3 | 60.61 |
| 3sARGCN [27] | 71.36 | 69.35 | 67.92 |
| PBGCN [29] | 83.96 | 82.75 | 80.4 |
| 2sAGCN [25] | 73.33 | 71.34 | 68.83 |
| StyleNet | **91.79** | **87.57** | **85.24** |

Table 3. Test accuracy for noisy data experiment on *60Typing10*. Training conducted as usual, but during test time, we randomly zeroed joint $(x, y, score)$ to simulate a situation where the data is noisy or some joint's location is missing. $[\alpha, \beta, \gamma]$ denotes number of sentences given for train, validation, and test, respectively

According to the experiment's results in table 3, our model is much more robust to noisy data. The non-local approach helps the model rely less on a particular joint and provides a more global analysis of the typing style, which increases the model's robustness in cases of noisy data.

### 5.5. Uncontrolled environment

In this experiment, we compared our method with VAC RGB-based methods in an uncontrolled environment scenario. Even though RGB based methods perform well in a controlled environment, their performance tends to decrease severely under alternating scene properties such as lighting and noise. Even though data augmentation can increase these methods robustness to challenging environmental conditions, it is impossible to simulate all possible scenarios. Therefore, using an RGB-based approach in real-world scenarios tends to fail in the wild. Therefore, we explored our method's robustness under challenging environmental conditions to verify the skeleton-based approach superiority in the task of *typing style learning for person identification*.

We performed this experiment in a similar manner to 5.2, but with some differences. We trained each model using data augmentation techniques such as scaling, lighting, and noise. Later, during test time, we applied different data augmentations, *e.g.*, different lighting, and noise models, than those used during training on the input videos.

Results for this experiment appear in table 4. While all the compared methods achieved a high accuracy rate under a controlled environment, their accuracy rate dropped in an uncontrolled environment scenario. Our method's performance did not change except for a slight decline of less than 0.5% in its accuracy rate. It is much easier to train a joint detector to operate in an uncontrolled environment since it locates the joints by the input image and the hand context altogether. Unlike the image appearance, the hand context is not dependent on the environment. Therefore, the joints localizer can better maintain its performance under varying conditions, making our pipeline resilient to this scenario.

| Model | [4,2,4] Acc(%) | | [3,2,5] Acc(%) | | [2,2,6] Acc(%) | |
|---|---|---|---|---|---|---|
| | Controlled env. | Uncontrolled env. | Controlled env. | Uncontrolled env. | Controlled env. | Uncontrolled env. |
| I3D [3] | 99.68 | 63.12 | 99.75 | 59.16 | **99.7** | 62.30 |
| T3D [6] | 98.85 | 56.89 | 99.01 | 54.67 | 98.64 | 54.06 |
| StyleNet | **99.84** | **99.59** | **99.77** | **99.57** | 99.5 | **99.17** |

Table 4. Test accuracy for uncontrolled environment experiment on *60Typing10*. RGB models trained with data augmentation while during test time, a different set of augmentations applied. $[\alpha, \beta, \gamma]$ denotes the number of sentences for train, validation, and test, respectively. env. denotes environment

| Model | [5,10,35] Acc(%) | [10,10,30] Acc(%) | [15,10,25] Acc(%) | [20,10,20] Acc(%) | [25,10,15] Acc(%) |
|---|---|---|---|---|---|
| HCN [18] | 92.46 | 96.27 | 97.3 | 98.32 | 98.82 |
| STGCN [35] | 95.72 | 97.92 | 98.24 | 98.79 | 98.96 |
| 3sARGCN [27] | 94.7 | 97.76 | 98.08 | 98.56 | 98.89 |
| PBGCN [29] | 98.51 | 99.07 | 99.48 | **99.61** | 99.7 |
| 2sAGCN [25] | 97.75 | 98.33 | 98.73 | 98.96 | 99.01 |
| StyleNet | **99.46** | **99.48** | **99.51** | 99.58 | **99.79** |

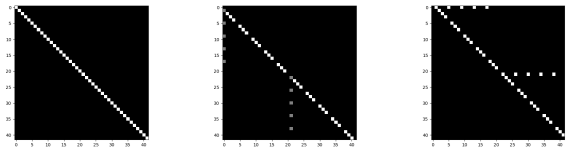
Table 5. Test accuracy of user classification on unseen sentences on *How We Type* using 3D input data. $[\alpha, \beta, \gamma]$ denotes the number of sentences for train, validation and test, respectively

| Model | [5,10,35] Acc(%) | [10,10,30] Acc(%) | [15,10,25] Acc(%) | [20,10,20] Acc(%) | [25,10,15] Acc(%) |
|---|---|---|---|---|---|
| StyleNet 2D | 99.41 | 99.47 | **99.54** | **99.59** | 99.78 |
| StyleNet 3D | **99.46** | **99.48** | 99.51 | 99.58 | **99.79** |

Table 6. Test accuracy of user classification on unseen sentences on *60Typing10* when using 3D or 2D input data. $[\alpha, \beta, \gamma]$ denotes the number of sentences for train, validation, and test, respectively.

### 5.6. 2D Vs. 3D data

In this experiment, we evaluate our model using 3D input data, and explore the trade-off between 3D and 2D input data. We used How We Type dataset [8] that contains 3D coordinates of 52 joints from both hands and a total of 30 different persons, where each person typed 50 sentences. Overall, we tested five different splits of the data, where each split contains a different number of training sentences. We randomly divided the data between training, validation, and test in a similar manner to 5.2 according to the partitioning setting of each split. We repeated this scheme several times for an accurate assessment of the model's performance.



(a) 1st subset (b) 2nd subset (c) 3rd subset

Figure 5. Adjacency matrices of two hands. Each matrix is built by diagonally concatenating two replicas of its one-hand version.

We used 21 out of 26 joints for each hand for consistency with all other experiments and followed [35] partition strategy, which was mentioned in the paper. Figure 5 contains the adjusted adjacency matrix that enables our model to learn the unique dependencies between the joint of both hands. When we tested our model with 3D coordinates as input, $z$ axis data replaced the $score$ input. Therefore, each frame data consist of 42 $(x, y, z)$ coordinates of joints from both hands.

The results for this experiment appear in table 5, where we can see that even though our model trained on only 10% of the entire data, it achieved a high accuracy rate and outperformed all other models. Results for the trade-off between 2D and 3D input data appear in table 6. According to the results, we can see that our model achieves similar performance when provided either with 2D or 3D input data. Unlike other tasks where the model benefits from the 3rd dimension, it seems unneeded in this task.

## 6. Conclusions

We introduced *StyleNet*, a novel new architecture for skeleton-based typing style person identification. Motivated by [33], we redesigned the spatial-temporal relationships allowing for a better longitudinal understanding of actions. *StyleNet* evaluated on the newly presented *80Typing2* and *60Typing10* datasets and outperformed all compared skeleton-based action classification models by a large margin when tested in the presence of noisy data and outperformed when tested under controlled conditions.

## References


[1] Roberto Brunelli and Tomaso Poggio. Face recognition: Features versus templates. *IEEE transactions on pattern analysis and machine intelligence*, 15(10):1042–1052, 1993.

[2] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. In *arXiv preprint arXiv:1812.08008*, 2018.

[3] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[4] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[5] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in neural information processing systems*, pages 3844–3852, 2016.

[6] Ali Diba, Mohsen Fayyaz, Vivek Sharma, Amir Hossein Karami, Mohammad Mahdi Arzani, Rahman Yousefzadeh, and Luc Van Gool. Temporal 3d convnets: New architecture and transfer learning for video classification. *arXiv preprint arXiv:1711.08200*, 2017.

[7] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1110–1118, 2015.

[8] Anna Maria Feit, Daryl Weir, and Antti Oulasvirta. How we type: Movement strategies and performance in everyday typing. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4262–4273, New York, NY, USA, 2016. ACM.

[9] Chelsea Finn, Ian Goodfellow, and Sergey Levine. Unsupervised learning for physical interaction through video prediction. In *Advances in neural information processing systems*, pages 64–72, 2016.

[10] Simon Fong, Yan Zhuang, and Iztok Fister. A biometric authentication model using hand gesture images. *Biomedical engineering online*, 12(1):111, 2013.

[11] DK Isenor and Safwat G Zaky. Fingerprint identification using graph matching. *Pattern Recognition*, 19(2):113–122, 1986.

[12] Arpit Jain, Abhinav Gupta, Mikel Rodriguez, and Larry S Davis. Representing videos using mid-level discriminative patches. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2571–2578, 2013.

[13] Herve Jegou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *IEEE transactions on pattern analysis and machine intelligence*, 34(9):1704–1716, 2011.

[14] Rick Joyce and Gopal Gupta. Identity authentication based on keystroke latencies. *Communications of the ACM*, 33(2):168–176, 1990.

[15] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017.

[16] Okan Kopuklu, Neslihan Kose, and Gerhard Rigoll. Motion fused frames: Data level fusion strategy for hand gesture recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 2103–2111, 2018.

[17] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[18] Chao Li, Qiaoyong Zhong, Di Xie, and Shiliang Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. *arXiv preprint arXiv:1804.06055*, 2018.

[19] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. In *Thirty-second AAAI conference on artificial intelligence*, 2018.

[20] Doug Mahar, Renee Napier, Michael Wagner, William Laverty, RD Henderson, and Michael Hiron. Optimizing digraph-latency based biometric typist verification systems: inter and intra typist differences in digraph latency distributions. *International journal of human-computer studies*, 43(4):579–592, 1995.

[21] Fabian Monrose and Aviel Rubin. Authentication via keystroke dynamics. In *Proceedings of the 4th ACM conference on Computer and communications security*, pages 48–56, 1997.

[22] Michalis Raptis, Iasonas Kokkinos, and Stefano Soatto. Discovering discriminative action parts from mid-level video representations. In *2012 IEEE conference on computer vision and pattern recognition*, pages 1242–1249. IEEE, 2012.

[23] Joseph Roth, Xiaoming Liu, and Dimitris Metaxas. On continuous user authentication via typing behavior. *IEEE Transactions on Image Processing*, 23(10):4611–4624, 2014.

[24] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[25] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12026–12035, 2019.

[26] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014.

[27] Yi-Fan Song, Zhang Zhang, and Liang Wang. Richly activated graph convolutional network for action recognition with incomplete skeletons. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 1–5. IEEE, 2019.

[28] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6479–6488, 2018.

[29] Kalpit Thakkar and PJ Narayanan. Part-based graph convolutional network for action recognition. *arXiv preprint arXiv:1809.04983*, 2018.

[30] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 588–595, 2014.

[31] He Wang, Feixiang He, Zexi Peng, Yongliang Yang, Tianjia Shao, Kun Zhou, and David Hogg. Smart: Skeletal motion action recognition attack. *arXiv preprint arXiv:1911.07107*, 2019.

[32] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action recognition by dense trajectories. In *CVPR 2011*, pages 3169–3176. IEEE, 2011.

[33] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

[34] Shuangjie Xu, Yu Cheng, Kang Gu, Yang Yang, Shiyu Chang, and Pan Zhou. Jointly attentive spatial-temporal pooling networks for video-based person re-identification. In *Proceedings of the IEEE international conference on computer vision*, pages 4733–4742, 2017.

[35] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Thirty-second AAAI conference on artificial intelligence*, 2018.