TouchEnc: a Novel Behavioural Encoding Technique to Enable Computer Vision for Continuous Smartphone User Authentication

1st Peter Aaby, 3rd William J Buchanan, 4th Zhiyuan Tan
School of Computing, Engineering & The Built Environment
Edinburgh Napier University
Edinburgh, United Kingdom

2nd Mario Valerio Giuffrida
School of Computer Science
University of Nottingham
Nottingham, United Kingdom

***Abstract*—We are increasingly required to prove our identity when using smartphones through explicit authentication processes such as passwords or physiological biometrics, e.g., authorising online banking transactions or unlocking smartphones. However, these methods are often annoying to input and do not guarantee that the genuine user remains the same. Thus, a modern verification process should differ from traditional authentication. In touch-based biometrics, a new approach must not verify what we draw but *how* we draw it. Our research proposes TouchEnc, a Deep Learning approach that outperforms conventional methods. Unlike Machine Learning methods, TouchEnc automates the feature extraction from touch gestures. TouchEnc achieves this by transforming and encoding touch behaviour into images, enabling continuous authentication through modern computer vision. Our approach has been tested on a popular and publicly available dataset to demonstrate its effectiveness. Results show that users can authenticate using TouchEnc with a single gesture containing users' on-screen navigational behaviour, independent of drawing up, down, left, or right. TouchEnc achieves an 8.4% Equal Error Rate and a 96.7% Area Under the Curve using a single gesture. Furthermore, TouchEnc achieves up to 65% better Equal Error Rates when combining gestures compared to the related work.**

***Index Terms*—Behavioural Biometrics, Continuous Authentication, Computer Vision, Deep Learning**

I. Introduction

From 2022 to 2027, identity theft and fraudulent banking transactions are projected to increase, with costs to merchants exceeding $343 billion [1]. Widely popularised approaches such as multi-factor authentication provide the opportunity to increase the protection of user accounts but are often inconvenient [2], [3]. However, the FIDO Alliance recently proposed a passwordless approach, where users can replace passwords with an internal or external authenticator, such as mobile tokens [4]. Mobile tokens could be an Android smartphone with embedded biometric authentication or other applications and lock screen protection.

In this work, we propose TouchEnc as a passive, implicit and Continuous Authentication (CA) mechanism on mobile tokens that can automatically extract personal gestures from finger movement recorded on touchscreens beyond the point of entry. Thus, CA captures and verifies behavioural biometrics and ensures user authenticity over time. While other sensors are available, such as accelerometer and gyroscopic movement [5], this work presents a method to authenticate users exclusively by behaviour extracted from *on-screen* gestures to allow comparison with other results using the same dataset. We achieve state-of-the-art performance by encoding touchscreen records from a public dataset [6] into images and cropping the essential screen area for automatic feature extraction. An example of a single gesture and corresponding important screen area can be seen in Figure 1, for a user drawing a downwards-moving gesture containing several touchpoints. Our approach utilises the information captured in touchpoints to encode behaviour into images using the Red, Green, and Blue colour channels. Thus, each gesture becomes an image suitable for classification using computer vision and DL.

A. Motivation and contributions

CA has seen increasing interest from the research community looking to harness information from sensors such as accelerometers, gyroscopes, and location, among others [5], [7], to alleviate the frustration of smartphone users authenticating on mobile devices. However, touch-based CA still suffers due to several factors and challenges, including (i) adequately engineered features [5], (ii) personally selected features [8], and (iii) faster detection, e.g., not relying on multiple gestures [7], [9].

To overcome these challenges, we contribute the following:

1) Proposing the 'TouchEnc' image encoding approach and removing the need for manual feature engineering or selection.
2) Defining and empirically testing six image plotting styles for the encodings.
3) Improved performance over the related work independent of the drawn direction and achieving fast detection based on a single model and gesture.

The paper is structured as follows. Section II presents the related work and baseline performances using the same dataset. Section III describes the TouchEnc approach, and Section IV demonstrates the implementation. Section V presents the results. A summary of limitations and future work appears in Section VI before concluding the work in Section VII.

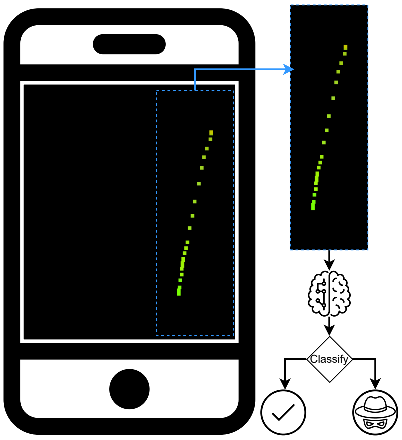


Fig. 1. An overview of the automatic extraction of visual touch behaviour for continuous user authentication, where each touchpoint encodes pressure, displacement, and acceleration

II. Related work

In 2021, Frank et al. [10] demonstrated that touchscreen inputs can be used for CA. Soon after, Serwadda et al. [6] published a larger dataset and investigated the best classifiers through a different feature set, individually modelling vertical and horizontal gestures for each screen orientation. In [11], the authors defined a new gesture direction as oblique, which occurs when a gesture curves during a horizontal or vertical interaction. Like [6], [10], each model is trained according to the drawn direction, and analysis shows that the best performance is derived from oblique gestures. However, comparing these works remains challenging since they utilise different data, feature sets, and methods to aggregate gestures [12]. [12], [13] studies the differences in directional modelling using data from [6] and five common feature sets. They conclude that models can be trained as one, independent of the gesture direction.

Since this work uses data published by [6], we focus on and present a comparison of the performance achieved on this data in Table I. The focus ensures fairer comparisons with our work and avoids bias towards private data sets, which often perform better but are challenging to verify [14]. Table I also highlights the differences in the number of features used, the number of required gestures for accurate authentication, and whether results rely on multiple models for good performance. It is also noted that an increase in the number of users appears to cause a decline in performance, which is consistent with the findings by Frank et al. [10]. [12] further identify the 40 users are required to get meaningful results. Despite authors using the same data, it proves challenging to ascertain the number of users in other studies [9], [15]–[21] and their inclusion criteria.

The work presented here employs a unique approach to automatically extracting touch features using an image-based method. Despite not using the data provided by Serwadda in 2013 [6], the only three other papers utilising images for touch-based CA [9] are briefly summarised. However, neither of them explicitly applies DL. First, [22] proposed a Graphic Touch Gesture Feature (GTGF) to extract identity traits and classify users using a Support Vector Machine (SVM). Later, they extend and improve their work using a Statistical Feature Model [23]. More recently, [24] proposed and applied a modified Edge Orientation Histogram to extract ten features. Their features are then used to classify users using an SVM. To distinguish ourselves from these works, we employ three methods: (i) propose three scalar values as colour encodings, (ii) reduce computational requirements by cropping a natural section of the drawing canvas, and (iii) effectively apply computer vision and DL for automatic feature extraction and classification. Thus, our approach's simplicity and enhanced performance could make it attractive for researchers looking to approach touch-based CA from a DL perspective.

TABLE I
Overview of the related work. "Number of Features, NF" used. Performance reported using a "Single Model, SM". "Number of Gestures, NG" combined. "Accuracy, †". "Horizontal only, ‡"

Study	Data	Users	NF	SM	NG	EER%
[10]	[10]	41	28	2	1	13.00
[22]	[22]	30	Image	4	6	4.31
[6]	[6]	106	28	4	10	15.50
[23]	[22]	78	Image	6	6	4.70
[24]	[24]	25	Image	5	N/A	†80.27
[15]	[6]	N/A	5	8	10	18.50
[16]	[6]	N/A	5+28	8	10	6.98
[17]	[6]	N/A	112	4	4	7.86
[11]	[11]	45	4-16	1	9	†95.85
[18]	[6]	N/A	28	‡✓	1	22.50
[19]	[6]	N/A	33	4	10	24.16
[20]	[6]	N/A	33	4	33	15.04
[9]	[10]	N/A	125	N/A	1	21.00
[21]	[6]	N/A	28	2	10	16.48
[13]	[6]	35	28	✓	5	17.90

III. Proposed approach

While most related work focuses on feature engineering and extraction, we take a fundamentally different approach by converting raw touch data into graphical gestures. For each drawn gesture, a user will generate several touchpoints. Traditionally, these touchpoints are grouped per interaction and computed into features representing time, direction, speed, and force [6], [10], [11], [13], [25]. However, a fixed feature set may not work for all users since behaviour is personal [8]. Instead, we demonstrate how graphical gestures enable automatic feature extraction using computer vision to overcome the challenges of manual feature engineering. However, the question is *how* to represent drawn gestures as images and which neural network is better suited for automatic feature extraction. The following subsections describe the dataset, user selection, and how gestures are encoded into images.

A. Data selection and preparation

In this work, we utilise the public data set published by [6] since it contains both areas covered by the finger and pressure information for each touchpoint. Several other data sets are available, as presented in [9], but they do not qualify due to missing area, pressure values, or too few samples per user. The data from [6] is provided in two sessions, each separated by at least one day between data captures for each user. The data is also captured in landscape and portrait, but we focus exclusively on portrait since it is the most commonly used orientation [16]. Figure 2 presents the first ten users and the number of gestures recorded when drawing gestures in horizontal and vertical directions for their first session.

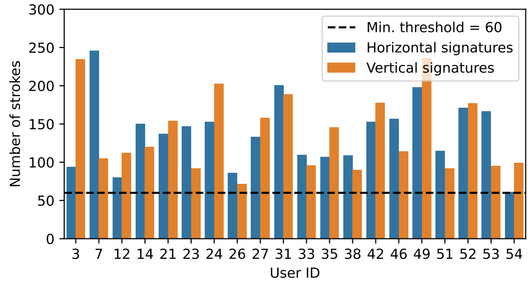


Fig. 2. The number of gestures the first ten users provided, given the direction of their drawn gestures. Any user above the minimum threshold is included.

Figure 2 illustrates the number of gestures per horizontal and vertical direction varies among users. This may be due to personal preference or subconscious behaviour. Some users navigate shorter and more frequently, while others move quickly and may draw longer gestures. Previous studies often fix the number of gestures per user when training classifiers [8], [11], [24]. Since related work finds 120 samples are required to perform well [11], we set a minimum threshold of 60 gestures in each horizontal and vertical direction and instead argue for this minimum threshold per user and use all their data. The criteria are applied for both sessions to allow data subsetting without affecting the minimum required number of gestures. Like [6], [21], our user selection protocol fairly considers and includes *any* user within the scope; thus, 74 of 106 users provide enough data.

We subset the data into training, validation, and testing to ensure no leakage between training and evaluation. The splits are grouped per user, session, and direction to respect the underlying distribution described in Figure 2. For each user, the last 20 gestures in each group are selected for testing, the previous 20 for validation, and the remaining for training. Thus, the validation and testing sets are balanced. Qualifying gestures must also have at least five touchpoints; otherwise, it is discarded as a click action [6], [8]–[10].

B. Data cleaning and cropping the gesture canvas

Directional variations happen when users draw gestures on their device screen, e.g., swerving when scrolling down rather than drawing a straight line. However, if a user changes their mind halfway through an interaction, a gesture may become invalid since it deviates significantly from the intended direction. Thus, gestures where the average moving angle between five touchpoints' differs by more than 90 degrees are removed. Following data cleaning, a blank canvas with the maximum screen resolution is generated to accommodate drawing the gesture.

Specific DL architectures require significant memory when dealing with high-resolution images. Downsizing image resolution may seem like a solution, but it can result in signal loss. Instead, we propose cropping out the gesture from the canvas as shown in Figure 1 and removing the empty parts of the canvas. We analyse the entire dataset to determine the maximum screen area used in gestures and which cropping resolution captures the most gestures. While cropping the gesture may remove important location information, we address how to mitigate this issue in Section III-C.

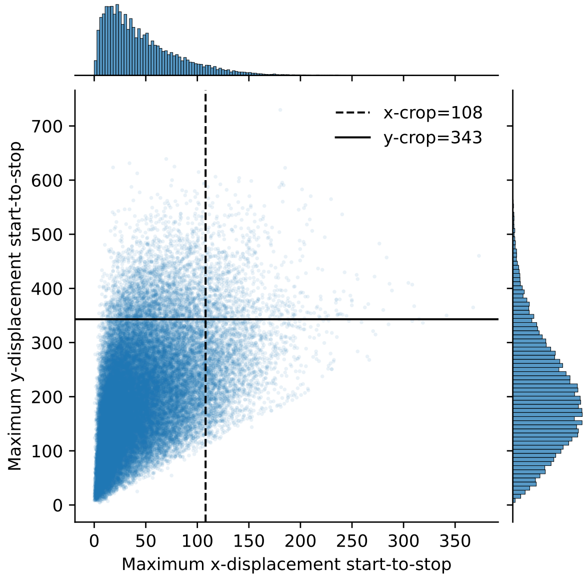


Fig. 3. Scatter point for each gesture across all users, describing the maximum touch displacement on the x and y axis. Histograms show the axis-specific distribution, and the legend signifies the 90^{th} percentile used for cropping.

Figure 3 shows the distribution of maximum vertical gestures and the dimensions for x and y cropping. We perform outlier analysis on horizontal and vertical gestures but exclude the horizontal figure for conciseness since the distributions show a similar trend. To define the cropping dimension, we use the 90^{th} percentile on each axis, which helps to eliminate outliers where users' gestures swerve excessively. For horizontal gestures, the y-crop becomes the x-crop since the orientation and longest axis are swapped, and vice versa for the shorter axis. The outlier removal causes a minor data loss, resulting in 5,535 out of 31,432 gestures being dropped for horizontal and 7,333 out of 42,473 for vertical. Since image classification often requires the same image dimension, the horizontal gesture is rotated 90 degrees counterclockwise.

C. Biometric colour encodings

The raw touch data, including the x, y-coordinates, associated *pressure*, *area*, and *timestamps*, are obtained from [6]. While this information has traditionally been used to engineer features manually, we propose transforming these

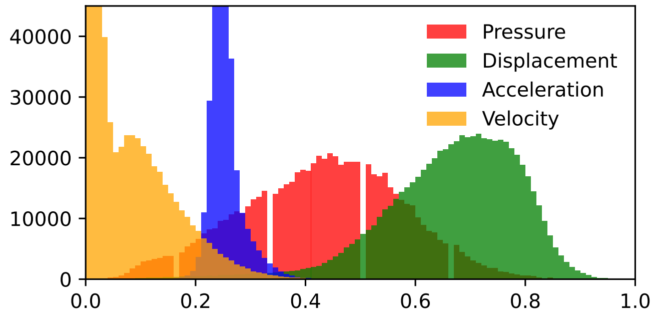


Fig. 4. Distribution of the proposed colour encodings.

raw data into images using our TouchEnc encodings.Using the empty canvas, a square box represents each touchpoint using the x, y coordinates. This box is then scaled according to the raw *area occluded by the finger* and coloured Red, Green, and Blue (RGB) in the range 0-1 according to the chosen encoding. Red encodes *pressure*, where zero means no pressure, and one is the maximum possible. The original canvas location is encoded as the displacement using Equation (1) from the screen origin (0,0) to mitigate any loss from cropping the image. The displacement is then coloured green, where zero is close to the screen origin, and one is furthest away. Finally, time information is encoded as *acceleration* or *velocity* between touchpoints using formulas Equations (2) and (3). Although *acceleration* and *velocity* may rely on the same time component, the distribution differs Figure 4. Thus, the following section investigates the difference between two RGB combinations. The first contains Pressure, Displacement, and Acceleration (PDA), and the second contains Pressure, Displacement, and Velocity (PDV).

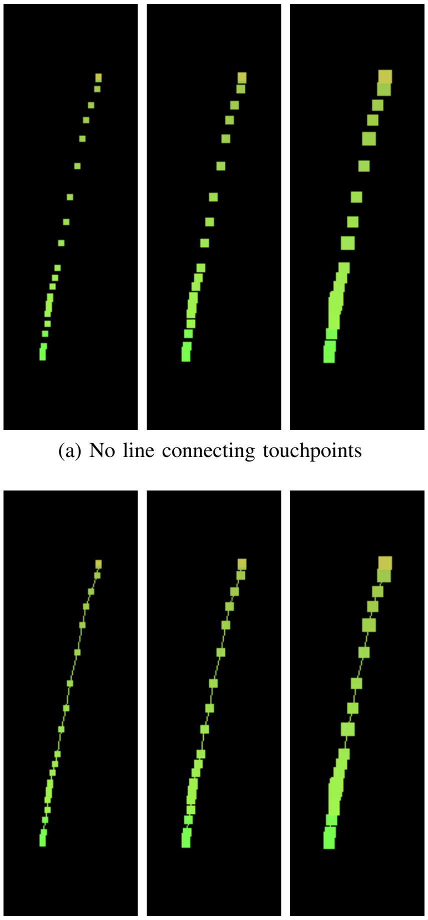
$$\text{displacement} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{1}$$

$$\text{velocity} = \frac{\Delta\text{displacement}}{\Delta\text{time}} \tag{2}$$

$$\text{acceleration} = \frac{\Delta\text{velocity}}{\Delta\text{time}} \tag{3}$$

D. Plotting styles

Before assigning colours, the *occluded area of the screen* caused by the finger is considered to draw a square box proportionate to the area. However, the raw area data is reported as values between 0-1 that cannot directly be used to define the dimension of the square. Thus, the area is experimentally multiplied by 5, 10, or 15 and rounded. For example, if the phone reports an area of 0.2 scaled by 10, plot a square with 2x2 pixels and colour it according to the RGB touch encodings. Drawing connecting lines between touchpoints could also increase performance by extrapolating information between touchpoints. As such, we plot a variation for each area scale, with and without connecting lines, and train several image classifiers on the plotting styles. Interestingly, if the touchpoints are dense, a more significant scaling factor causes the squares to overlap, potentially losing unique gesture behaviour and occluding the connecting lines. This can be observed in Figure 5b.



(a) No line connecting touchpoints

(b) With line connecting touchpoints

Fig. 5. Example of the same gesture plotted using different variations of area scaling and line styling

IV. Implementation

To examine the effectiveness in extracting suitable features from TouchEnc encodings, six different image plotting variations illustrated in Figure 5 are tested. Image classifiers are implemented using the PyTorch DL framework [26], which offers a range of well-researched neural network architectures. Given the focus on mobile devices, we opted for classifiers designed explicitly for lower computational resources, such as the MobileNetV3 (MNV3) [27] with 1,5mill parameters and a larger EfficientNetB0 (ENB0) [28] with 4,1mill parameters. We chose the minor variant for each architecture to conserve training time. The loss function for all models uses cross-entropy and is optimised with AdamW [29].

To ensure effective training, a modern training recipe is inspired by [30]–[32], which includes learning rate annealing after a short linear learning rate warm-up. This concept initially helps speed up convergence and mitigates significant weight updates as the learning rate increases, while annealing combats issues where the optimiser may get stuck at a specific learning rate. Additionally, to prevent overfitting, Label Smoothing [33], Weight Decay [29], and Random Erasing [34] techniques are applied. Classification output utilises the Softmax activation function to produce class probabilities. One vs. Rest classification is used when computing the AUC scores for each user [6], [10], [35]. The macro-averaged AUC score is tracked against the validation set for a maximum of 50 epochs during training. Early stopping restores the best checkpoint if validation performance decays for ten consecutive epochs.

Hyperparameter search is implemented to determine the optimal touch encodings, plotting styles, and model settings. The best MNV3 encodings are also tested using an ENB0 to understand whether a more complex model can further improve performance. Table II outlines a shared parameter grid used in the parameter search. The best parameters for any model are chosen based on the highest macro-averaged AUC score when evaluating the validation set during training. Optimising for better AUC scores is effective since it improves overall performance independent of the classification decision threshold [36]. Consequently, we compute 72 MNV3 and 36 ENB0 models due to the search space. The following section presents the best five models for each grid search and later combines gestures to compare against Table I.

TABLE II
HYPERPARAMETER USED IN THE GRID SEARCH.

Parameter	Search space
Area Scale (AS)	5, 10, 15
Line Style (LS)	with (✓), without lines (✗)
Learning Rate (LR)	1e-2, 1e-3, 1e-4
Linear Warm-up	5 epochs
Cosine Annealing [32]	Maximum Epochs
Weight Decay [29]	0.05
Label Smoothing [33]	0.1
Random Erase [34]	0.25
Pre-trained Weights [37]	False
Batch Size (BS)	32, 64
Maximum Epochs	50

V. EVALUATION AND RESULTS

Section III-C describes the PDA and PDV encodings used to train MNV3 models. Table III displays the top ten results, revealing that *acceleration* outperforms *velocity* as encoding in combination with *pressure* and *displacement*. Thus, PDA is selected as the superior encoding method. TouchEnc also performs well in single-gesture authentication, with a 23% increase compared to the best single-gesture result of 13% [10]. This improvement is measured over 74 users, which is 33 more users than evaluated in [10]. Table III also presents the training time, which may be interesting when deciding between the results in the following section, where the ENB0 results are presented.

TABLE III
TOP FIVE MNV3 MODELS COMPARING PDA TO PDV. "LEARNING RATE, LR", "AREA SCALE, AS", "BATCH SIZE, BS", "TRAINING TIME, TT".

Enc	LS	AS	LR	BS	EER(%)	AUC(%)	TT (sec)
PDA	✗	15	0.001	64	10.36	95.33	2175
PDA	✗	15	0.001	32	10.40	95.35	5852
PDA	✗	10	0.001	32	10.43	95.01	5655
PDA	✓	15	0.001	64	10.66	95.24	2215
PDA	✓	10	0.001	32	10.73	95.14	5328

A. EfficientNet improvements

While the MNV3 performs well, the optimal encodings may improve performance in tandem with larger and more complex models such as ENB0. Table IV presents the top five results when training an ENB0 model with the TouchEnc PDA encodings. The results show that the best models converge at parameters such as the Batch Size (BS) and Learning Rate (LR), with stable performance independent of the plotting style. We highlight these results are based on verifying users by analysing gestures individually. Hence, the results are conservative since most related works offer their performance by aggregating gestures. While the performance has increased from the MNV3, so has the time to model. This is a natural trade-off between complexity and performance, which could be interesting to study further. Regardless, the best ENB0 model further increases the TouchEnc performance compared to Table III with a 43% improvement over [10] when authenticating individual gestures.

TABLE IV
TOP FIVE ENB0 USING PDA ENC. "LEARNING RATE, LR", "AREA SCALE, AS", "BATCH SIZE, BS", "TRAINING TIME, TT".

Enc	LS	AS	LR	BS	EER(%)	AUC(%)	TT (sec)
PDA	✗	15	0.01	32	8.42	96.69	10666
PDA	✓	10	0.01	32	8.60	96.60	10534
PDA	✗	5	0.01	32	8.91	96.39	10599
PDA	✗	15	0.01	64	8.92	96.42	7156
PDA	✓	15	0.001	32	9.08	96.26	7035

Each model is reported with the corresponding AUC score as part of the results. The values are related to the Receiver Operation Characteristic (ROC), which explains the model performance as a function of different thresholds. Thus, Figure 6 visualises the ROC curve for the best ENB0 model and compares the validation and testing results for the model. A concern could emerge if the curves are significantly different with indications of over or under-fitting. Judging by the plot, the ENB0 model generalises well to the unseen testing data. However, the standard deviation suggests that certain users are more easily classified than others. Additionally, users have the ability to prioritise reducing false positives or false negatives, but doing so may come at a cost in user experience, such as being mistakenly granted or denied access.

B. The 'best' plotting variant

The best-performing model is an ENB0 since it trains reasonably fast and outperforms the MNV3. However, note that any of the six plotting variations presented in Figure 5 are applicable, although some perform better than others in

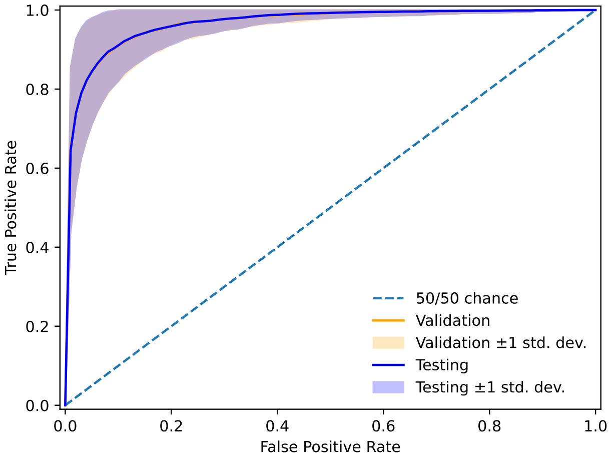


Fig. 6. ROC curve showing the best performing Efficient Net according to the lowest EER score and difference between validation and testing performance.

specific contexts. For example, while the MNV3 are generally faster and cheaper to train, they are also more sensitive to lower AS, with a preference for scaling 10-15 times and benefiting from connecting the touchpoints. On the contrary, the ENB0 perform well in almost any plotting style but appears to converge faster with larger AS and not connecting touchpoints. These and more observations can be inspected in [38], which contains all runs and results. Despite the difference in performance, single-gesture authentication can be insufficient for some users, and the next section, therefore, presents the performance when combining n Number of Gestures (NG)

C. Single vs. multi-gesture authentication

The best model architecture can be defined in several ways, e.g., by the highest AUC, accuracy score, or lowest EER score. The AUC score provides a model that performs well regardless of thresholds, and the highest AUC score is achieved using the ENB0. However, the evaluation has focused on single-gesture authentication while most related work combines gestures, as seen in Table I. As such, we implement an average moving window of 2 gestures, then 5, 9, and 10, to illuminate how well TouchEnc compares against the state-of-the-art performance of the related work. The results are shown in Table V where NS is the Number of Signatures aggregated. In this work, we aggregate using moving average windows over the predicted probabilities, similar to others [8], [10]. When $NS = 1$, no gestures are aggregated, such as in Figure 6. Generally, a model with good single gesture performance is also expected to perform well when combining gestures, and this behaviour is visually presented in Figure 7. The figure shows that our best ENB0 model and our automatic feature extraction approach are superior to the work of others. In the case of combining five gestures, we compare our results to [17] in Table I since they achieve good performance on the same data without combining too many gestures. We achieve 4% EER compared to [17], which reaches 7.86% EER. That is a 65% improvement, with diminishing gains when aggregating more gestures. [13] found similar diminishing returns but needed more gestures before the performance converged.

TABLE V
PERFORMANCE WHEN COMBINING n GESTURES. ALL VALUES ARE REPORTED AS "MEAN / MEDIAN (STD)"

n	EER(%)	AUC (%)	Accuracy (%)
1	8.4 / 8.1 (.049)	96.7 / 97.4 (.0306)	91.6 / 92.1 (.049)
2	5.7 / 5.0 (.040)	98.2 / 98.9 (.0201)	94.4 / 95.1 (.039)
3	4.6 / 3.8 (.037)	98.7 / 99.5 (.0168)	95.5 / 96.7 (.037)
4	4.0 / 3.1 (.033)	98.9 / 99.6 (.0149)	96.1 / 97.3 (.034)
5	3.6 / 2.5 (.032)	99.1 / 99.7 (.0134)	96.5 / 97.6 (.031)
6	3.2 / 2.4 (.029)	99.2 / 99.8 (.0119)	96.9 / 97.9 (.029)
7	3.2 / 2.3 (.029)	99.3 / 99.8 (.0110)	97.0 / 97.8 (.029)
8	3.1 / 2.1 (.028)	99.3 / 99.8 (.0102)	97.1 / 98.0 (.028)
9	3.1 / 2.3 (.027)	99.4 / 99.8 (.0095)	97.1 / 98.1 (.027)
10	3.1 / 2.4 (.027)	99.4 / 99.8 (.0090)	97.1 / 98.0 (.026)

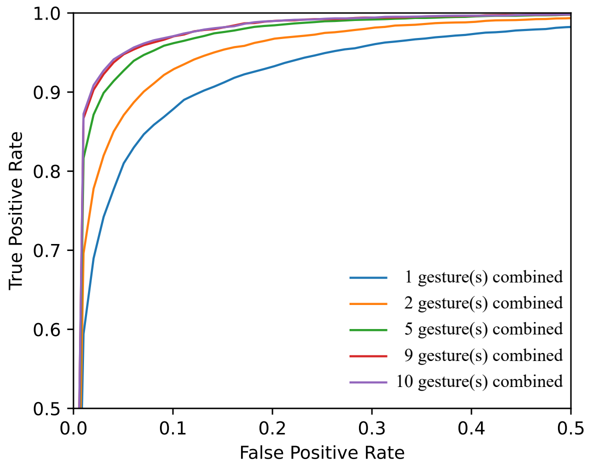


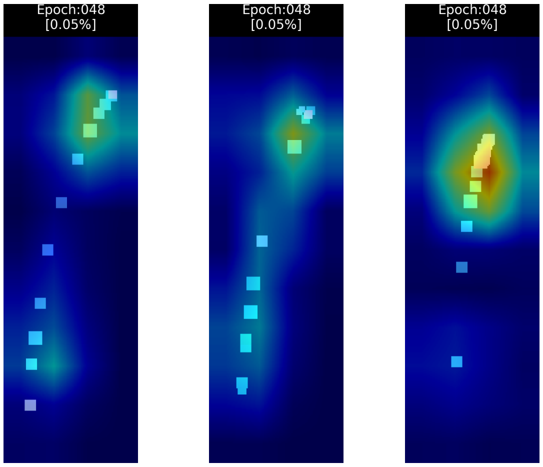
Fig. 7. Enlarged ROC plot showing the impact of combining n gestures. One gesture is similar to Figure 6.

D. Confirming the TouchEnc attention

Since ENB0 performs well and trains fast, we recommend and use the architecture to analyse and present Figure 8, which shows a GradCam [39] analysis of the activation maps for three upwards-moving gestures drawn by the same user, in sequence. Demonstrated by the brighter colours, the network has automatically given attention to the touchpoints along the trajectory. As with many DL models, explaining why a particular activation appears can be challenging. For example, it is peculiar to see Figure 8a appear to have skewed attention towards the right side of the first touchpoint. Still, a pattern can be observed relating to the gesture with more attention where the finger would have lifted from the screen.

VI. LIMITATIONS AND FUTURE WORK

This work is limited to proposing the encodings and verifying automatic feature extraction is possible using two off-the-shelf computer vision models. While these architectures are commonly used, larger, more complex models could yield better results. It would also be interesting to experiment



(a) Signature 1 (b) Signature 2 (c) Signature 3

Fig. 8. GradCam [39] visualisation of activation maps using the best performing Efficient Net for automatic feature extraction

further by designing custom architectures or applying other models, such as Swin Transformers [31] or ConvNeXt [30]. Thus, we recognise that the performance measures may be conservative, considering the original models are designed for image recognition.

A. Optimal encoding and transformation

While TouchEnc encodes the drawn gestures effectively, other encodings may improve the feature extraction further. In this work, we rule out *velocity* and replace it with *acceleration*, but different encodings may be better. Furthermore, the image dimensions are fixed, but other sizes could allow further improvements. Similar to the max permitted swerving defined in Section III-B, a minimum x and y displacement may be required. Adding additional channel depths could also improve TouchEnc by encoding accelerometer force into a fourth hyper-spectral colour channel.

B. Adversarial attacks

During testing, the objective is to authenticate each user individually and reject all others. Thus, the other class are effectively attackers trying to bypass the CA system. However, these attackers have been seen during training and are currently invalid for adversarial analysis [12]. Due to space limitations, we recognise this limitation and suggest excluding n attackers in future work.

C. Deep metric learning

For this paper, the feature outputs are optimised using Cross Entropy loss and evaluated using Softmax to demonstrate that automatic feature extraction is possible using our image transformation technique. Consequently, the probabilities are constrained for the learned users in our multi-class one-vs-rest scenario. However, such an approach is unrealistic for deployment, where gestures are available only for the valid owner of a device. Fortunately, deep metrics can also be mined from these images. Our next area of study is demonstrating the effectiveness of deep metric learning using our approach, which could enable one-class zero-shot learning of novel users.

VII. Conclusion

Touch-based CA typically requires manual feature engineering, extraction, and selection. However, this work demonstrates a new method to conveniently and passively authenticate users by automatically detecting *how* they draw using TouchEnc encodings. We have improved on the state-of-the-art performance by shifting from a traditional ML-based and propose converting touch gestures into images. We encode touch pressure, displacement, and acceleration into colour channels, enabling off-the-shelf models such as Efficient Nets to extract behavioural features automatically. We achieve 8.4% mean EER and 91.5% accuracy using a single gesture, while combining five gestures improves the mean EER to 3.6% and accuracy to 96.5%. Lastly, our approach opens the door to exploiting other benefits of computer vision, such as mining deep metrics and applying zero-shot learning.

References

[1] Juniper research, “Fighting Online Payment Fraud in 2022 & Beyond,” 2022. [Online]. Available: https://www.juniperresearch.com/whitepapers/fighting-online-payment-fraud-in-2022-beyond

[2] M. Harbach, E. von Zezschwitz, A. Fichtner, A. D. Luca, and M. Smith, “It’s a Hard Lock Life: A Field Study of Smartphone (Un)Locking Behavior and Risk Perception,” *SOUPS ’14: Proceedings of the Tenth Symposium On Usable Privacy and Security*, pp. 213–230, 2014, iSBN: 978-1-931971-13-3. [Online]. Available: https://www.usenix.org/conference/soups2014/proceedings/presentation/harbach

[3] A. Mahfouz, I. Muslukhov, and K. Beznosov, “Android users in the wild: Their authentication and usage behavior,” *Pervasive and Mobile Computing*, vol. 32, pp. 50–61, Oct. 2016, publisher: Elsevier.

[4] S. Ghorbani Lyastani, M. Schilling, M. Neumayr, M. Backes, and S. Bugiel, “Is FIDO2 the Kingslayer of User Authentication? A Comparative Usability Study of FIDO2 Passwordless Authentication,” in *2020 IEEE Symposium on Security and Privacy (SP)*, vol. 2020-May. IEEE, May 2020, pp. 268–285. [Online]. Available: https://ieeexplore.ieee.org/document/9152694/

[5] V. M. Patel, R. Chellappa, D. Chandra, and B. Barbello, “Continuous User Authentication on Mobile Devices: Recent progress and remaining challenges,” *IEEE Signal Processing Magazine*, vol. 33, no. 4, pp. 49–61, Jul. 2016.

[6] A. Serwadda, V. V. Phoha, and Z. Wang, “Which verifiers work?: A benchmark evaluation of touch-based authentication algorithms,” in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*. Arlington, VA, USA: IEEE, Sep. 2013, pp. 1–8.

[7] A. Z. Zaidi, C. Y. Chong, Z. Jin, R. Parthiban, and A. S. Sadiq, “Touch-based continuous mobile device authentication: State-of-the-art, challenges and opportunities,” *Journal of Network and Computer Applications*, vol. 191, p. 103162, Oct. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804521001740

[8] P. Aaby, M. Valerio Giuffrida, W. J. Buchanan, and Z. Tan, “Towards Continuous User Authentication Using Personalised Touch-Based Behaviour,” in *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*. Calgary, AB, Canada: IEEE, Aug. 2020, pp. 41–48. [Online]. Available: https://ieeexplore.ieee.org/document/9251130/

[9] M. Georgiev, S. Eberz, and I. Martinovic, “Techniques for Continuous Touch-Based Authentication,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer Science and Business Media Deutschland GmbH, 2022, vol. 13620 LNCS, pp. 409–431. [Online]. Available: https://link.springer.com/10.1007/978-3-031-21280-2_23

[10] M. Frank, R. Biedert, E. Ma, I. Martinovic, and D. Song, “Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication,” *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 136–148, Oct. 2012.

[11] Y. Yang, B. Guo, Z. Wang, M. Li, Z. Yu, and X. Zhou, “BehaveSense: Continuous authentication for security-sensitive mobile apps using behavioral biometrics,” *Ad Hoc Networks*, vol. 84, pp. 9–18, Mar. 2019.

[12] M. Georgiev, S. Eberz, H. Turner, G. Lovisotto, and I. Martinovic, “Common Evaluation Pitfalls in Touch-Based Authentication Systems,” in *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security*. New York, NY, USA: ACM, May 2022, pp. 1049–1063. [Online]. Available: https://dl.acm.org/doi/10.1145/3488932.3517388

[13] P. Aaby, M. V. Giuffrida, W. J. Buchanan, and Z. Tan, “An omnidirectional approach to touch-based continuous authentication,” *Computers & Security*, vol. 128, p. 103146, May 2023. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167404823000561

[14] P. Kałużny, “Touchscreen Behavioural Biometrics Authentication in Self-contained Mobile Applications Design,” in *Business Information Systems Workshops*, ser. Lecture Notes in Business Information Processing, W. Abramowicz and R. Corchuelo, Eds. Cham: Springer International Publishing, 2019, pp. 672–685.

[15] A. Pozo, J. Fierrez, M. Martinez-Diaz, J. Galbally, and A. Morales, “Exploring a statistical method for touchscreen swipe biometrics,” in *2017 International Carnahan Conference on Security Technology (ICCST)*, Oct. 2017, pp. 1–4, iSSN: 2153-0742.

[16] J. Fierrez, A. Pozo, M. Martinez-Diaz, J. Galbally, and A. Morales, “Benchmarking Touchscreen Biometrics for Mobile Authentication,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 11, pp. 2720–2733, Nov. 2018.

[17] S. Y. Ooi and A. B.-J. Teoh, “Touch-Stroke Dynamics Authentication Using Temporal Regression Forest,” *IEEE Signal Processing Letters*, vol. 26, no. 7, pp. 1001–1005, Jul. 2019, conference Name: IEEE Signal Processing Letters.

[18] S. Keykhaie and S. Pierre, “Mobile Match on Card Active Authentication Using Touchscreen Biometric,” *IEEE Transactions on Consumer Electronics*, vol. 66, no. 4, pp. 376–385, Nov. 2020, conference Name: IEEE Transactions on Consumer Electronics.

[19] N. Pokhriyal and V. Govindaraju, “Learning Discriminative Factorized Subspaces With Application to Touchscreen Biometrics,” *IEEE Access*, vol. 8, pp. 152 500–152 511, 2020, conference Name: IEEE Access.

[20] M. Santopietro, R. Vera-Rodriguez, R. Guest, A. Morales, and A. Acien, “Assessing the Quality of Swipe Interactions for Mobile Biometric Systems,” in *2020 IEEE International Joint Conference on Biometrics (IJCB)*, Sep. 2020, pp. 1–8, iSSN: 2474-9699.

[21] A. Z. Zaidi, C. Y. Chong, R. Parthiban, and A. S. Sadiq, “A framework of dynamic selection method for user classification in touch-based continuous mobile device authentication,” *Journal of Information Security and Applications*, vol. 67, p. 103217, Jun. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2214212622000928

[22] X. Zhao, T. Feng, and W. Shi, “Continuous mobile authentication using a novel Graphic Touch Gesture Feature,” in *2013 IEEE Sixth International Conference on Biometrics: Theory, Applications and Systems (BTAS)*, Sep. 2013, pp. 1–6.

[23] X. Zhao, T. Feng, W. Shi, and I. A. Kakadiaris, “Mobile User Authentication Using Statistical Touch Dynamics Images,” *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 11, pp. 1780–1789, Nov. 2014. [Online]. Available: https://ieeexplore.ieee.org/document/6882159

[24] J. Ahmad, M. Sajjad, Z. Jan, I. Mehmood, S. Rho, and S. W. Baik, “Analysis of interaction trace maps for active authentication on smart devices,” *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4069–4087, Feb. 2017, publisher: Springer New York LLC. [Online]. Available: http://link.springer.com/10.1007/s11042-016-3450-y

[25] Z. Syed, J. Helmick, S. Banerjee, and B. Cukic, “Touch gesture-based authentication on mobile devices: The effects of user posture, device size, configuration, and inter-session variability,” *Journal of Systems and Software*, vol. 149, pp. 158–173, Mar. 2019.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” *Advances in neural information processing systems*, Dec. 2019, arXiv: 1912.01703. [Online]. Available: http://arxiv.org/abs/1912.01703

[27] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, “Searching for MobileNetV3,” in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. Seoul, Korea (South): IEEE, Oct. 2019, pp. 1314–1324. [Online]. Available: https://ieeexplore.ieee.org/document/9008835/

[28] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *36th International Conference on Machine Learning, ICML 2019*, vol. 2019-June, pp. 10 691–10 700, May 2019, arXiv: 1905.11946 Publisher: International Machine Learning Society (IMLS) ISBN: 9781510886988. [Online]. Available: https://arxiv.org/abs/1905.11946v5

[29] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” *7th International Conference on Learning Representations, ICLR 2019*, Nov. 2017, arXiv: 1711.05101 Publisher: International Conference on Learning Representations, ICLR. [Online]. Available: https://arxiv.org/abs/1711.05101v3

[30] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Jun. 2022, pp. 11 966–11 976. [Online]. Available: https://ieeexplore.ieee.org/document/9879745/

[31] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, F. Wei, and B. Guo, “Swin Transformer V2: Scaling Up Capacity and Resolution,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Institute of Electrical and Electronics Engineers (IEEE), Nov. 2021, pp. 11 999–12 009, arXiv: 2111.09883. [Online]. Available: http://arxiv.org/abs/2111.09883

[32] I. Loshchilov and F. Hutter, “SGDR: Stochastic Gradient Descent with Warm Restarts,” in *International Conference on Learning Representations*, 2017. [Online]. Available: https://openreview.net/forum?id=Skq89Scxx

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-December, pp. 2818–2826, Dec. 2015, arXiv: 1512.00567 Publisher: IEEE Computer Society ISBN: 9781467388504. [Online]. Available: https://arxiv.org/abs/1512.00567v3

[34] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, “Random Erasing Data Augmentation,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 13 001–13 008, Apr. 2020. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/7000

[35] R. Rifkin and A. Klautau, “In Defense of One-Vs-All Classification,” *Journal of Machine Learning Research*, vol. 5, no. Jan, pp. 101–141, 2004. [Online]. Available: https://www.jmlr.org/papers/v5/rifkin04a.html

[36] Z. Wang and Y.-C. I. Chang, “Marker selection via maximizing the partial area under the ROC curve of linear risk scores,” *Biostatistics*, vol. 12, no. 2, pp. 369–385, Apr. 2011, publisher: Oxford Academic.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Jun. 2009, pp. 248–255. [Online]. Available: https://ieeexplore.ieee.org/document/5206848/

[38] Aaby, “TouchEnc: repository and runs,” Sep. 2023, language: en. [Online]. Available: https://zenodo.org/record/8332523

[39] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, vol. 2017-October. IEEE, Oct. 2017, pp. 618–626. [Online]. Available: http://ieeexplore.ieee.org/document/8237336/