KHATULISTIWA SMART : JOURNAL OF ARTIFICIAL INTELLIGENCE
e-ISSN : 3089-9710
p-ISSN : XXXX-XXXX

Khatulistiwa Smart : Journal of Artificial Intelligence
Vol. 1 No. 2 (2025) : 67-78
Available online at : https://journal.literasikhatulistiwa.org/index.php/smart

Application of Siamese Neural Network for Offline Signature Verification Based on Similarity Level

Ahmad Halimi [1]
[1] Universitas Nurul Jadid
ahmadhalimi@unuja.ac.id[1]

Doi

Received: 27 Mei 2025 *Accepted: 18 Juni 2025* *Published: 30 Juni 2025*

Abstract The increasing demand for secure and accurate identity verification systems has led to the development of various biometric technologies, one of which is signature verification. Despite the rise of digital authentication methods, signatures remain a widely accepted and legally binding form of identity verification, especially in paper-based systems. This research explores the application of the Siamese Neural Network (SNN) method for offline signature verification based on image similarity levels. The study aims to reduce human error, speed up verification time, and increase accuracy in identifying genuine and forged signatures. The dataset used in this study consists of 210 signature images collected from 14 respondents, including 7 with genuine signatures and 7 with forged signatures (categorized as random, unskilled, and skilled forgeries). Preprocessing steps such as scanning, resizing, and CSV data generation were conducted to optimize input for the SNN model. The model was trained using contrastive loss to learn signature similarity representations and was evaluated using a confusion matrix. The training dataset included 147 image pairs, and the testing set contained 63 image pairs, resulting in 168 prediction possibilities. The SNN achieved an accuracy rate of 94%, correctly predicting 159 cases while misclassifying 8 due to image quality and unclear signature strokes. These results indicate that the Siamese Neural Network is effective for offline signature verification and demonstrates strong potential for real-world implementation in identity authentication systems. This research contributes to the field of computer vision, particularly in biometrics, by providing an efficient, learning-based approach to signature validation using deep learning techniques.

Keywords *Signature; Siamese Neural Network; Pre-processing; Confusion Matrix*

1. INTRODUCTION

The rapid development of advanced technology above the average, today is very helpful for humans individually and in groups in completing various things. Various new technologies have been successfully developed, one example is in the field of security (security). systems). Such as smart lock with fingerprint, face recognition, eyes detector and many others to gain access to an application or a system. However, there are some things that do not allow us to use some of these technologies to access something, for example, to get permission to withdraw money, check validation, and various things that still use conventional technology (paper). based). A signature is an undeniable and unique way of proving one's identity. For reasons of uniqueness and simplicity, signatures have a special and important place in the field of biometrics (Jain, Singh, & Singh, 2020) . So signature identification is a biometric verification of identity that is widely accepted in society and is still the main mechanism for authentication and authority in many transactions (Rateria, 2018) .

In general, things that use paper based still use file identification with a signature. The signature identification process can be done manually by verifying the signature indication at the time of the transaction using an original and valid signature. This manual system is prone to the problem that the (human) signature examiner often makes mistakes (human error) and is not careful when doing the matching. Signature pattern recognition identification technology is included in biometrics that uses natural human behavioral traits or behaviors.

This is the concern of researchers to examine a system with a method that can analyze the characteristics of the signature to facilitate the signature verification process. This will reduce the occurrence of human errors, accelerate identification and verification time when compared to conventional identification.

In research (Wu et al. 2019) raised the issue of Most studies related to optimizing global distance goals and accommodating low discriminatory power due to the loss of temporal information from the implementation of the Siamese Neural Network . This study proposes an end-to-end artificial neural network based framework to study local delegation of time series and demonstrate its effectiveness for online signature verification. The validation related to online signature verification shows the advantages of the framework that the researcher proposes over other techniques that use handmade or learned feature representations. This study does not explain the differences and accuracy of comparisons between frameworks. Subsequent research proposes an artificial neural network-based framework to study local representations based on varying time series and demonstrates its effectiveness for online signature verification. Researchers combine the Siamese Neural Network into Dynamic Time Warping (DTW). Point it to a loss-optimized Prewarping Siamese Neural Network (PSN) with local embedding. The researcher concludes that the validation of the online signature verification data set shows the superiority of the proposed king over other techniques that use handcrafted or learned feature representations. The differences and accuracy of each framework in this study have not been explained (Wu and Uchida 2019) .

Then another study proposed the use of the Siamese Neural Network to overcome the problem of offline signature verification by random guessing in an author-independent context. The solution from this research, can be used on new signatures without the need for additional training . It was concluded that the best verification results were obtained from combining real and fake signatures in training (Ruiz et al. 2019) . Recent research points to the failure of the traditional Siamese Neural Network to fully represent the author's writing style and to experience a decline in performance when the distribution of samples of positive and negative signatures is unequal. The researcher proposes a Siamese Neural Network model with two stages of verification. First adopts Siamese Neural Network to verify the original enhanced signature together. The second utilizes local losses to address the imbalance between positive and negative offline signatures. This proposal solves the existing problems and achieves good performance on datasets with different languages (Verification, 2022) .

Based on the background and some explanations from the research that has been described. So the researchers raised the title "Signature Verification Based on Similarity Level With Siamese Neural Network". The expected contribution of this research is to contribute knowledge in Computer Vision. In particular, the use of the Siamese Neural Network method in signature verification.

2. RESEARCH METHODS

The author uses the experimental method with the aim of knowing the structure of the presentation and processing of information that involves processing has an impact on decision making and presentation of results . Where in this case the first stage starts from data collection to drawing conclusions. The stages of the research are described in Fig 1

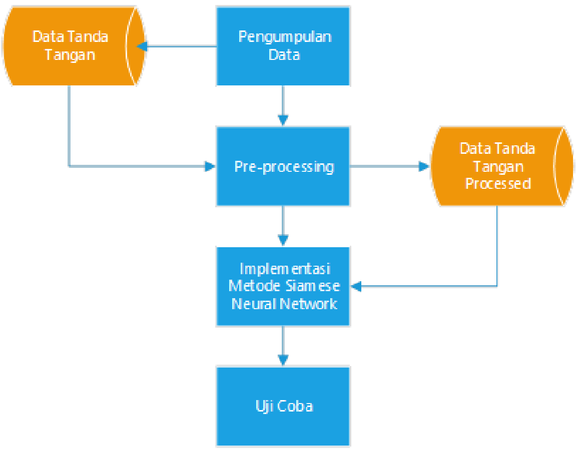

Figure 1. Research Stages

2.1. Data collection

This study uses data in the form of photo signatures on paper taken from 14 respondents. With each respondent amounted to 7 people. Respondents with original signatures were 7 people with 10 signatures each. Respondents with fake signatures were 7 people, each of whom gave 20 signatures with 3 kinds of patterns, namely random, unskilled and skilled fake signatures. The following is an explanation of the types of fake signature patterns (Bharadi & Kekre, 2010):

2.1.1. Random Fake Signatures: Simple or random fake signatures that are very easy to detect even with the naked eye.

2.1.2. Unskilled Fake Signature: A fake signature whose impersonator uses his own style with no knowledge of spelling and no prior experience.

2.1.3. Skilled Fake Signature: A fake signature whose imitator is trained or professional in copying signatures.

From the signature dataset that was collected from 14 respondents with the conditions described previously, 210 signature data were collected to be used. Figure 2. is an example of the collected dataset.

Original Signature Fake Signature

Figure 2. Example of a Signature Dataset

2.2. Preprocessing

Preprocessing of the signature image is carried out to reduce the influence of the background, size and location of the signature on the verification performance (Liu, Huang, Yin, & Chen, 2021). This stage is carried out for the effectiveness of using the Siamese Neural Network method. The preprocessing stages that will be carried out are scanning the original image, separating the training data and testing data, resizing the image and inputting data to csv. Figure 3. is an illustration of the preprocessing step.

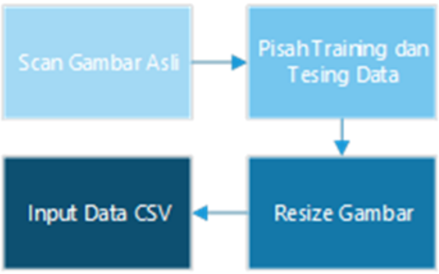

Figure 3. Preprocessing Step

2.3. Siamese Neural Network Implementation

Siamese Neural Network can train one image input and one image target. Recently, the Siamese Neural Network was used to recognize between two similar or different images. The Siamese Neural Network has two identical branch networks with CNNs that share the same parameters and weights. This framework is used to reduce dimensions that have a constructive deficiency function at the bottom of the network (B.

J. B., Sawat, & Hegadi, 2019). The calculation of the similarity and dissimilarity between pairs of dengens using a distance metric is called the Euclidean distance, the contractive loss function is presented in equation 1.

$$L(c, s_1, s_2,) = (1 - c)\frac{1}{2}(D_w)^2 + (c)\frac{1}{2}\{max(0, m - D_w)\}^2 \quad (1)$$

The process of applying the Siamese Neural Network method to the training data image will be tested with data testing. The results of the process will be stored as reference data during the trial process. The resulting data is in the form of truth value data from the process.

2.4. Trials

The trial phase is carried out to ensure the method used works as expected and is good. This stage is implemented in the Siamese Neural Network method to get the accuracy value from the testing results. From this stage it will be known whether this method will work or not. Calculation of accuracy in the Siamese Neural Network method using the Confusion Matrix. Confusion Matrix is a method that is generally used as a method of calculating the level of accuracy in data mining. This method contains an info about the prediction of classification correctly by a classification system (Gunawan, Pratiwi, & Pratama, 2018). The following formula is used to calculate the level of accuracy in the image test, described in equation 2:

$$Akurasi = \frac{\sum Data\ Benar}{\sum Data\ Kemungkinan}\ x\ 100\ \%\ (2)$$

Information :
Correct Data = Lots of Correct Data
Possible Data = Lots of Possible Data

3. RESULTS AND DISCUSSION

In the results and discussion stage, the implementation of the Siamese Neural Network on signature verification is discussed. The steps include collecting datasets, preprocessing results, implementing the Siamese Neural Network method and testing.

3.1. Dataset Collection

As has been explained, the dataset was taken from 14 respondents with the division of 7 respondents with original signatures and 7 respondents with fake signatures. The 7 respondents with original signatures were obtained from the leadership ranks of the Faculty of Engineering, University of Nurul Jadid Paiton Probolinggo, including the Head of Electrical Engineering Study Program, Head of Informatics Engineering Study Program, Head of Information Systems Study Program, Head of Information Technology Study Program, Head of Software Engineering Study Program, Secretary of the Electrical & Informatics Study Program, and the Dean Faculty of Engineering. Meanwhile, 7 respondents got fake signatures from Nurul Jadid University students.

From 14 respondents, 210 datasets were obtained, with 70 datasets of original signatures and 140 datasets of fake signatures being divided. Then the 210 datasets will be re-divided into training data and testing data. With each as many as 147 training data and 63 testing data. The division process is done manually. Examples of training and testing data can be seen in Figures 4 and 5.

Figure 4. Training Data

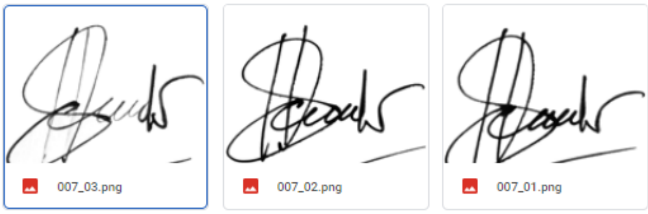

Figure 5. Testing Data

3.2. Preprocessing

For the effective implementation of the Siamese Neural Network method, the following preprocessing stages must be carried out, namely as follows:

Image Scan: The image scan process is carried out to further emphasize the pattern on the signature image, so that the image is easier to process. The process of scanning images using the Online Cam Scanner website on the website www.onlinecamscanner.com. Where in the scan process the image is saved in png format. The results of the image scan process are shown in Figure 6.

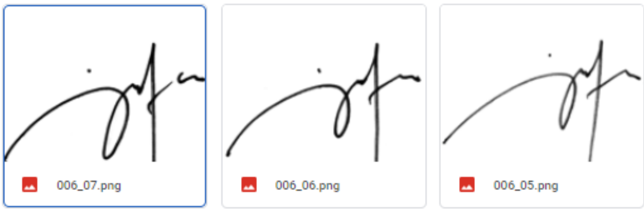

Figure 6. Image Scan Process Results

Image Resize: At this stage the resizing process is carried out from the image which initially has a size of 1.71 MB with a resolution of 3096 x 3096 pixels to a smaller size of 14 KB with a resolution of 150 x 150 pixels. This process is carried out in Python. The resizing process is carried out on training data and testing data. The syntax can be seen in Figure 7.

```python
import PIL
import os
import os.path
from PIL import Image


root_folder =
'/content/drive/MyDrive/dataa/png4/training'
folders = [os.path.join(root_folder, x) for x in
('/content/drive/MyDrive/dataa/png4/training/001',
'/content/drive/MyDrive/dataa/png4/training/002',
'/content/drive/MyDrive/dataa/png4/training/003',
'/content/drive/MyDrive/dataa/png4/training/004',
'/content/drive/MyDrive/dataa/png4/training/005',
'/content/drive/MyDrive/dataa/png4/training/006',
'/content/drive/MyDrive/dataa/png4/training/007')]
all_images = [img for folder in folders for img in
(folder)]
tgt_base_path =
"/content/drive/MyDrive/dataa/dataset_150x150/train
ing"


for cur_path in os.listdir(root_folder):
    src_sub_path = os.path.join(root_folder,
cur_path)
    tgt_sub_path = os.path.join(tgt_base_path,
cur_path)

    if not os.path.isdir(tgt_sub_path):
        os.mkdir(tgt_sub_path)
    jdx = 0
    for filename in os.listdir(src_sub_path):
        current_img = Image.open(src_sub_path +
'//' + filename)
        target_path = os.path.join(tgt_sub_path,
"%s-%03d.png" % (cur_path, jdx+1))
        print('Working on image: ' +
os.path.splitext(filename)[0])
        print(
            f'Format: {current_img.format}, Size:
{current_img.size}, Mode: {current_img.mode}')
        print(target_path)

        # Resize gambar + simpan output ke folder
        img = current_img.resize((150,150))
        img.save(target_path)
        jdx += 1
```

Figure 7. Image Resize Syntax

Creating a CSV File: After the resizing stage, the next step is to input the image name and image storage folder using Microsoft Excel in CSV format. There are 2 files needed, namely the CSV file for training data and testing data. The training data CSV file contains 980 possible signatures from 14 respondents, consisting of 7 original signatures and 7 fake signatures, each of which contains 14 images so that a total of 147 possible signature images are made. And for the testing data, it contains 168 possible images from 14 respondent's signatures consisting of 7 original signature datasets, each of which has 3 images and 7 fake signature datasets, each of which contains 6 images, so that a total of 63 images are possible. . Tables 1 and 2 show examples of csv files for training data and testing data .

Table 1. Training Data Csv File

001/001_01.png	001/001_02.png	0
001/001_01.png	001/001_03.png	0
001/001_01.png	001/001_04.png	0
001/001_01.png	001/001_05.png	0
001/001_01.png	001/001_06.png	0
…	…	…
007/007_07.png	007_duplicate/0007_10.png	1
007/007_07.png	007_duplicate/0007_11.png	1
007/007_07.png	007_duplicate/0007_12.png	1

007/007_07.png	007_duplicate/0007_13.png	1
007/007_07.png	007_duplicate/0007_14.png	1

Table 2. Data Testing Csv File

001/001_01.png	001/001_02.png	0
001/001_01.png	001/001_03.png	0
001/001_01.png	001_duplicate/0001_1.png	1
001/001_01.png	001_duplicate/0001_2.png	1
001/001_01.png	001_duplicate/0001_3.png	1
…	…	…
007/007_03.png	007_duplicate/0007_2.png	1
007/007_03.png	007_duplicate/0007_3.png	1
007/007_03.png	007_duplicate/0007_4.png	1
007/007_03.png	007_duplicate/0007_5.png	1
007/007_03.png	007_duplicate/0007_6.png	1

3.3. Implementation of the Siamese Neural Network Method

There are several steps that must be taken in implementing the Siamese Neural Network method, here are the steps. Data preparation : The first step is data preparation, training and testing data is prepared, both data in the form of images and csv files. The syntax of load data is shown in Figure 8.

```
training_dir="/content/drive/MyDrive/dataa/dataset_150x150/training"
training_csv="/content/drive/MyDrive/dataa/csv/trainingcsv.csv"
testing_csv="/content/drive/MyDrive/dataa/csv/testingcsv.csv"
testing_dir="/content/drive/MyDrive/dataa/dataset_150x150/testing"
```

Figure 8. Load Data Syntax

Pythorch Costume: Next performs a pythorch costume to generate a predictive or probable image pair, a value of 0 for the original signature image pair and 1 for the fake signature image pair. An example of the implementation of the pythorch costume is in Figure 9.

	001/001_01.png	001/001_02.png	0
4	001/001_01.png	001/001_07.png	0

Figure 9. Example of Pythorch Kostum Costume Implementation Results

Training Data: After the dataset gets a label from the Python costume process, the next stage is the training stage using the Siamese Neural Network method. This stage will go through several other stages, namely the implementation of the Siamese Neural Network method, then determining the Contrastive Loss, then training data. Loss

Contrastive works on a pair of samples, it defines a binary indicator Y for each sample pair which states whether a pair of samples should be considered similar or not, and a learnable distance function D_W(x_1, x_2) between a pair of samples x_1, x_2, parameterized by the weight W in the neural network, , where m>0 is the margin. Margin defines a radius around the sample insertion space so that different sample pairs only contribute to the Contrastive Loss Function if the distance D_W is within the margin. Intuitively, this Loss Function encourages neural networks to learn embedding to place samples with the same label close to each other, while keeping samples with different labels in the embedding space. The syntax of Contrastive Loss is shown in Figure 10.

```
class ContrastiveLoss(torch.nn.Module):

    def __init__(self, margin=1.5):
        super(ContrastiveLoss, self).__init__()
        self.margin = margin

    def forward(self, output1, output2, label):
        euclidean_distance = F.pairwise_distance(output1, output2)
        loss_contrastive = torch.mean((1-label) * torch.pow(euclidean_distance, 2) +
                                      (label) * torch.pow(torch.clamp(self.margin - euclidean_distance, min=0.0), 2))


        return loss_contrastive
```

Figure 10. Contrastive Loss

For the data training process using epoch 110, epoch is a hyperparameter that determines how many times the learning algorithm will work to process the entire training dataset. One epoch means that each sample in the training dataset has the opportunity to update the internal data model parameters with the input results from the dataset processing that has been done previously. The syntax for training data is shown in Figure 11.

```python
def train():
    loss= []
    counter=[]
    iteration_number = 0

    for epoch in range(1,10):
        for i, data in enumerate(train_dataloader,0):
            img0, img1 , label = data
            img0, img1 , label = img0.cuda(), img1.cuda() , label.cuda()
            optimizer.zero_grad()
            output1,output2 = net(img0,img1)
            loss_contrastive = criterion(output1,output2,label)
            loss_contrastive.backward()
            optimizer.step()

        print("Epoch {}\n Current loss {}\n".format(epoch,loss_contrastive.item()))

        counter.append(iteration_number)
        loss.append(loss_contrastive.item())
    plt.plot(loss)
    return net
```

Figure 11. Training Data Syntax

Then the results of the training data are stored in the form of a model with the extension ".pt" which will be a model at the data testing stage.

3.4. Trials

The trial phase is a continuation of the implementation phase of the model that has been made in the process of implementing the Siamese Neural Network method. At this stage, the accuracy of the Siamese Neural Network model will be known. The syntax of this step is shown in Figure 12.

```python
for i, data in enumerate(test_dataloader,0):
  x0, x1 , label = data
  concat = torch.cat((x0,x1),0)
  output1,output2 = model(x0.to(device),x1.to(device))

  eucledian_distance = F.pairwise_distance(output1, output2)

  if eucledian_distance<=0.50:
    label="Asli"
  else:
    label="Palsu"

  imshow(torchvision.utils.make_grid(concat))
  print("Hasil ke-",count)
  print("Jarak kemiripan: " '%.2f' % eucledian_distance.item())
  print("Label: ",label)
  count=count+1
  if count ==168:
     break
```

Figure 12. Data Testing Syntax

The trial will be carried out with 14 test data consisting of 7 Heads of Study Programs and 7 Students of the Faculty of Engineering. Using the results of epoch 110 from the training process, the test results are shown in Table 3.

Table 3. Data Testing Results

Many Image Possibilities	Lots of True Data	Lots of Wrong Data	Value Accuracy
168	159	8	94%

4. CONCLUSIONS

The Siamese Neural Network method can recognize the signatures of the leadership ranks of the Nurul Jadid Faculty of Engineering well. The introduction uses 7 respondents with original signatures and 7 respondents with fake signatures with a total of 147 images, and a file with *csv format containing 980 possibilities for the training phase so as to get 200 epochs. The data was tested using 7 original signature datasets. and 7 fake signature datasets with a total of 63 images, and files in *csv format containing 168 possibilities, from the test results obtained 8 possibilities that produce wrong predictions, wrong predictions are mostly caused by broken signature patterns, and signature patterns. If the hand is not clear, there are 159 possibilities that produce correct predictions, so that the trial results in an accuracy value of 94%.

5. REFERENCES

Jain, A., Singh, S. K., & Singh, K. P. (2020). Handwritten signature verification using shallow convolutional neural network. Proceedings of International Conference on Emerging Trends in Communication, Control and Computing. https://doi.org/10.1007/s11042-020-08728-6

Rateria, A. (2018). Off-line signature verification through machine learning. In 2018 5th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON) (pp. 1-7). https://doi.org/10.1109/UPCON.2018.8597090

Wu, X., Kimura, A., Iwana, B. K., Uchida, S., & Kashino, K. (2019). Deep dynamic time warping: End-to-end local representation learning for online signature verification. In Proceedings of the International Conference on Document Analysis and Recognition (ICDAR). https://doi.org/10.1109/ICDAR.2019.00179 https://doi.org/10.1109/ICDAR.2019.00179

Wu, X., & Uchida, S. (2019). Prewarping Siamese network: Learning local representations for online signature verification. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). https://doi.org/10.1109/ICASSP.2019.8683036 https://doi.org/10.1109/ICASSP.2019.8683036 PMCid:PMC6554277

Ruiz, V., Linares, I., Sanchez, A., & Velez, J. F. (2019). Off-line handwritten signature verification using compositional synthetic generation of signatures and Siamese neural networks. Neurocomputing. https://doi.org/10.1016/j.neucom.2019.09.041 https://doi.org/10.1016/j.neucom.2019.09.041

(Author not specified). (2022). A two-stage Siamese network model for offline handwritten signature verification. Symmetry, 14(1). [Catatan: Harap lengkapi nama penulis jika tersedia.] https://doi.org/10.3390/sym14061216

Bharadi, V. A., & Kekre, H. B. (2010). Off-line signature recognition systems. International Journal of Computer Applications, 1(2). https://doi.org/10.5120/499-815 https://doi.org/10.5120/499-815

Liu, L., Huang, L., Yin, F., & Chen, Y. (2021). Offline signature verification using a region-based deep metric learning network. Pattern Recognition, 118, 108009. https://doi.org/10.1016/j.patcog.2021.108009 https://doi.org/10.1016/j.patcog.2021.108009

B. J. B., A., Sawat, D. D., & Hegadi, R. S. (2019). Siamese network for learning genuine and forged offline signature verification (Vol. 1). Springer. https://doi.org/10.1007/978-981-13-9187-3 https://doi.org/10.1007/978-981-13-9187-3

Gunawan, B., Pratiwi, H. S., & Pratama, E. E. (2018). A sentiment analysis system in product reviews using the Naive Bayes method. Jurnal Pendidikan dan Peneliti Informasi (JPPI), 4(2), 113-120. https://doi.org/10.26418/jp.v4i2.27526 https://doi.org/10.26418/jp.v4i2.27526