

Desain Absensi Mahasiswa Dengan Tanda Tangan Digital Terverifikasi Berbasis Convolutional Neural Network di Masa Pandemi Covid-19

Asrul Abdullah^{*1}, Syafaat Agung Prakoso², Medi Taruk³

^{1,2}Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Muhammadiyah Pontianak, Pontianak

³Program Studi Informatika, Universitas Mulawarman, Samarinda

e-mail: ^{*1}asrul.abdullah@unmuhpnk.ac.id, ²syafaat.agung@unmuhpnk.ac.id,
³meditaruk@gmail.com

Abstrak

Pandemi Covid-19 telah membawa banyak perubahan dalam pelaksanaan proses pembelajaran. Sebelum pandemi Covid-19, proses pembelajaran dilaksanakan secara luring dengan memanfaatkan berbagai media. Namun, pandemi Covid-19 membuat proses pembelajaran berubah menjadi daring. Selama proses pembelajaran online yang telah dijalankan pada masa pandemi Covid-19 ini, absensi tidak dilakukan oleh mahasiswa melainkan oleh dosen dengan cara melihat kehadiran mahasiswa saat melakukan tatap muka melalui video conference. Akan tetapi proses absensi seperti ini dirasa kurang efektif dan memiliki kendala seperti kecurangan kehadiran yang dilakukan oleh mahasiswa dengan cara tetap online saat video conference tetapi mahasiswa bersangkutan tidak ikut interaktif di dalamnya, dosen kesulitan dalam menentukan syarat mengikuti UAS yakni sebesar 75%. Oleh sebab itu, penggunaan tanda tangan digital terverifikasi dalam proses absensi mahasiswa perlu diterapkan di lingkungan UM Pontianak. Tujuan dari penelitian ini adalah mencari tingkat similaritas dari model dengan gambar tanda tangan dengan menggunakan Convolutional Neural Network. Metode dalam penelitian ini adalah menggunakan Siamese Convolutional Neural Network dengan tahapan sebagai berikut : collect tanda tangan digital, ReLu, Max Pooling dan Dropout, Flatten and Dense serta Softmax dan pengujian. Hasil dari penelitian ini adalah Model yang telah dibuat telah berhasil melakukan uji similaritas sehingga dapat membedakan mana tanda tangan asli dan palsu. Siamese Convolutional Neural Network sangat baik dalam melakukan modelling terhadap data gambar dan video.

Kata kunci— Convolutional Neural Network, CNN, Covid-19, Siamese

1. PENDAHULUAN

Pandemi Covid-19 telah membawa banyak perubahan dalam pelaksanaan proses pembelajaran. Sebelum pandemi Covid-19, proses pembelajaran dilaksanakan secara luring dengan memanfaatkan berbagai media. Namun, pandemi Covid-19 membuat proses pembelajaran berubah menjadi daring. Menariknya, dukungan teknologi informasi seakan membuat pembelajaran daring menjadi semakin fleksibel. Dosen dapat memberikan materi secara daring kepada mahasiswa meskipun tidak menjamin materi yang disampaikan akan bisa diserap sepenuhnya oleh mahasiswa. Salah satu indikator yang menunjang proses belajar mengajar adalah kehadiran mahasiswa saat melakukan tatap muka secara daring. Hadirnya mahasiswa ditandai dengan absensi melalui tanda tangan yang dibubuhkan pada berita acara perkuliahan. Tentunya pandemi ini membuat perubahan metode absensi mahasiswa yang semula *offline* menjadi *online*.

Selama proses pembelajaran online yang telah dijalankan pada masa pandemi Covid-19 ini, absensi tidak dilakukan oleh mahasiswa melainkan oleh dosen dengan cara melihat kehadir

mahasiswa saat melakukan tatap muka melalui *video conference*. Sedangkan jika perkuliahan online hanya dilakukan dengan pemberian tugas dan materi, kehadiran dilihat dari tugas yang dikumpulkan oleh mahasiswa. Akan tetapi proses absensi seperti ini dirasa kurang efektif dan memiliki kendala seperti kecurangan kehadiran yang dilakukan oleh mahasiswa dengan cara tetap online saat *video conference* tetapi mahasiswa bersangkutan tidak ikut interaktif di dalamnya, dosen kesulitan dalam menentukan syarat mengikuti UAS yakni sebesar 75%. Oleh karena itu diperlukan adanya inovasi yang dapat memudahkan dosen dalam memeriksa kehadiran mahasiswa saat proses pembelajaran berlangsung. Salah satunya adalah melalui tanda tangan digital terverifikasi yang dapat dilakukan oleh mahasiswa seperti saat perkuliahan *offline*. Pentingnya tanda tangan membuat banyak orang yang tidak bertanggung jawab sering melakukan pemalsuan tanda tangan sehingga diperlukan sebuah verifikasi untuk membuktikan keaslian sebuah tanda tangan. Teknik dari verifikasi tanda tangan itu sendiri terbagi menjadi dua yakni *offline (static signature verification technique)* dan *online (dynamic signature verification technique)*.

Teknik *online* menggunakan tekanan elektronik tablet yang sensitif untuk mengekstrak informasi tentang tanda tangan dan mengambil informasi dinamis seperti tekanan, kecepatan, kecepatan penulisan, jumlah urutan guratan dan pena tekanan pada setiap titik untuk keperluan verifikasi itu membuat tanda tangan lebih unik dan lebih sulit buat ulang [1][2]. Untuk verifikasi tanda tangan online, bentuk kecerdasan buatan yang paling umum adalah jaringan saraf tiruan dasar. Salah satu pendekatan yang akan diterapkan menggunakan *Deep Learning* yakni *Convolutional Neural Network (CNN)*. CNN sebagian besar digunakan algoritma dalam pengenalan gambar dan wajah. CNN adalah sejenis jaringan saraf tiruan yang menggunakan metode konvolusi untuk mengekstrak fitur dari input data untuk meningkatkan jumlah fitur. CNN pertama kali diusulkan oleh LeCun dan pertama kali diterapkan dalam pengenalan tulisan tangan [3]. Oleh sebab itu, pendekatan CNN akan saya terapkan pada pembuatan desain tanda tangan digital terverifikasi dalam penelitian ini. Sejumlah penelitian terdahulu yang menjadi *state of the art* dalam penelitian ini dilakukan oleh [3], mereka mengimplementasikan salah satu algoritma di dalam *deep learning* yakni *backpropagation* dalam mengenal kode pos sebuah wilayah yang ditulis dengan tangan. Hasil penelitian mereka adalah *backpropagation* mampu mengenali kode pos yang ditulis dengan tangan. Selain itu, pendekatan yang mereka lakukan menjadi dasar bagi penelitian selanjutnya dalam pengenalan digit. Penelitian tentang pengenalan dan verifikasi tanda tangan dilakukan oleh [4] dengan menggunakan *artificial neural network*. Tanda tangan terverifikasi berdasarkan *parameter* yang diekstraksi dari tanda tangan menggunakan berbagai pemrosesan gambar (*image processing*). Untuk verifikasi tanda tangan beberapa fitur unik perlu diekstraksi. Hasil fitur yang diekstraksi digunakan untuk melatih jaringan saraf dengan menggunakan algoritma *backpropagation*.

Pengenalan dan kemampuan verifikasi sistem dapat ditingkatkan dengan menggunakan fitur tambahan dalam input dataset. Penelitian ini bermaksud mengurangi minimum kasus pemalsuan dalam transaksi bisnis [5]. Penelitian berbeda yakni dengan verifikasi tanda tangan online dilakukan oleh [6]. Mereka menggunakan *neural network* yang diperkuat dengan ekstraksi fitur *principle component analysis (PCA)*. PCA adalah salah satu metode statistik yang paling populer digunakan untuk ekstraksi fitur, pengurangan dimensi, dan representasi data dalam pengenalan pola dan *computer vision* [7]. Hasil penelitiannya adalah tingkat akurasi dengan menggunakan *artificial neural network* dengan ekstraksi fitur PCA sebesar 93,1%.

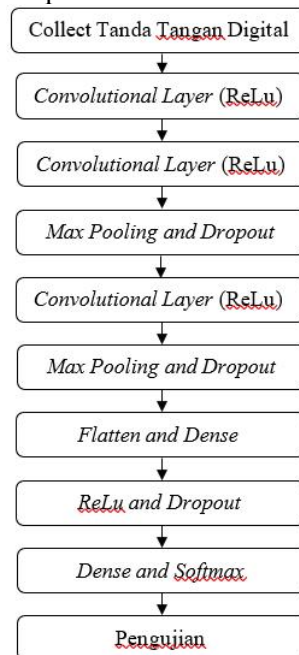
Penelitian dengan verifikasi tanda tangan dengan *offline* menggunakan *Deep Learning* dilakukan oleh [8]. Hasil penelitiannya adalah ekstraksi fitur dengan *Convolutional Neural Network* menggunakan SVM (*Support Vector Machine*) dengan fungsi kernel menggunakan Cubic dengan tingkat akurasi mencapai 96,6%. Penelitian yang dilakukan oleh [9] menggunakan *Convolutional Neural Network* untuk verifikasi tanda tangan *offline*. Mereka melakukan *train* data tanda tangan dengan menggunakan model yang berbeda yakni *writer dependent* dan *writer independent*. Hasil uji eksperimental memperlihatkan *writer independent* memiliki persentase kesuksesan sebesar 62,5 % dan *writer dependent* memiliki persentase kesuksesan sebesar 75 %.

Model lainnya yang juga melakukan verifikasi tanda tangan menggunakan *tensorflow* dilakukan oleh [10]. Hasil penelitiannya adalah nilai FRR dan FAR adalah sebesar 5% yang berarti sistem memberikan akurasi hingga 90% pada 600 *images* pada fase *training*. Verifikasi tanda tangan *online* menggunakan *Deep Descriptors* dilakukan oleh [11]. Proses verifikasi menggunakan *Convolutional Neural Network* dengan diikuti oleh *Recurrent Neural Network*. Dataset yang digunakan adalah SVC 2004 dan SigComp2009. Akurasi pengujian mencapai 97,50 % dengan FAR (*false acceptance rate*) sebesar 5%, FRR (*false rejection rate*) sebesar 0%, EER (*equal error rate*) sebesar 5% dan ROC AUC (*receiver operationg characteristics area under cover*) sebesar 0.9975.

Jika pandemi Covid-19 berakhir, penggunaan tanda tangan digital terverifikasi ini juga dapat digunakan bagi dosen yang berhalangan hadir di kelas ataupun ingin melakukan perkuliahan jarak jauh. Di Universitas Muhammadiyah Pontianak, proses absensi belum menerapkan tanda tangan digital sebagai bukti bahwa mahasiswa mengikuti perkuliahan daring. Absensi yang ada di UM Pontianak untuk versi online-nya menggunakan informasi / keterangan yang menyatakan bahwa mahasiswa itu hadir tetapi bukti otentiknya tidak ada, sehingga membuat kesulitan dosen dalam melaporkan kinerja mengajar. Hal ini akan berimbas pada keterlambatan pencairan tunjangan sertifikasi dosen. Oleh sebab itu, penggunaan tanda tangan digital terverifikasi dalam proses absensi mahasiswa perlu diterapkan di lingkungan UM Pontianak

2. METODE PENELITIAN

Tahapan yang dilakukan dalam penelitian dari awal (*capturing image*) sampai pengujian tanda tangan seperti yang ditunjukkan pada Gambar 1.



Gambar 1 Alur Tahapan Penelitian

Gambar 1 merupakan flowchart dari tahapan penelitian sehingga tujuan penelitian dapat tercapai. Untuk dapat berpindah dari satu tahap ke tahap lain, maka setiap tahapan harus sudah selesai.

- Input Tanda Tangan Digital

Langkah pertama adalah mengumpulkan dataset tanda tangan baik yang original dan forgery (palsu).

- Convolutional Network

Convolutional network [3] atau yang biasa dikenal dengan *convolutional neural network* atau CNNs adalah jenis khusus dari jaringan saraf untuk pemrosesan data yang diketahui seperti topologi grid / kotak. Nama dari *convolutional neural network* memberi petunjuk bahwa jaringan menggunakan operasi matematika yang disebut dengan *convolution* / lilitan. *Convolution* adalah jenis khusus dari operasi linear sehingga dapat diambil kesimpulan bahwa convolutional networks hanyalah jaringan saraf yang menggunakan konvolusi sebagai gantinya perkalian matriks umum setidaknya pada salah satu lapisannya.

Pada layer ini sejumlah operasi (*filter*) terhadap image (tanda tangan) dilakukan. Sebuah gambar memiliki 3 parameter yakni tinggi, lebar dan tebal. Pada *filter* juga demikian memiliki ukuran tinggi, lebar dan tebal. *Filter* ini diinisialisasi dengan nilai tertentu, dan nilai dari filter inilah yang menjadi parameter untuk dilakukan perubahan (*update*) dalam proses deep learning.

- Relu Activation

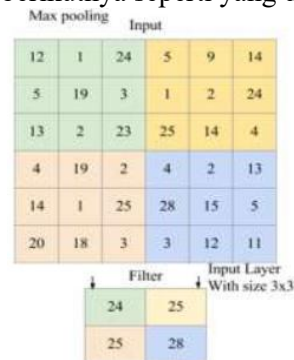
The Rectified Linear Unit (ReLU)[12] adalah fungsi aktivasi yang beroperasi pada output dari *convolutional layer*. Hasil fungsi ReLu sama dengan input jika input > 0. Selain itu output dipetakan ke 0. Fungsi ini secara matematis direpresentasikan pada **persamaan (1)**.

$$f(x; W, c, w, b) = w^T \{0, W^T x + c\} + b \quad (1)$$

Dimana x adalah input yang diberikan ke fungsi aktivasi. Menggunakan ReLu membantu dalam waktu training lebih sedikit di jaringan dan juga menyediakan karakteristik non linear sehingga jaringan berkinerja baik bahkan jika input yang diterima sedikit rumit.

- Max Pooling

Lapisan ini memisahkan data input menjadi beberapa daerah bukan negatif di kedua arah vertikal dan horisontal di langkah N . Dalam setiap langkah lapisan mengambil nilai maksimum di jendela saat ini dan menggabungkannya menjadi yang baru matriks yang berfungsi sebagai input ke matriks berikutnya seperti yang ditunjukkan pada Gambar 2.



Gambar 2 Contoh Max Pooling[13]

- Dropout

Ini adalah teknik sederhana untuk mengurangi ukuran jaringan secara acak dengan mengabaikan neuron saat pelatihan. Teknik ini mengurangi waktu training dari CNN dan membuat jaringan kurang sensitif terhadap bobot tertentu. Ini membuat jaringan lebih umum dan mengurangi kemungkinan *overfitting* jaringan.

- Flatten and Dense

Flatten layer meratakan argumen input menjadi *array* satu dimensi terlepas dari bentuk input yang diberikan kepada layer. *Dense layer* adalah nama lain dari *fully connected layer*. Neuron pada lapisan ini memiliki koneksi ke semua neuron aktivasi pada lapisan sebelumnya.

- Softmax

Lapisan SoftMax [14] bertindak sebagai kalkulator probabilitas untuk setiap kelas yang diberikan. Ini adalah fungsi eksponensial yang bila dijumlahkan selalu memberikan satu sebagai output karena jumlah semua probabilitas sama dengan satu. Secara matematis direpresentasikan dalam **persamaan (2)**.

$$\sigma(Z)_j = \frac{e^z_j}{\sum_{k=1}^k e^z_k} \quad (2)$$

3. HASIL DAN PEMBAHASAN

Beberapa tahapan yang harus dilalui agar dapat menghasilkan model yang dapat mengukur tingkat original dan palsu dalam sebuah tanda tangan antara lain sebagai berikut :

- Data Preprocessing

Langkah pertama adalah melakukan preprocessed terhadap data *image* yakni berupa tanda tangan dan kemudian disimpan sebagai *.npy* (*numpy*). Ini bertujuan untuk memudahkan transfer.

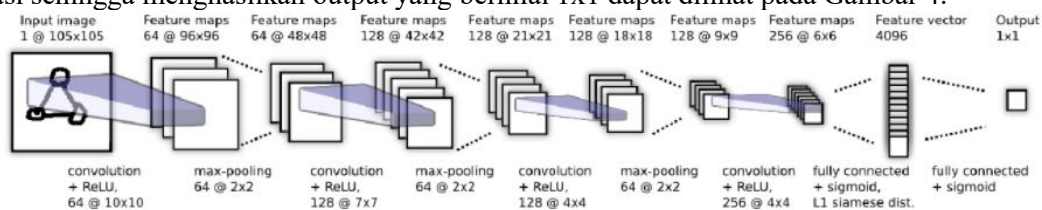
Beberapa langkah-langkah yang dilakukan dalam *preprocessing* seperti memberi label dan path dari *images* tanda tangan yang telah disimpan baik di lokal atau cloud. Langkah berikutnya buka gambar tanda tangan yang path-nya sudah benar. Lakukan image transformasi. Untuk melihat visualisasi dari gambar tanda tangan dapat dilihat pada Gambar 3.



Gambar 3 Hasil Load Dataset

- Define the Siamese Network

Tahap ini merupakan design dari arsitektur CNN, sebagai contoh mudah dari beberapa iterasi sehingga menghasilkan output yang bernilai 1x1 dapat dilihat pada Gambar 4.



Gambar 4 Tahapan Design Arsitektur CNN

Pada Gambar akan menghasilkan input 100px*100px dan memiliki 3 *fully connected layers* setelah layer convolutional. Dikarenakan untuk kebutuhan verifikasi tentunya diperlukan setidaknya ada dua image yang akan diuji similaritasnya. Langkahnya adalah mengkalkulasi loss value (*contrastive loss*) dari kedua images dan kemudian lakukan *backpropagation*. Hasilnya inilah yang disebut metrics (*accuracy*). Adapun *contrastive loss* ditunjukkan pada Gambar 5.

```
class ContrastiveLoss(torch.nn.Module):
    """
    Contrastive loss function.
    Based on: http://yann.lecun.com/exdb/publis/pdf/hadsell-chopra-lecun-06.pdf
    """

    def __init__(self, margin=2.0):
        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
```

Gambar 5 Contrastive Loss

Langkah berikutnya adalah menentukan *Optimizer*. Ada banyak optimizer yang bisa digunakan seperti RMSprop, Mini Batch Gradient Descent, Adam Optimizer, namun dalam penelitian ini menggunakan RMSProp. Beberapa cuplikan baris kode seperti yang ditunjukkan pada Gambar 6.

```
# Declare Siamese Network
net = SiameseNetwork().cuda()
# Declalre Loss Function
criterion = ContrastiveLoss()
# Declare Optimizer
optimizer = optim.RMSprop(net.parameters(), lr=1e-4, alpha=0.99, eps=1e-8,
weight_decay=0.0005, momentum=0.9)
```

Gambar 6 Optimizer

- Training

Untuk pembuatan model dari data training sudah selesai. Model hasil train dengan gambar yang telah di preprocessing sampai mendapatkan epoch yang terbaik. Beberapa cuplikan kode yang digunakan untuk melakukan training seperti yang ditunjukkan pada Gambar 7.

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

    for epoch in range(0, Config.train_number_epochs):
        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()
            if i % 50 == 0:
                print("Epoch          number          {}\n          Current          loss
{}\n".format(epoch, loss_contrastive.item()))
                iteration_number += 10
                counter.append(iteration_number)
                loss_history.append(loss_contrastive.item())
    return net
```

Gambar 7 Training Model

Dari Gambar 9 dapat disimpulkan proses yang dilakukan yakni melakukan pengukuran loss function terhadap kedua image yang ingin diukur similaritasnya, kemudian kalkulasikan juga gradients, setelah itu update bobot menggunakan optimizer dengan menggunakan RMSprop. Hasil *loss function* seperti yang ditunjukkan pada Gambar 8.

```

Epoch number 8
Current Loss 2.342238759994587

Epoch number 8
Current Loss 1.9322389765628

Epoch number 8
Current Loss 1.2943383189467266

Epoch number 8
Current Loss 1.1868386298391724

Epoch number 8
Current Loss 1.649484393988833

Epoch number 19
Current Loss 1.8586238553192139

Epoch number 19
Current Loss 1.8641468418781172

Epoch number 19
Current Loss 1.2465916872824536

Model Saved Successfully

```

Gambar 8 Loss Function

Langkah selanjutnya adalah melakukan pengujian. Pengujian yang dilakukan menggunakan One-Shot Learning yang di-train dengan *Siamese Network for Classification* atau *Verification*. Data uji yang diberi nama test image X dan akan dikategorikan dalam *one of C categories*. Untuk setiap C category merupakan $X_c = \{X_0, X_1, X_2, \dots, X_{c-1}\}$ images. Kalkulasikan skor similaritas dari X dan X_c images kemudian dilakukan prediksi kelas apakah original atau palsu dengan similaritas paling maksimum. Beberapa cuplikan baris kode seperti yang ditunjukkan pada Gambar 9.

```

# Load the test dataset
test_dataset = SiameseNetworkDataset(training_csv=testing_csv, training_dir=testing_dir,
transform=transforms.Compose([transforms.Resize((105,105)),
transforms.ToTensor()])
test_dataloader = DataLoader(test_dataset, num_workers=6, batch_size=1, shuffle=True)

```

Gambar 9 Load Dataset Training

- Membuat prediksi

Langkah terakhir dari semua tahapan dalam penelitian ini adalah memprediksikan image tanda tangan dari peserta dan memberikan label *Original* ataupun *Forged*. Caranya adalah dengan membandingkan tanda tangan asli dengan tanda tangan yang baru. Jika tanda tangan baru memiliki kemiripan dengan tanda tangan asli sangat tinggi maka dinamakan Original, begitu juga sebaliknya. Beberapa cuplikan kode baris yang dilakukan untuk melakukan prediksi seperti yang ditunjukkan pada Gambar 10.

```

# Print the sample outputs to view its dissimilarity
counter=0
list_0 = torch.FloatTensor([[0]])
list_1 = torch.FloatTensor([[1]])
for i, data in enumerate(test_dataloader,0):
    x0, x1, label = data
    concatenated = torch.cat((x0,x1),0)
    output1,output2 = model(x0.to(device),x1.to(device))
    euclidian_distance = F.pairwise_distance(output1, output2)
    if label==list_0:
        label="Orginal"
    else:
        label="Forged"
    imshow(torchvision.utils.make_grid(concatenated),'Dissimilarity: {:.2f} Label:
{}'.format(euclidian_distance.item(),label))
    counter=counter+1
    if counter ==20:
        break

```

Gambar 10 Cuplikan Kode untuk Prediksi



Gambar 11 Visualisasi Hasil Prediksi

4. KESIMPULAN

Berdasarkan hasil analisa diatas maka dapat disimpulkan sebagai berikut :

1. Model yang telah dibuat telah berhasil melakukan uji similaritas sehingga dapat membedakan mana tanda tangan asli dan palsu.
2. Siamese Convolutional Neural Network sangat baik dalam melakukan modelling terhadap data gambar dan video.
3. Model dapat melakukan prediksi dari data test yang diberikan oleh pengguna

5. SARAN

1. Penelitian ini menggunakan Siamese Network yang bukan satu-satunya neural network yang terbaik untuk mendeteksi tanda tangan digital yang asli atau palsu, sehingga perbandingan neural network lain dengan deteksi *tanda tangan digital* perlu dilakukan.
2. Sebagai salah satu inovasi dalam penelitian ini perlu ditambahkan pad yang di-embed di dalam aplikasi sehingga tanda tangan digital bisa langsung dideteksi secara *realtime*.

DAFTAR PUSTAKA

- [1] O. . Abikoye, M. . Mabayoje, and R. Ajibade, "Offline Signature Recognition & Verification using Neural Network," *Int. J. Comput. Appl.*, vol. 35, no. 2, pp. 44–51, 2011.
- [2] V. Pandey and S. Shantaiya, "Signature Verification Using Morphological Features Based on Artificial Neural Network," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 2, no. 7, 2012.
- [3] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, and R. E. Howard, "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, 1989.
- [4] S. Sthapak, M. Khopade, and C. Kashid, "Artificial Neural Network Based Signature Recognition & Verification," *Int. J. Emerg. Technol. Adv. Eng.*, vol. 3, no. 8, pp. 191–197, 2013.
- [5] S. T. Kolhe and S. E. Pawar, "Offline Signature Verification Using Neural Network," *Int. J. Mod. Eng. Res.*, vol. 2, no. 3, pp. 1171–1175, 2012.

-
- [6] V. Iranmanesh, S. M. S. Ahmad, W. A. W. Adnan, S. Yussof, O. A. Arigbabu, and F. L. Malallah, "Online Handwritten Signature Verification Using Neural Network Classifier Based on Principal Component Analysis," *Sci. World J.*, pp. 1–8, 2014.
 - [7] M. Suganthy and P. Ramamoorthy, "Principal component analysis based feature extraction, morphological edge detection and localization for fast iris recognition," *J. Comput. Sci.*, vol. 8, no. 9, pp. 1428–1433, 2012.
 - [8] M. Hanmandlu, A. B. Sronothara, and S. Vasikarla, "Deep Learning based Offline Signature Verification," in *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, 2018, pp. 732–737.
 - [9] M. M. Yapıcı, A. Tekerek, and N. Topaloglu, "Convolutional Neural Network Based Offline Signature Verification Application," in *International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism*, 2018, pp. 30–34.
 - [10] R. D. Raj and J. . Lather, "Handwritten Signature Verification using TensorFlow," in *IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT-2018)*, 2018, pp. 2012–2015.
 - [11] A. Singh and S. Viriri, "Online Signature Verification using Deep Descriptors," in *2020 Conference on Information Communications Technology and Society (ICTAS)*, 2020, pp. 1–6.
 - [12] H. Ide and T. Kurita, "Improvement of learning for CNN with ReLU activation by sparse regularization," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 2684–2691.
 - [13] S. Tayeb, M. Pirouz, B. Cozzens, R. Huang, M. Jay, and Kyle, "Toward data quality analytics in signature verification using a convolutional neural network," in *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 2644–2651.
 - [14] A. M. S. Chowdhury and M. S. Rahman, "Towards optimal shallow ANN for recognizing isolated handwritten Bengali numerals," in *2016 9th International Conference on Electrical and Computer Engineering (ICECE)*, 2016, pp. 194–197.
-