International Journal of Scientific Research in Science, Engineering and Technology

Print ISSN - 2395-1990
Online ISSN : 2394-4099

OPEN ACCESS

Available Online at : www.ijsrset.com
doi : https://doi.org/10.32628/IJSRSET2310612

Signature Verification using ResNet-50 Model

Deepali Narwade, Vaishali Kolhe

Department of Computer Engineering, D. Y. Patil College of Engineering, Akurdi, Pune, Maharashtra, India

ARTICLEINFO

Article History :

Accepted: 01 Dec 2023
Published: 09 Dec 2023

Publication Issue :
Volume 10, Issue 6
November-December-2023

Page Number :
278-289

ABSTRACT

Signed documents are widely accepted as a means of confirming identification, which offers signature verification systems a major advantage over other kinds of technologies. There are two types of approaches to solving this issue using a signature verification system: online and offline. Offline signature verification uses less electronic administration and uses recorded signature images from a camera or scanner. An offline signature verification method uses extracted features from the scanned signature image. This study's primary contribution is the understanding of how deep learning network ResNet-50 can be applied to offline signature verification systems.

This paper proposes the use of ResNet-50 for offline signature verification. One kind of pretrained model that enables us to extract higher representations for the image content is called ResNet-50. CNN trained the model using the raw pixel data from the image, then automatically extracted the features for improved categorization. ResNet-50's primary advantage over its predecessors is that it has the highest accuracy of all image prediction algorithms and can automatically identify essential characteristics without human supervision. The accuracy of the ResNet-50 model was 75.8%, indicating good performance.

Keywords : Deep Learning, CNNs, ResNet-50 Model, Signature Verification.

I. INTRODUCTION

The necessity for effective automated signature verification solutions has grown since signatures serve as the main means of authorization and authenticity in legal transactions. In contrast to identity data such as passwords, PINs, PKIs, or key cards, which can be misplaced, stolen, or exchanged, the values obtained from a handwritten signature are specific to a single person and nearly impossible to replicate. The main benefit of signature verification systems over other technologies is that people already accept signatures as a standard means of confirming identification.

The process of automatically confirming a signature to ascertain its authenticity is known as signature verification. Offline (static) and offline (dynamic) signature verification are the two primary types. Verifying a document signature after it has been created is known as static, or offline verification; in contrast, dynamic, or online, verification occurs while a person produces their signature on a digital tablet or other comparable device. Subsequently, the disputed signature is contrasted with earlier examples of that individual's signature, which established the database. A digital signature that is already saved in a data format can be utilized for investigations, while a handwritten signature on a document requires the computer to scan the samples for analysis. One of the most widely recognized personal characteristics for identification verification, whether for banking or business, is a handwritten signature.

Deeper neural networks are more difficult to train. Here presented a residual learning framework to ease the training of networks that are substantially deeper than those used previously. Here explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. It provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth [1].

Neural networks (NNs) are widely used in pattern recognition due in large part to their ease of use and power. First, using many samples from various signers, extract a feature set (details like length, height, duration, etc.) defining the signature. The network can be shown test signatures that can be identified as coming from a certain signer once this relationship has been learnt.

Thus, NNs are very well adapted to simulating the overall characteristics of handwritten signatures. New trends based on the use of RNNs, which is a specific neural network architecture, are becoming more and more important nowadays for modelling sequential data with arbitrary length [9].

The primary advantage that signature verification systems have over other type's technologies is that signatures are already accepted as the common method of identity verification [11].

Features retrieved from the scanned signature image are used in an off-line signature verification method. For offline signature verification, substantially less complicated characteristics are used. Only the pixel picture needs to be analyzed in this case. However, designing off-line systems is challenging because they lack several desirable qualities, such as the sequence of strokes, velocity, and other dynamic data. The features that can be retrieved from the trace of the static signature images must be the sole basis for the verification process. Different technologies have been applied to Handwritten Signature Verification (HSV), particularly offline HSV, and the field is continuously being researched [12].

A simple approach is to firstly extract a feature set representing the signature (details like length, height, duration, etc.), with several samples from different signers. The second step is for the NN to learn the relationship between a signature and its class (either "genuine" or "forgery"). Once this relationship has been learned, the network can be presented with test signatures that can be classified as belonging to a particular signer [13].

In offline signature verification, template matching and Hidden Markov model techniques are generally employed. These techniques based on the structure of the signature. Template matching is a technique in digital image processing for fine small parts of an image which match a template image. When it comes to template matching, metrics like n square error or structural similarity index, or a warping method can be used which warps one curve onto another that the original shape is maintained. Here show the use of Deep CNNs to identify who the signature belongs to and whether it is a forgery. This is done in a two-phase approach: writer-independent feature learning, and writer dependent classification. Working this paper simplifies this approach by treating the signatures and their forgeries as separate classes. It completely writer independent approach [14].

1. Residual Neural Network

ResNet, which was proposed in 2015 by Microsoft Research experts, offered a novel architecture known as Residual Network.

A Residual Neural Network (also known as ResNet) is a deep learning model in which the weight layers learn residual functions based on the layer inputs. A Residual Network is a network with skip connections that execute identity mappings and are added to layer outputs.

Residual Network (ResNet) is a deep learning model used in computer vision. It's a Convolutional Neural Network (CNN) architecture with hundreds or thousands of convolutional layers.

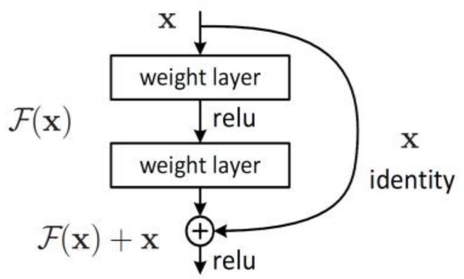


Figure 1: ResNet-50 Model [1]

ResNet, short for Residual Networks, is a traditional neural network that serves as the foundation for many computer vision applications. In 2015, this model won the ImageNet challenge. The fundamental innovation with ResNet was that it enabled us to successfully train incredibly deep neural networks with 150+ layers.

All convolutional layers in ResNet models use the same convolutional window of size 3*3. The number of filters rises with network depth, from 64 to 512 (for ResNet-18 and ResNet-34), and from 64 to 2048 (for ResNet-50, ResNet-101, and ResNet-152).

The Residual Network (ResNet) architecture is a type of artificial neural network in which the model can skip layers without compromising performance.

To handle a complex problem, here usually stack some additional layers in Deep Neural Networks, which improves accuracy and performance.

The reasoning behind layering on more layers is that they will eventually pick up more sophisticated traits. Convolutional neural networks have a classic in ResNet, no doubt about it. There are now efficient architectures like the EfficientNet. even mobile device-optimized networks, such the MobileNet design.

2. Convolutional Neural Networks (CNNs) are a class of deep learning models primarily designed for processing and analyzing visual data, such as images and videos. They have revolutionized various fields, including computer vision, image recognition, and even natural language processing. CNNs are inspired by the human visual system and are capable of learning hierarchical patterns and features from raw data.

A convolutional neural network (CNN) is a deep neural network, defined with a special operation termed as convolution operation in which a kernel is convolved over an input to generate feature map. Generally, the architecture of CNN is designed as a sequence of four basic layers, which are defined as follows.

1. **Convolution layer:** This layer performs the convolutional operation wherein an input image is convolved with a learnable kernel also known as a filter where each neuron processes data from its receptive field by taking the dot product between the kernel values and image pixels. The summation of the dot product output produces defines the feature map.

2. **Activation Layer:** This layer introduces non-linearity in the system by mapping the generated feature map into non-linear values. The popular non-linearity functions are rectified linear unit (ReLu), tanh, and sigmoid functions. Generally, this layer is applied in combination with the convolutional layer.

3. **Pooling layer:** This layer extracts important and relevant features of an image by performing down-sampling over the output of the activation layer. The essence of pooling layer is to identify the existence of a feature rather than its exact location. Additionally, this benefits in reducing the spatial dimensionality of processed input which further lowers the computational complexity. Max-pooling is the most common pooling layer.

4. **Drop-out Layer:** To reduce the overfitting in the model, this layer assigns a value zero to a random set of activation values from the previous layer. This generalizes the model over the training data.

5. **Fully connected:** In a fully connected layer, every neuron in the previously hidden layer is connected to every neuron of this layer. This layer is conventional to the hidden of the simple artificial neural network.

Generally, CNNs have been successfully applied in many computer vision applications such as object recognition, image analysis, Image colorization, Scene Labeling, etc. Influence of convolutional neural networks started with the introduction of Alexnet by Alex in the ImageNet competition where it achieved the error rate of 15.3%, far below the competing models [2].

3. Benefits of CNNs

1. **Feature Hierarchies:**

CNNs automatically learn hierarchies of features from the data. Lower layers capture basic features like edges and corners, while deeper layers learn more complex features.

2. **Translation Invariance:**

CNNs are invariant to translations in the input data due to the shared weights in convolutional layers. This property makes them well-suited for tasks like image recognition.

3. **Parameter Sharing:**

CNNs use parameter sharing, meaning the same set of weights is applied to different parts of the input. This reduces the number of parameters and makes the model more efficient.

4. Applications of CNNs

1. **Image Classification:**

CNNs are widely used for image classification tasks, where the goal is to assign a label to an image from a predefined set of categories.

2. **Object Detection:**

CNNs can detect and localize objects within an image, which is crucial in applications like self-driving cars, surveillance, and robotics.

3. **Segmentation:**

CNNs can segment an image into different regions, assigning labels to each pixel. This is useful for tasks like medical image analysis and scene understanding.

4. **Style Transfer:**

CNNs can transfer artistic styles from one image to another, enabling the creation of visually appealing images.

5.Convolutional Neural Network (CNN) Architecture

A Convolutional Neural Network (CNN) architecture is designed specifically for processing grid-like data, such as images. Its architecture is characterized by several layers that work together to learn features from the input data and make predictions. Here's an overview of the typical layers found in a CNN:

1. Input Layer:

The input layer accepts the raw data, which is usually an image or a set of images. Images are represented as grids of pixels, with each pixel containing color information.

2. Convolutional Layer:

Convolutional layers are the core building blocks of CNNs. They consist of a set of learnable filters (kernels) that slide over the input data, extracting relevant features. Convolutional operations involve element-wise multiplication of the filter with a local region of the input data, followed by summation. This process results in feature maps that represent the presence of specific features in different spatial locations.

3. Activation Layer:

Activation functions (such as ReLu) introduce non-linearity to the network. They apply a mathematical function to each element of the feature maps, enabling the network to capture complex relationships within the data.

4. Pooling Layer:

Pooling layers down sample the spatial dimensions of the feature maps while retaining the most important information. Max pooling and average pooling are common techniques. Pooling helps reduce the computational load and makes the network more invariant to small variations in the input.

5. Fully Connected Layer:

Fully connected layers connect every neuron in the current layer to every neuron in the subsequent layer. These layers process the high-level features learned by the convolutional and pooling layers and produce final predictions or classifications.

6. Output Layer:

The output layer produces the final prediction based on the processed features. The number of neurons in this layer corresponds to the number of classes in a classification task.

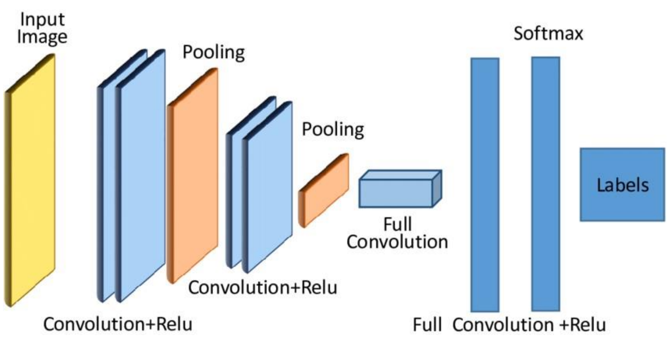


Figure 2: CNN architecture [15]

After the first CNN-based architecture (AlexNet) that win the ImageNet 2012 competition, every subsequent winning architecture uses more layers in a deep neural network to reduce the error rate. This works for less number of layers, but when increase the number of layers, there is a common problem in deep learning associated with that called the Vanishing/Exploding gradient. This causes the gradient to become 0 or too large. Thus when increase number of layers, the training and test error rate also increases.

The observations express that a 56-layer CNN gives more error rate on both training and testing datasets than a 20-layer CNN architecture. After analyzing more on error rate the authors were able to reach conclusion that it is caused by vanishing/exploding gradient.

The organization of this document is as follows. In Section 1(Introduction), Introduction and basics of neural network, ResNet-50 Model, Convolutional Neural Networks (CNNs). In Section 2 (**Literature Review**), review of techniques used for signature verification. In Section 3 (Methodology) implementation process using ResNet-50. In Section 4 (**Result and Discussion**), present results and analysis of those results. Finally, in Section 5 (**Conclusion**) concluded ResNet-50 network model with its accuracy and performance.

II. LITERATURE REVIEW

Different theories for offline signature verification are examined. Template matching and Hidden Markov model approaches are commonly utilized in offline signature verification. These methods are predicated on the signature's structure. In digital image processing, template matching is a technique used to find fine, small portions of an image that match a template image. Metrics such as the structural similarity index, n square error, or a warping technique that maintains the original shape of the curve can be applied when comparing templates. Here will see how to use Deep CNNs to determine the signature's owner and whether or not it is a fake. Four popular convolutional neural network based models namely, AlexNet, VGG16, Resnet50 and InceptionV3 are explained in this section.

1. AlexNet

AlexNet won the Imagenet large-scale visual recognition challenge in 2012. The model was proposed in 2012 in the research paper named Imagenet Classification with Deep Convolution Neural Network by Alex Krizhevsky and his colleagues.

Important points:

1. It has 8 layers with learnable parameters.
2. The input to the Model is RGB images.
3. It has 5 convolution layers with a combination of max-pooling layers.
4. Then it has 3 fully connected layers.
5. The activation function used in all layers is ReLu.
6. It used two Dropout layers.
7. The activation function used in the output layer is Softmax.
8. The total number of parameters in this architecture is 62.3 million.

AlexNet is a pioneering convolutional neural network (CNN) used primarily for image recognition and classification tasks. It won the ImageNet Large Scale Visual Recognition Challenge in 2012, marking a breakthrough in deep learning. AlexNet's architecture, with its innovative use of convolutional layers and rectified linear units (ReLu), laid the foundation for modern deep learning models, advancing computer vision and pattern recognition applications.

The AlexNet has eight layers with learnable parameters. The model consists of five layers with a combination of max pooling followed by 3 fully connected layers and they use ReLu activation in each of these layers except the output layer. They found out that using the ReLu as an activation function accelerated the speed of the training process by almost six times. They also used the dropout layers, that prevented their model from overfitting. Further, the model is trained on the Imagenet dataset. The Imagenet dataset has almost 14 million images across a thousand classes.

Architecture of AlexNet

Alexnet is a deep architecture, the authors introduced padding to prevent the size of the feature maps from reducing drastically. The input to this model is the images of size 227X227X3.

Table 1: Layers in AlexNet

Layer	# filters / neurons	Filter size	Stride	Padding	Size of feature map	Activation function
Input	-	-	-	-	227 x 227 x 3	-
Conv 1	96	11 x 11	4	-	55 x 55 x 96	ReLU
Max Pool 1	-	3 x 3	2	-	27 x 27 x 96	-
Conv 2	256	5 x 5	1	2	27 x 27 x 256	ReLU
Max Pool 2	-	3 x 3	2	-	13 x 13 x 256	-
Conv 3	384	3 x 3	1	1	13 x 13 x 384	ReLU
Conv 4	384	3 x 3	1	1	13 x 13 x 384	ReLU
Conv 5	256	3 x 3	1	1	13 x 13 x 256	ReLU
Max Pool 3	-	3 x 3	2	-	6 x 6 x 256	-
Dropout 1	rate = 0.5	-	-	-	6 x 6 x 256	-

Convolution and Maxpooling Layers

Then apply the first convolution layer with 96 filters of size 11X11 with stride 4. The activation function used in this layer is ReLu. The output feature map is 55X55X96.

In case, you are unaware of how to calculate the output size of a convolution layer

$$\text{output}= ((\text{Input-filter size})/ \text{stride})+1$$

Also, the number of filters becomes the channel in the output feature map.

Next, the first Maxpooling layer, of size 3X3 and stride 2. Then, get the resulting feature map with the size 27X27X96.

After this, apply the second convolution operation. This time the filter size is reduced to 5X5 and have 256 such filters. The stride is 1 and padding 2. The activation function used is again ReLu. Now the output size got is 27X27X256.

Again applied a max-pooling layer of size 3X3 with stride 2. The resulting feature map is of shape 13X13X256.

Now apply the third convolution operation with 384 filters of size 3X3 stride 1 and also padding 1. Again the activation function used is ReLu. The output feature map is of shape 13X13X384.

Then have the fourth convolution operation with 384 filters of size 3X3. The stride along with the padding is 1. On top of that activation function used is ReLu. Now the output size remains unchanged i.e 13X13X384.

After this, the final convolution layer of size 3X3 with 256 such filters. The stride and padding are set to one also the activation function is ReLu. The resulting feature map is of shape 13X13X256.

So if you look at the architecture till now, the number of filters is increasing as going deeper. Hence it is extracting more features as move deeper into the architecture. Also, the filter size is reducing, which means the initial filter was larger and as go ahead the filter size is decreasing, resulting in a decrease in the feature map shape.

Next, apply the third max-pooling layer of size 3X3 and stride 2. Resulting in the feature map of the shape 6X6X256.

Fully Connected and Dropout Layers

After this, our first dropout layer. The drop-out rate is set to be 0.5.

Then the first fully connected layer with a ReLu activation function. The size of the output is 4096. Next comes another dropout layer with the dropout rate fixed at 0.5.

Table 2: Fully connected and Dropout layer

Layer	# filters / neurons	Filter size	Stride	Padding	Size of feature map	Activation function
-	-	-	-	-	-	-
-	-	-	-	-	-	-
-	-	-	-	-	-	-
Dropout 1	rate = 0.5	-	-	-	6 x 6 x 256	-
Fully Connected 1	-	-	-	-	4096	ReLU
Dropout 2	rate = 0.5	-	-	-	4096	-
Fully Connected 2	-	-	-	-	4096	ReLU
Fully Connected 3	-	-	-	-	1000	Softmax

This followed by a second fully connected layer with 4096 neurons and ReLu activation.

Finally, the last fully connected layer or output layer with 1000 neurons as 10000 classes in the data set. The activation function used at this layer is Softmax.

This is the architecture of the Alexnet model. It has a total of 62.3 million learnable parameters.

2. VGGnet Model

Simonyan and Zisserman introduced the VGG model [3] in 2014. VGG model focuses on the following points. In this model, the depth of the network has been increased. To achieve this, here used 33 small convolution filters (rather than 7x7 filter used previously) in all layers. The sense behind using this configuration of 3x3 is that it is the smallest size to capture the notion of left/right, up/down, center.

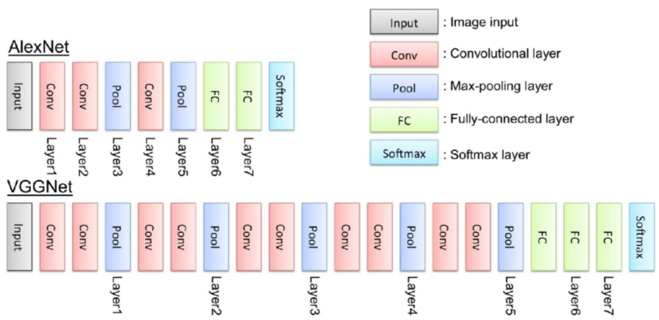


Figure 3: VGGNet & AlexNet [16]

Using a stack of smaller receptive field such as 2 layers of 3x3 provides an effective receptive field of 5x5 and stacking up 3 such layers sums up to 7x7 convolution layer. Through using a stack of 3x3 sized filters, authors have tried to achieve several advantages. The first advantage of using such configuration is that a combination of non-linear rectified layers can be used. This helps with making the decision function more discriminative. The second advantage is that by using this, the number of parameters decreases drastically.

3. Inception Model

The inception model [5] has been presented in three different versions where each successor had several advantages over the previous models. While the previous model discussed is made up of stacking layers, this model is highly complex. This engineering provides a better performance, both in speed and accuracy. The disadvantage of this complexity is that it is hard to make changes according to ourselves and if changed without caution, the model would lose its computational advantages. While it's hard to choose the kernel size for a given set of problems and have to be changed according to the problem, this model provides an advantage by using different kernel size at same level thus rather than making the network deeper it becomes wider. In the paper,here used 3 different kernel sizes 1x1, 3x3, 5x5 and a maxpooling layer at the same level forming a single module. The output would be concatenated and passed on further. As adding all these layers at the same level would make this model more computationally expensive.

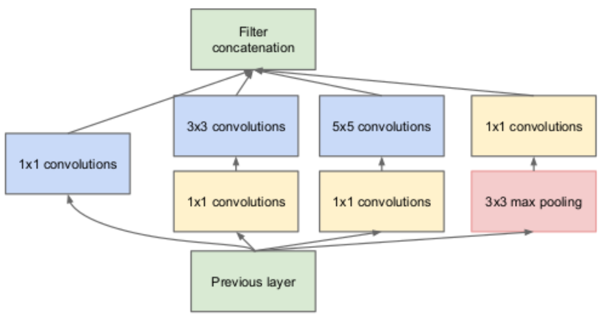


Figure 4: Inception Model [17]

To mitigate this, the authors have used a 1x1 convolution layer before 3x3 and 5x5 layer and after max pooling layer. The use of this layer is that it that 1x1 are way cheaper than 5x5 layer and would reduce the number of channels in input. The final module is depicted in figure. Since this model was 22 layers deep, so it was very much prone to vanishing gradient problem.

Inception model uses different kernel sizes at the same level which makes it better for different uses and increases its efficiency drastically. While it can be computationally very expensive, authors have found a turnaround this by using the smaller kernel to reduce the dimension of original data thus decreasing the recourses required.

III. METHODOLOGY

There are major four steps:

1. **Data Collection:** Images are collected into a dataset with specific size.

Kaggle: It is a data science competition platform. It enables users to find and publish datasets, explore and build models in a web based data science environment. Kaggle also offers public data sets, machine learning notebooks, and tutorials to help users learn and practice their skills in data science and machine learning.

In this project, the collected signatures from SignverOD Dataset (A Dataset Signature Object Detection) will be used for training purpose.

2. **Preprocessing:** The process of manipulating raw image raw into usable and meaningful format. It allows to eliminate unwanted distortions and enhance specific qualities of essential computer vision applications. Preprocessing is a crucial first step to prepare your image data before feeding it into any neural network model.

Various operations are applied to signature images.

a) **Resizing:** Resizing images to a uniform size is important for DL algorithms to function properly. OprnVC's resize() method to resize images.
b) **Grayscaling:** Converting color images to grayscale can simplify your image data and reduce computational needs for some algorithms. The cvtColor() method can be used to convert RGB to grayscale.
c) **Noise reduction:** Smoothing, blurring, and filtering techniques can be applied to remove unwanted noise from images. The

GaussianBlur () and medianBlur () methods are commonly used for this.

d) **Normalization:** Normalization adjusts the intensity values of pixels to a desired range, often between 0 to 1. This can improve the performance of machine learning models. Normalize () from scikit-image can be used for this.

e) **Binarization:** Binarization converts grayscale images to black and white by thresholding. The threshold () method is used to binarize images in OpenCV.

3. **Feature Extraction**: Feature Extraction means process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. Likewise, image data is converted in the form of array. At last, this numeric values given to classifier.

Some features of the signature are extracted in this step. These extracted features are the inputs to the training and recognition stages. The features can be categorized into global, mask, and grid features. Global features give wavelet coefficients and Fourier coefficients. Mask features give information about the signature lines' directions. Grid features give information about the overall appearance of a signature. The selection of feature sets in signature verification systems is a complicated task due to the fact that the user features must be appropriate for the application [8].

4. **Signature Verification:** This is divided into two steps:

A. Training stage

B. Testing stage

5. ResNet-50 Model

In literature, it has been proven that the performance of deep neural networks increases with the increase in the depth of model. But with an increase in depth of network arises the problem of vanishing/exploding gradients [4].

To avoid the problem of vanishing gradient, ResNet uses skipwise network. Rather than creating a simple mapping of layers, ResNet uses a modified version by using the mapping function H(x) = F(x) + I, where F(x) refers to x refers the identity function where input and output are same [5].

ResNet stands for residual network, which refers to the residual blocks that make up the architecture of the network. ResNet-50 is based on a deep residual learning framework that allows for the training of very deep networks with hundreds of layers. The ResNet architecture was developed in response to a surprising observation in deep learning research: adding more layers to a neural network was not always improving the results.

This was unexpected because adding a layer to a network should allow it to learn at least what the previous network learned, plus additional information. To address this issue, the ResNet team, led by Kaiming He, developed a novel architecture that incorporated skip connections. These connections allowed the preservation of information from earlier layers, which helped the network learn better representations of the input data. With the ResNet architecture, they were able to train networks with as many as 152 layers.

The results of ResNet were groundbreaking, achieving a 3.57% error rate on the ImageNet dataset and taking first place in several other competitions, including the ILSVRC and COCO object detection challenges.

This demonstrated the power and potential of the ResNet architecture in deep learning research and applications.

ResNet-50 Architecture

ResNet-50 consists of 50 layers that are divided into 5 blocks, each containing a set of residual blocks. The residual blocks allow for the preservation of information from earlier layers, which helps the network to learn better representations of the input data.

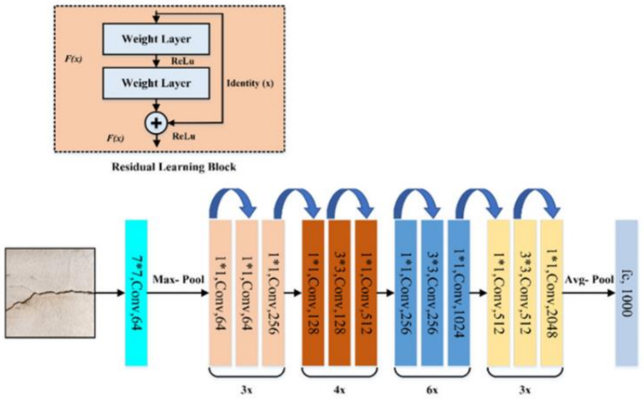


Figure 5: ResNet-50 Model [18]

The following are the main components of ResNET.

1. Convolutional Layers

The first layer of the network is a convolutional layer that performs convolution on the input image. This is followed by a max-pooling layer that downsamples the output of the convolutional layer. The output of the max-pooling layer is then passed through a series of residual blocks.

2. Residual Blocks

Each residual block consists of two convolutional layers, each followed by a batch normalization layer and a rectified linear unit (ReLu) activation function. The output of the second convolutional layer is then added to the input of the residual block, which is then passed through another ReLu activation function. The output of the residual block is then passed on to the next block.

3. Fully Connected Layer

The final layer of the network is a fully connected layer that takes the output of the last residual block and maps it to the output classes. The number of neurons in the fully connected layer is equal to the number of output classes.

Concept of Skip Connection

Skip connections, also known as identity connections, are a key feature of ResNet-50.

They allow for the preservation of information from earlier layers, which helps the network to learn better representations of the input data.

Skip connections are implemented by adding the output of an earlier layer to the output of a later layer.

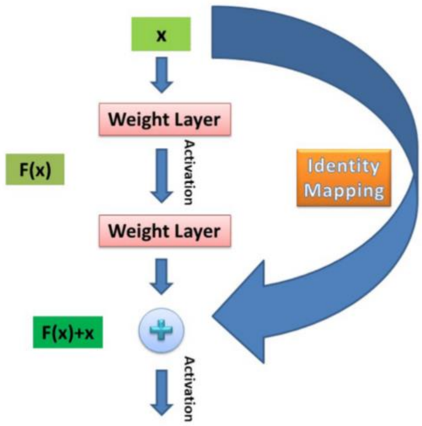


Figure 6: Skip Connection [18]

Key Features of ResNet-50

1. 152 layer model for ImageNet
2. Has other variants also (with 35, 50, 101 layers)
3. Every 'residual block' has two 3×3 convolution layers
4. No FC layer, except one last 1000 FC softmax layer for classification.
5. Global average pooling layer after the last convolution.
6. Batch Normalization after every convolution layer
7. No dropout used

Advantages of ResNet-50 Over Other Networks

ResNet-50 has several advantages over other networks. One of the main advantages is its ability to train very deep networks with hundreds of layers.

This is made possible by the use of residual blocks and skip connections, which allow for the preservation of information from earlier layers.

Another advantage of ResNet-50 is its ability to achieve state-of-the-art results in a wide range of image-related tasks such as object detection, image classification, and image segmentation.

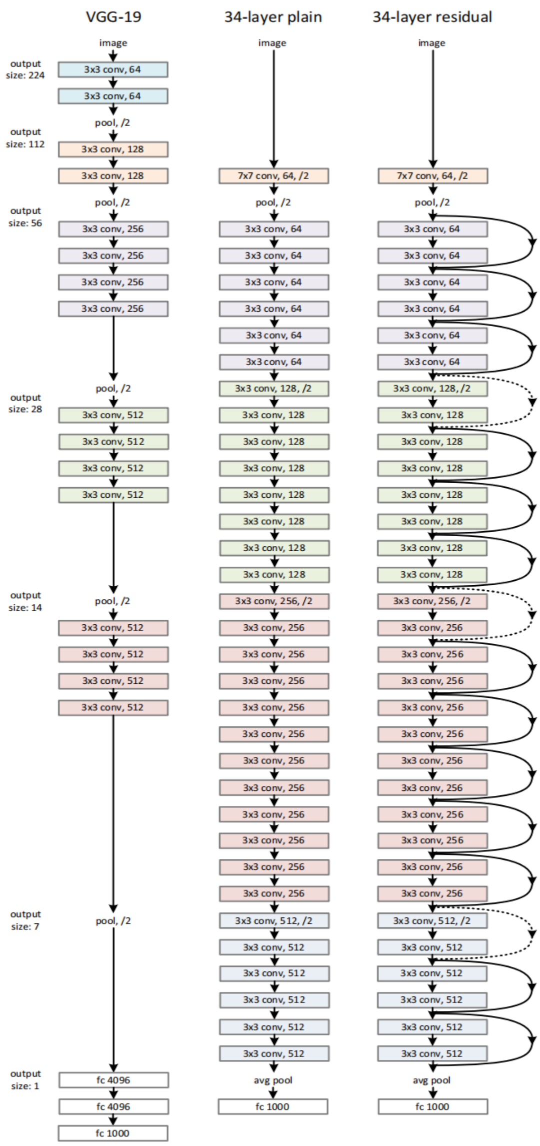


Figure 7: ResNet-50 Architecture [1]

The combination of these skipwise blocks is used to give out a bigger network which prevents the problem of vanishing and blasting gradients [6].

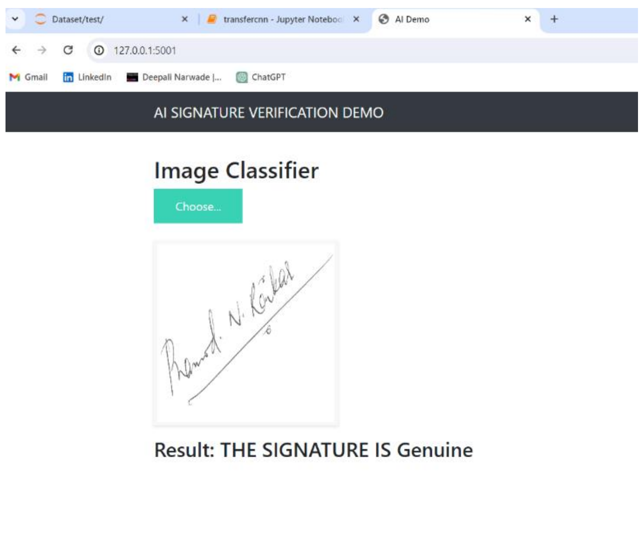
MobileNets are based on a streamlined architecture that uses depth-wise separable convolutions to build light weight deep neural networks. We introduce two simple global hyper-parameters that efficiently trade off between latency and accuracy. These hyper-parameters allow the model builder to choose the right sized model for their application based on the constraints of the problem. We present extensive experiments on resource and accuracy tradeoffs and show strong performance compared to other popular models on ImageNet classification. We then demonstrate the effectiveness of MobileNets across a wide range of applications and use cases including object detection, finegrain classification, face attributes and large scale geo-localization [7].

Machine Learning and Deep Learning concepts applied for creation Signature Verification System. The majority of places perform manual verification, which might be trouble some at times. With this project implementation the manual work of verifying the signature will get reduced [10].

IV. RESULTS AND DISCUSSION

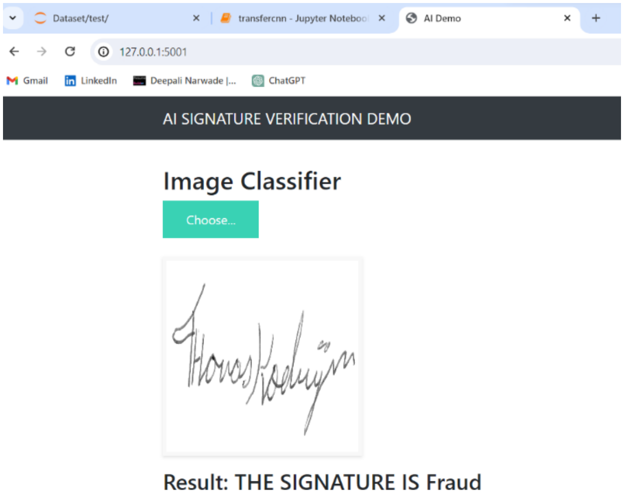
Gathering data is the first and most important stage. The signatures gathered from the SignverOD Dataset (A Dataset Signature Object Detection) from Kaggle will be used for training the datasets.

For this project, all of the signature photos were initially trained. These images have been trained and tested and are kept in the training and testing folder. Next, the image data is transformed into an array. Finally, the classifier receives these numerical values. The retrieved image is then used to train the classifier. The model will now compare the attributes of the input signature with the features of the authentic signature, which is already included in the data set.



Snapshot 1: Output 1

If both signature's features matched, then model will give the output: "THE SIGNATURE IS Genuine" if the image is authentic, otherwise " THE SIGNATURE IS Fraud" if it is fake.



Snapshot 2: Output 2

1. ReLu activation function

A rectified linear unit (ReLu) is an activation function that introduces the property of nonlinearity to a deep learning model and solves the vanishing gradients issue. Here's why it's so popular. Written by Bharath Krishnamurthy. Published on Oct. 28, 2022.

ResNet has VGG's full 3 × 3 convolutional layer design. The residual block has two 3 × 3 convolutional layers with the same number of output channels. Each convolutional layer is followed by a batch normalization layer and a ReLu activation function.

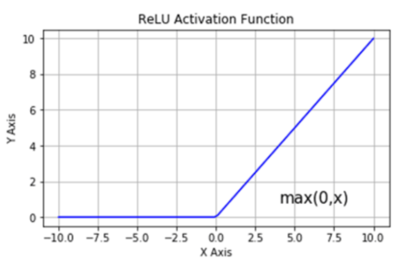


Figure 8: ReLu Function [19]

Each residual block consists of two convolutional layers, each followed by a batch normalization layer and a rectified linear unit (ReLu) activation function. The output of the second convolutional layer is then added to the input of the residual block, which is then passed through another ReLu activation function. The output of the residual block is then passed on to the next block.

2. Accuracy and Loss Function:

A confusion matrix is a performance evaluation tool in machine learning, representing the accuracy of a classification model. It displays the number of true positives, true negatives, false positives, and false negatives. This matrix aids in analyzing model performance, identifying mis-classifications, and improving predictive accuracy.

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the total number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

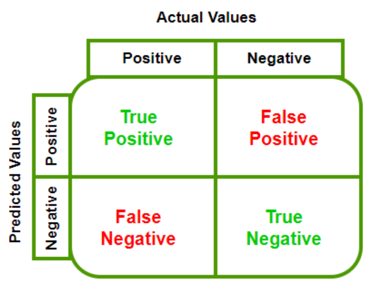


Figure 9: Confusion Matrix [20]

Important Terms in a Confusion Matrix

1. True Positive (TP)

The predicted value matches the actual value, or the predicted class matches the actual class.

The actual value was positive, and the model predicted a positive value.

2. True Negative (TN)

The predicted value matches the actual value, or the predicted class matches the actual class.

The actual value was negative, and the model predicted a negative value.

3. False Positive (FP) – Type I Error

The predicted value was falsely predicted.

The actual value was negative, but the model predicted a positive value.

Also known as the type I error.

4. False Negative (FN) – Type II Error

The predicted value was falsely predicted.

The actual value was positive, but the model predicted a negative value.

Also known as the type II error.

$$\text{Accuracy} \Rightarrow \frac{(TN+TP)}{(TN+FP+FN+TP)}$$

$$\text{Precision} \Rightarrow \frac{(TP)}{(TP+FP)}$$

$$\text{Recall} \Rightarrow \frac{(TP)}{(TP+FN)}$$

$$\text{F1-Score} \Rightarrow 2 \text{ x } \frac{(TP)}{(TP+FP)}$$

Table III: EXPERIMENTAL RESULT

Model Metric	**ResNet50**
Accuracy	75.8 %
Precision	76.5%
Recall	75.8%
F1-Score	76.2

Mean Squared Error (MSE) measures the amount of error in a statistical model. Evaluate the mean squared difference between observed and predicted values. If the model has no errors, the MSE is zero. Its value increases as the model error increases.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Where,

n = number of data points

Yi = Observed values

^Yi = Predicted values

V. CONCLUSION

ResNet-50 is a powerful and efficient deep neural network that has revolutionized the field of computer vision. Its architecture, skip connections, and advantages over other networks make it an ideal choice for a wide range of image-related tasks. The ResNet-50 consisted of A 7×7 kernel convolution, A max pooling layer, 9 more layers—3×3,64 kernel convolution and an output layer with 2 neurons representing genuine or forged signatures. Also, used a ReLu activation function after the first and last layers and the mean squared difference between the logits and targets as the loss function. The Adam optimizer was used to minimize the loss function. The ResNet-50 model performed well, achieving an accuracy of 75.8%.

VI.REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition”,IEEE 2016.

[2] Tejasv Agarwal, Himanshu Mittal, “Performance Comparison Of Deep Neural Networks On Image Datasets”, IEEE 2019.

[3] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[5] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, ser. Proceedings of Machine Learning Research, Y. W. Teh and M. Titterington, Eds., vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, pp. 249–256. [Online]. Available: http://proceedings.mlr.press/v9/glorot10a.html

[6] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” IEEE Transactions on Neural Networks, vol. 5, no. 2, pp. 157–166, March 1994.

[7] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” arXiv preprint arXiv:1704.04861, 2017.

[8]

[9] Deepali Narwade, Vaishali Kolhe, “A Survey: Deep Learning Approaches for Signature Verification”, IJSRSET 2023.

[10] Ruben Tolosana , Ruben Vera-Rodriguez , Julian Fierrez , Javier Ortega-Garcia, “DeepSign: Deep On-Line Signature Verification”, IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR, AND IDENTITY SCIENCE, VOL. 3, NO. 2, APRIL 2021.

[11] Yash Borse, Anjali Patil, Sumeet Shah, Anand Gharu, “Signature Verification Using Deep Learning”, 2023 IJCRT.

[12] Nakshita Pramod Kinhikar Dr.K.N. Kasat, “Offline Signature Verification using Python”, 2022 IJCRT.

[13] Kesana Mohana Lakshmi and Tummala Ranga Babu, “An Efficient Algorithm For Hand Written Signature Recognition Using Transform Based Approach With Image Statistics”, 2020.

[14] Pallavi V. Hatkar, Zareen J Tamboli, “Image Processing for Signature Verification”, International Journal of Innovative Research in Computer Science & Technology (IJIRCST) ISSN: 2347-5552, Volume-3, Issue-3, May- 2015.

[15] B.Akhila, G.Nikhila, A Lakshmi, G.Jahnavi, Mrs .J.Himabindhu, “Signature Verification Using Image Processing And Neural Networks”, 2021 IJCRT.

[16] Image Source: https://www.researchgate.net/figure/A-generic-CNN-Architecture_fig1_344294512

[17] https://www.researchgate.net/figure/AlexNet-and-VGGNet-architecture_fig1_282270749

[18] https://www.analyticsvidhya.com/blog/2018/10/understanding-inception-network-from-scratch/

[19] https://wisdomml.in/understanding-resnet-50-in-depth-architecture-skip-connections-and-advantages-over-other-networks/

[20] https://www.researchgate.net/figure/s-a-graphical-representation-of-the-ReLu-function-Equation-1-which-produces-the-same_fig4_369791899

[21] https://www.geeksforgeeks.org/visualize-confusion-matrix-using-caret-package-in-r/

Cite this article as :

Deepali Narwade, Vaishali Kolhe "Signature Verification using ResNet-50 Model", International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), Online ISSN : 2394-4099, Print ISSN : 2395-1990, Volume 10 Issue 6, pp. 278-291, November-December 2023. Available at doi : https://doi.org/10.32628/IJSRSET2310612
Journal URL : https://ijsrset.com/IJSRSET2310612