HANDWRITTEN SIGNATURE VERIFICATION

A PROJECT REPORT

Submitted by

SHANA PARWEEN (TKM20MCA2036)

to

The APJ Abdul Kalam Technological University

In partial fulfillment of the requirements for the award of the Degree of

MASTER OF COMPUTER APPLICATIONS

Thangal Kunju Musaliar College of Engineering
Kerala
DEPARTMENT OF COMPUTER APPLICATIONS

JULY 2022

DECLARATION

I undersigned hereby declare that the project report "HANDWRITTEN SIGNATURE VERIFICATION" , submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of Prof. Alshaina S. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kollam

Date: 18-07-2022

Shana Parween

DEPARTMENT OF COMPUTER APPLICATIONS

TKM COLLEGE OF ENGINEERING

C E R T I F I C A T E

This is to certify that, the project report entitled **" HANDWRITTEN SIGNATURE VERIFICATION "** is submitted by **SHANA PARWEEN (TKM20MCA-2036)** to the APJ Abdul Kalam Technological University in partial fulfillment of the requirements for the award of the degree of Master of Computer Applications, is a bonafide record of the project work carried out by her under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

Internal Supervisor Head of the Department External Examiner

ACKNOWLEDGEMENT

First and foremost I thank GOD almighty and my parents for the success of this project. I owe sincere gratitude and heart full thanks to everyone who shared their precious time and knowledge for the successful completion of my project.

I am extremely grateful to **Dr. Fousia M Shamsudeen**, Head of the Department, for providing me with best facilities.

I would like to thank my project guide **Prof. Alshaina S**, Department of Computer Applications, who motivated me throughout the project.

I profusely thank all other faculty members in the department and all other members of TKM College of Engineering, for their guidance and inspirations throughout my course of study.

I owe my thanks to my friends and all others who have directly or indirectly helped us in the successful completion of this project.

Shana Parween

ABSTRACT

As a consequence of the rapid development of computer science and information technology in many organisations, institutions, banks, and online businesses, for example, the demand for a person's identification is increasing on a daily basis and is expected to continue doing so in the foreseeable future. On the other hand, utilising technology for identification and signature verification allows for the routine detection of fraudulent activity and forged signatures. Since the beginning of the biometrics era, authenticity and verification of signatures have been vitally significant aspects. In intricate forgeries, such as when a forger possesses a person's signature and intentionally copies it, it may be difficult to establish a person's identity using a handwritten autograph. This may happen if a forger possesses the person's signature.Offline (static) signature verification loses dynamic information, making it difficult to create feature extractors. Offline verification is static. Because to this, using offline signature verification is much more difficult. The end consequence is a performance that is below average. It is my proposal that convolutional neural networks be used in order to train representations from signature photos in a way that is independent of the writer. This will make it possible for you to satisfy the demands of gaining the essential features while also boosting the system's general performance, which is a win-win situation. I present an innovative formulation of the issue that includes data from competent forgeries to boost feature learning. I can capture visual signals that differentiate real signatures from forgeries, no matter who signs the paper.

Contents

List of Figures

List of Tables

Chapter 1

INTRODUCTION

Advanced computerised identification systems are continually being created as a direct consequence of the exponential development in the usage of information technology and approaches based on machine learning in a range of industries. Character recognition, automatic translation, search engine optimization, identification of Parkinson's disease and Alzheimer's disease, visual salience prediction in natural video, identification of writers, classification of proteins, diagnosis of Parkinson's disease and Alzheimer's disease, verification of signatures, and search engines are some examples of these kinds of systems. The process of identifying people involves determining who they are based on either their outward appearance (such as veins, iris identification, face recognition, or DNA, among other things) or their behavioural features (such as gait, signature, voice, keystroke, among others). These characteristics are inextricably linked to a person's identity and cannot be removed, substituted, or ignored under any circumstance. Handwritten signatures are the most frequent method of biometric identification in government, academic, and financial organisations.

Despite the fact that it is not a newly discussed topic in academic circles, the identification and verification of automated handwritten signatures remains the most challenging and urgent challenge in the field of forensic science. More research on this topic is required. Biometric technology used in banks and other organisations verify a person's signature. The process of determining who signed a document may be accomplished via the use of signature recognition. Signature recognition systems consist of two parts: identification and verification. These two parts are what make up the whole system. A search of signature databases is performed by signature identification soft-

ware in order to accurately identify the signer. On the other hand, signature verification refers to the process of determining whether or not the signatures on a certain sample are authentic or forgeries. Influential enterprises, organisations, and institutions, as well as banks, intelligence agencies, governments, and other valuable marketplaces, must build a credible signature verification system. The signature verification system validates a person's identity by checking their scanned signature.

Live (dynamic) and offline (static) verification systems are the two distinct varieties of this kind of technology. A static, or offline, system records signatures on paper and scans them to retrieve the data. On mobile devices such as tablets and personal digital assistants, a mechanism that is dynamic is employed to record users' signatures. Dynamic information includes pressure, pen position, angle, and other parameters. Dynamic information on devices makes improving feature representation more difficult. The use of handwritten signatures is common place in almost all organisations dealing with finances and legal matters. However, since people's mental and physical conditions may have an effect on how they sign, there is a significant problem with signature verification because individuals do not always sign in the same manner. This makes it difficult to determine whether or not a signature is genuine. The two sorts of signatures that exist are ones that are authentic (real) and ones that have been forged (faked). Falsified data just reflects a copy of the original, but authentic signatures show persons to be who they say they are, and fake signatures are just copies. The verification system has access to a wide variety of forged signatures as a result of the process of data duplication. Simple, random, competent, unskilled, opposite-handed, or freehand. Finding a professional forger will be challenging.

Despite a lengthy history, automated signature verification is presently the most popular study subject. This is due to the fact that automated signature verification is much easier to implement. As a direct consequence of this, a wide variety of techniques have already been put to use for the purpose of verification. These techniques include neural networks, support vector machines, hidden Morkov models, evolutionary algorithms, Euclidean distance, k-nearest neighbours, and many more. In a manner comparable to this, geometric properties, in addition to local and global aspects, were extracted from the relevant literature. Many fields of image processing have resorted to deep convolutional neural networks to encode observable characteristics.

1.1 Problem Definition

In this project, I used deep learning algorithms such as VGG16, Inception-v3, ResNet50, and Xception with Adam as the optimizer to compare the accuracies of handwritten signatures to determine whether they were real or fake.

1.2 Objective

The project's major objective is:

- Comparing a signature to a reference signature to verify it.
- If there is a ready dataset of the signer's authentic and a sample of faked signatures, it can detect a fake signature in connection to the signer.
- To verify a person's identification by validating the supplied scanned sample.

Chapter 2

LITERATURE SURVEY

Literature review is the broad study and apprehension of literature that relates to a certain topic. When one utilizes literature review research questions are recognized, then one seek to answer this research questions by penetrating for and examining relevant literature. Some significance of literature reviews is that new perceptions can be developed by re-considering the results of the study.

A literature review summarises and explains the whole and most recent body of information about a subject that may be found in scholarly books and journal articles. You might write one of two different types of literature reviews: one that serves as a standalone project for a class, and the other that serves as preliminary work for longer pieces like theses or research reports. Depending on the type of review you are writing, your focus, attitude, and the style of hypothesis argument you present will all be determined. Reading published literature reviews or the introductory sections of theses and dissertations in your own field of study will help you to understand how these two forms differ from one another. Examine the manner they address the issues and the organisation of their arguments.

2.1 Purpose of the Literature Review

1. By selecting superior articles or studies that are applicable, meaningful, significant, vital, or valid and compiling them into a single report, it makes it simple for readers to conduct research on a certain subject.

2. By forcing them to summarise, estimate, and compare original research in that particular area, it provides an excellent starting point for researchers starting to perform study in a new field.

3. It ensures that researchers do not combine already completed work.

4. It can suggest areas for future research to focus on or provide indications as to where it might go.

5. It notes gaps, contradictions, and disputes in the literature.

6. It offers an insightful analysis of the methods and strategies used by other researchers.

2.2 Related Works

2.2.1 DeepSignature: fine-tuned transfer learning based signature verification system [1]

The demand for person authentication is increasing daily as a result of the quick development of technology and knowledge technology in many organisations, institutions, banks, or internet trade, etc. Similar to this, technologies for signature verification and identification are commonly used in fraud and forgery detection. For a long time, signature authentication and verification have been essential biometric properties.This article compares CNN architectures AlexNet, GoogleNet, ResNet, MobileNet, and DenseNet. To better support and recognise offline photos of signatures, a better transfer learning-based approach is being researched. Persian signatures from various people were used in the assessment, with the UTSig dataset serving as the criterion. The results of the experimental research demonstrate how superior the DenseNet architecture is to the other CNN architectures.

The most important challenge for security concerns is to prevent document falsification in a range of legal, financial, government, and other lucrative situations. Despite countless attempts, there are still many provocations in this area. The task of signature detection and verification was given to the AlexNet, GoogleNet, ResNet, MobileNet, and DenseNet architectures in this study.

They combined specialised feature extraction techniques with the DenseNet classifier for the classification and verification task using the UTSig dataset, and they found that it outperformed cutting-edge techniques. This system obtained an accuracy of 98.87% by using a fine-tuned method. This deep learner classifier has the advantage of being able to intuitively extract hidden features from signature photos, significantly improving the classifier's performance.

2.2.2 Online signature verification using signature down-sampling and signer-dependent sampling frequency [2]

Online signature verification approaches see signatures as time sequences comprising numerous computations. Digital technology was utilised to collect these signals, thus each includes samples. Several verifiers use resampling and intercalation as preprocessing steps to enhance or explain data. This study examines the direct influence of input signature sample rate on online signature verification systems without interpolation. It provides a unique online signature verification approach based on signer-dependent sample frequency. Twenty verifier arrangements were constructed for five signature databases, multiple preprocessing approaches, and 20–40 sample rates. They found that there is a suitable range for the quantity of sample points and sampling frequency that lowers the error rate of a verifier. They found that evaluations with sample frequencies between 15 and 50 Hz and signature point counts between 60 and 240 were the most accurate. Results with shorter spans were wrong, as one would assume, but it's fascinating to find that higher frequencies typically had worse verification accuracy. The findings show that by downsampling online signatures before further processing, better or at least equivalent verification accuracies may be acquired more quickly. By choosing the optimal sample frequency, our technique produced results similar to state-of-the-art systems. A technique for signature verification based on sample frequencies particular to each signer was also described, along with an analysis of the results of choosing separate sampling frequencies for each signer. When testing 500 various verification techniques, this process enhanced accuracy in 92

This study examined the accuracy of online signature verification systems in relation to the number sample points and sampling rate of input devices. Online signature verification should use DTW and signer-dependent sample frequency. Several DTW-based verification system configura-

tions at different sample rates were used to analyse EER. The 2800 trials allowed researchers to make findings independent of other variables that may have affected the system's accuracy. They haven't done research on online signature verification.

2.2.3 Off-line signature verification using elementary combinations of directional codes from boundary pixels [3]

Formal documents such as bank checks, certifications, contract forms, and bonds are hard to verify. Validity is based on how closely the signature on the documents matches the original. Signatures of authorised parties are presumed. This work offers a signature verification mechanism based on border pixel runs' quasi-straightness. First, they use simple permutations of the directional codes to separate the signature border pixels from the quasi-straight line segments. Semi-straight line segments, which mix slight curvatures with straightness, are good for signature verification. They used SVM to classify well-known signature datasets as GPDS-100 and CEDAR (Grupo de Procesado Digital de la Senal). The results indicate how well this strategy works.

This work proposes a novel feature set for signature verification based on the signature-boundary-edge picture pixel distribution. Previous attempts have chain coded or determined the longest pixel run to integrate directional features with other features. In this study, the distinguishing characteristics were determined using newly proposed classes of quasi-straight line segments. They tested utilising well-known datasets like the GPDS-100 and CEDAR. The studies' findings show that the suggested feature set produces noticeably good outcomes and might be useful in real-time applications. Convex hull form can be combined with other geometric factors, such as the number and location of endpoints, the number of closed loops, etc., to improve accuracy. The significance of the edge length threshold parameter for the definition and categorization of features must be highlighted. Therefore, it will be useful to investigate the automatic selection of the edge length threshold. The resolution of the signature images might have an impact on this criterion. It's also crucial to determine whether the local curvature information can be detected by comparing the non-singular code values, singular code values, and non-singular run lengths of two neighbouring quasi-straight line segments. The type and size of the curve can play a big part in how well something is recognised. The fact that the curvature information may be recovered without requiring

any further computation should be emphasised. This curvature information may help improve the model's accuracy.

2.2.4 Characterizing and evaluating adversarial examples for Offline Handwritten Signature Verification [4]

Hostile instances are gaining popularity in machine learning because they affect system security. Perceptually, adversarial examples resemble data distribution samples that "trick" machine learning classifiers. These incorrectly identified photographs have been expertly and almost imperceptibly edited for machine vision software. Using a well-established taxonomy of biometric system risks, they describe this behaviour in their research and disclose brand-new attacks against Offline Handwritten Signature Verification systems in particular.

They tested CNN-based and bespoke feature extractors on MCYT-75, CEDAR, GPDS-160, and the Brazilian PUC-PR datasets (CLBP). Even without access to the trained classifier or training signatures, attacks to reject a valid signature are easy to create. Object identification study shows that attacks that validate a deception are harder, entail more noise, and are no longer "invisible." They also investigated the influence of two defences on assault success rates and required noise volume.

This work focuses on adversarial machine learning issues in offline handwritten signature verification. They also examined the effects of adversarial situations on biometric systems. They demonstrate that in a number of situations, including those using systems that directly learn from picture pixels and those utilising specialised feature extractors, hostile scenarios provide new challenges for such systems. Type-I attacks, which involve modifying a legitimate signature so the system rejects it, are successful in all tested systems, even when access to signatures required to train writer-dependent classifiers is limited. The findings of this work support prior findings that assaults may transfer across various CNN classifiers, and they also demonstrate that systems based on bespoke feature extractors are susceptible to this transferability (CLBP). When CNN is trained with a new group of users, transferability is reduced (as opposed to a distinct set of samples from the same classes). They found that TypeII assaults, which try to pass as a real document, are substantially more challenging to carry out, have lower overall success rates, and need more noise

for the effective gradient-based technique. Object recognition literature mentions effective attacks with substantially lower noise (less than 3 orders of magnitude) (even in a targeted environment).

They ultimately looked at alternative ways and confirmed that the Madry defence increases the noise needed to create hostile images. In this study, they demonstrate that this defence is still effective throughout feature learning and WD classifier training. Strong attacks (Carlini) can detect hostile instances, but noise is required. Despite how weak this protection is, their noise reduction research shows it might lessen the assault success rate when the attacker is unaware of it (the adversary can incorporate this knowledge during the attack generation process). This problem is unresolved. Signal analysis might be a defensive move (a pen trajectory in 2D space). Future research will show how physical assaults influence people (for example, by printing adversarial noise on top of a signature).

2.2.5 AVN: An Adversarial Variation Network Model for Handwritten Signature Verification [5]

The verification of handwritten signatures is a crucial yet challenging subject. Despite major developments since previous research, they nevertheless passively take in distinguishing characteristics from the given data. This research presents a unique adversarial variation network (AVN) model for handwritten signature verification that actively modifies and produces new data to extract important properties. The AVN consists of three modules: the extractor, which extracts discriminative characteristics from handwritten signatures; the discriminator, which bases verification choices on the derived features; and the variator, which actively produces signature variants to develop a more discriminative model. A single end-to-end framework is created by combining all three different sorts of modules. The three parts of this technique cooperate and compete to enhance the overall system's capabilities and, as a result, the efficiency of the signature verification. This adversarial model uses a min-max loss function. They tested their approach on CEDAR, BHSig-Hindi, BHSig-Bengali, and GPDS Synthetic Signature. Several studies and discussions have shown this approach's efficacy.

The authors suggest a writer-independent adversarial variation network in this study. The network has three modules: an extractor that extracts discriminative features from handwritten sig-

natures, a discriminator that bases verification choices on the features, and a variator that creates signature variants to develop a more discriminative model. These lessons are taught using adversarial learning from start to finish. This technique is tested using four tough datasets in various languages.

The results of their trials show that their strategy improves the effectiveness of signature verification.The newly constructed variation consistency mechanism is more resistant to picture variation than standard data augmentation approaches. To highlight the approach's potential, they discuss its usage in multimedia, biometrics, and computer vision.

2.2.6 Learning Features for Offline Handwritten Signature Verification using Deep Convolutional Neural Networks [6]

In the face of competent forgeries, in which a forger has access to a person's signature and consciously reproduces it, it may be difficult to validate a handwritten signature. Offline (static) signature verification loses dynamic information about the signature-writing process, making it difficult to create effective feature extractors. Verification mistakes of roughly 7To recognise legitimate signatures from forgeries irrespective of user, they leverage data on sophisticated forgeries from a subset of users. GPDS, MCYT, CEDAR, and PUC-PR datasets were tested. By achieving 1.72 percent Equal Error Rate on the GPDS-160, they increased state-of-the-art performance. They also proved that the characteristics outperformed other datasets without modifying the representations.

In this paper, they presented a number of learning models for offline signature verification formulations. In contrast to techniques that rely on hand-engineered features, they showed that features learnt in a writer-independent way may be quite successful for signature verification.

They offered an example of a problem formulation in which knowing forgery data from a subset of users boosts the effectiveness of learned attributes for recognising forgeries for users who cannot be watched. Using this strategy, they reduce the EER in the GPDS-160 dataset from 6.97

The Writer-Independent format generalises to users of the GPDS dataset and other datasets, outperforming state-of-the-art in studies with the MCYT, CEDAR, and Brazilian PUC-PR datasets. The model constructed utilising forgeries from the GPDS dataset did not always perform better. Future study will examine if the forgeries in the two datasets had distinct features. Combining

offline and online signature verification is another area of investigation. Since it is harder to build a fake that is wrongly identified by both classifiers—a forgery with comparable strokes in terms of speed and that visually resembles a genuine user signature—the system's resilience may be strengthened.

2.2.7 Writer-Independent feature learning for offline signature verification using deep convolutional neural networks [7]

Researchers have studied Automatic Offline Handwritten Signature Verification from signal processing, graphology, and computer vision viewpoints. Despite advances, it's still challenging to build classifiers that can identify actual signatures from skilled forgeries (forgeries created especially for a particular signature). They indicate feature learning. In the absence of a reliable data-producing model, they believe that learning features from data is superior to hand-crafted signature qualities. They train writer-dependent classifiers using Deep Convolutional Neural Networks. They evaluated their techniques using PUC-PR and GPDS-960. Their study shows that user features may be utilised to discriminate persons across datasets, exceeding the state-of-the-art for the Brazilian PUC-PR dataset and the GPDS dataset.

They created a two-stage offline signature verification method based on writer-independent feature learning and writer-dependent classification. This technique trains students to use data without author aid by emphasising on data rather than human-developed qualities. This technique performed well on the GPDS and Brazilian PUC-PR datasets, indicating its potential. Using the GPDS dataset's properties, they found that the features generalised well on the Brazilian PUC-PR dataset. Even with fewer samples per user, this strategy may still work (4-5 samples). Last, they note that False Rejection and False Acceptance Rates are unequal and inconsistent between users and datasets, despite Equal Error Rates. This highlights the need of a reliable technique for determining user-specific thresholds, which they want to study further.

2.2.8 Meta-learning for fast classifier adaptation to new users of signature verification systems [8]

Over the last several decades, researchers have employed a variety of fields of study to analyse Automatic Offline Handwritten Signature Verification, including signal processing, graphology, and computer vision. Despite field improvements, distinguishing real signatures from professional forgeries is tough (forgeries created especially for a particular signature). They indicate feature learning. In the absence of a strong model of the data production process, they favour learning features from data to hand-crafted signature qualities. They use Deep Convolutional Neural Networks to learn writer-independent characteristics, then train writer-dependent classifiers. To test their approaches, they used PUC-PR and GPDS-960. The Brazilian PUCPR dataset beat the state-of-the-art, but the GPDS dataset came close, according to their experimental results, which revealed that features learned from a subset of users may be utilised to discern separate individuals across datasets.

Two-stage offline signature verification based on writer-independent feature learning and writer-dependent classification. This strategy uses data instead of manually designed features to exploit data without writers. This technique outperformed or matched state-of-the-art on the GPDS and PUC-PR datasets. Applying the GPDS characteristics to the Brazilian PUC-PR dataset demonstrated that they generalise well. Even with fewer samples per user, this technique proved beneficial (4-5 samples). Despite these strategies' modest Equal Error Rates, false rejection and acceptance rates are uneven and unpredictable between users and datasets. This underscores the necessity for a trustworthy mechanism for determining user-specific thresholds, which they'll study further.

2.2.9 Learning discriminative feature hierarchies for off-line signature verification [9]

The most frequent type of personal authentication is a handwritten signature. Handwritten signature verification (HSV) is a major pattern recognition challenge. In this study, they suggest enhancing offline HSV performance by training supervised convolutional neural networks to build discriminative feature hierarchies (CNN). The feature space is considered to reflect the core qualities of handwritten signatures since it models both global and local information. Learned charac-

teristics train writer-dependent SVMs for verification. According to the findings of their tests, this method effectively competes on the reference data sets MCYT-75 and CEDAR.

Automatic handwritten signature verification is an important job and a hot topic in pattern recognition research. In this study, they suggest utilising supervised convolutional neural networks to develop discriminative feature hierarchies to enhance offline HSV performance (CNN). During writer-independent feature learning, a DCNN architecture collects and combines shallow and deep features. Position-Dependent Siamese Network (PDSN) is also recommended to replicate local feature structure and learn a more discriminative feature space. Learned representation is used to verify writer-dependent SVM classifiers. When compared to cutting-edge methods, the experimental findings demonstrated that their methodology performed well on many benchmark datasets. The development set for this study was built using a number of unique signature databases. Despite their observation, PDSN overfitting was obvious.

2.2.10 Writer independent offline signature recognition using ensemble learning [10]

Handwritten signature verification is still an open research problem after years of dedicated work. Offline (static) signature verification loses dynamic information about the signature-writing process, making it difficult to create effective feature extractors. This verification effort gets considerably more difficult in writer independent situations, which are unquestionably fiscal in real-world settings.Offline (static) signature verification loses the dynamic information of the signature writing process, making it harder to construct effective feature extractors. After feature extraction with two CNNs, they employed RGBT classification and stacking to get the final prediction vector. They conducted extensive trials on a variety of datasets from various sources in order to maintain the dataset's volatility. They have delivered performance that is cutting edge on a variety of datasets.

In this paper, they provided a system for offline signature verification that is based on ensemble learning. Additionally, their cross-domain dataset studies demonstrate how well their ensemble model can identify fraud in a variety of signers' and forgers' handwriting styles from various backgrounds and scripts. Furthermore, for the bulk of benchmark signature datasets from dif-

ferent scripts and domains, the Ensemble Model they created outperformed cutting-edge results. The multimodal model, which unifies both online and offline signatures into a single model with multiple sorts of data augmentation, will be the focus of their future work on this subject.

Chapter 3

METHODOLOGY

A person's identity can be shown incontrovertibly and uniquely by their handwritten signature. Due to its simplicity and originality, it holds a crucial position in the field of behavioural biometrics. Secured authentication is crucial since signatures are the most frequently used biometric attribute for verification by law enforcement officials and agencies, particularly in financial institutions, legal transactions, etc. In the era of the digital age, there are a lot of transactions happening where the organisations, like banks, etc., demand handwritten signature verification. In these situations, the signature verification procedure should be quick, accurate, and secure, allowing for real-time verification.

3.1 Proposed System

In the process of the handwritten signature verification, the offline image and the online data of the signatures are collected, extracted it and then put into the classifier to obtain the model. In the test stage, the signatures are put into the classifier for computation and output verification result.

The proposed methodology uses various deep learning algorithms like VGG16, Inception-v3, ResNet50, Xception that extracts the features from the signature images and compares the accuracies of created models.

The proposed system consist of two major phases:

- Preprocessing and Feature Extraction

- Classification and Verification

3.2 System Architecture

The main purpose of the project is to verify the handwritten signatures. So by using some deep learning algorithms with Adam as optimizer, handwritten signatures can be verified by comparing the accuracies. For the purpose of assessing the accuracies, this study employs four well-known convolutional neural network models: VGG16, Inception-v3, ResNet50, and Xception. While the next sections provide thorough explanations, Fig. 3.1 provides an overview of the overall workflow.

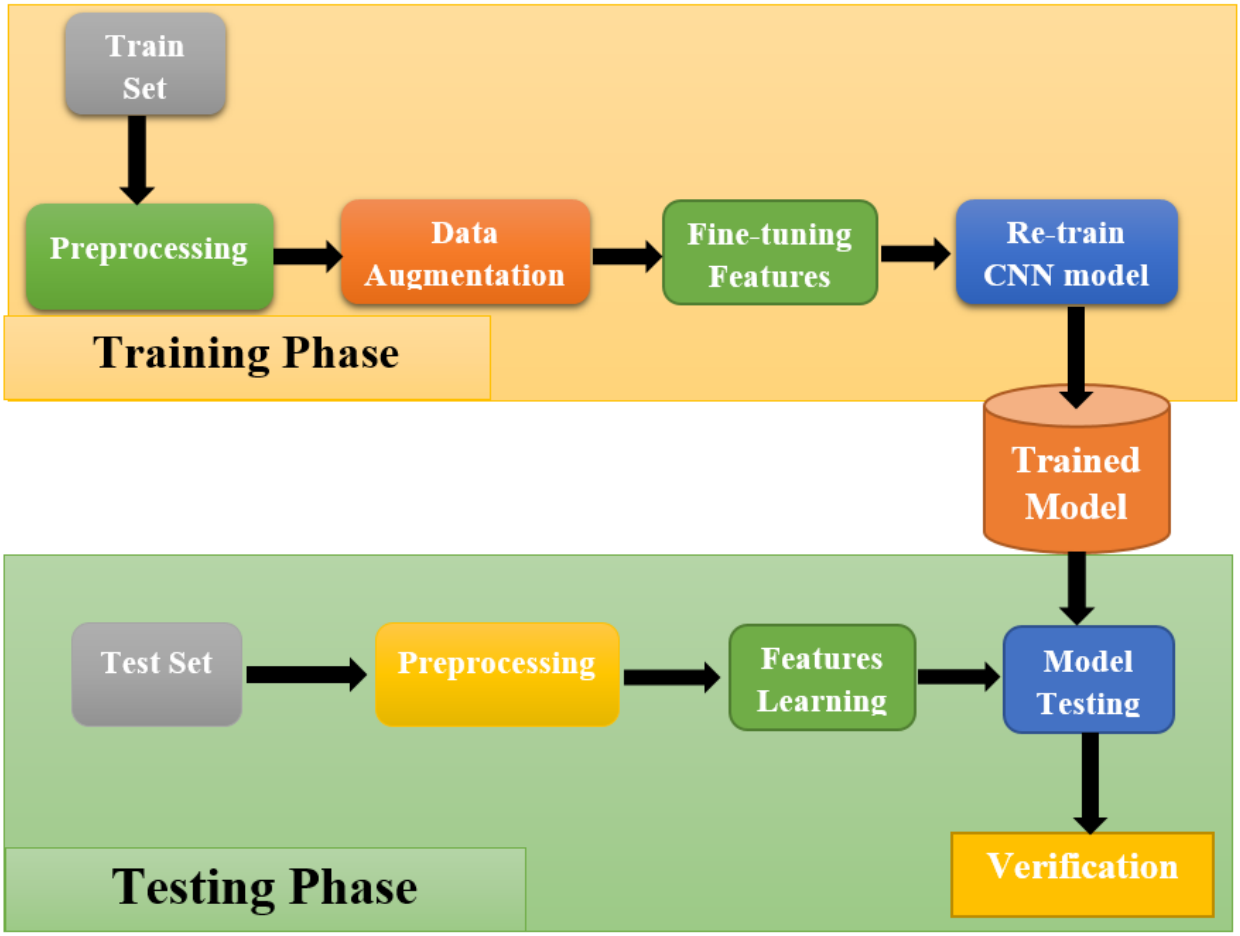

Figure 3.1: System Architecture

3.2.1 Dataset

Handwritten Signature Verification employs a dataset obtained from the Kaggle website (ICDAR 2011 Signature Verification Competition (SigComp2011)). Offline (static) signature verification loses dynamic signature building information, making feature extractors less efficient. All of the images are handwritten signatures in RGB format. The obtained dataset was made up of training, testing, and validating images, each divided by Genuine and Forged signatures. Data for testing is extracted from the ICDAR 2011 Signature Dataset and organised pairwise.

DUTCH DATASET

TRAINING SET

Ten reference writers' signatures, as well as expert imitations of these signatures for use in both online and offline modes. Total online signatures: 449, total offline signatures: 362 Additionally, trainers can make use of the 2009 competition's open data.

TEST SET

The signatures of 54 reference writers as well as expert forgeries of these signatures were used in both online and offline modes. There were 1907 online signatures and 1932 offline.

CHINESE DATASET

TRAINING SET

Ten reference writers' signatures, as well as expert imitations of these signatures for use in both online and offline modes. Total online signatures: 659, total offline signatures: 575.

TEST SET

Ten reference writers' signatures, as well as expert imitations of these signatures for use in both online and offline modes. Total online signatures: 680; total offline signatures: 602

3.3 Preprocessing and Feature Extraction

Examples of data-augmentation techniques used to solve generalisation, enormous input space, and over-fitting difficulties include image sharpening and boundary extraction. Then, by adjusting models based on transfer learning, a large number of deep visual and salient features are extracted. The detection and verification of a person's true signature is being examined and explored using a variety of CNN-based architectures, including VGG16, Inception-v3, ResNet, and Xception. Finally, a fully linked layer that performs classification and verification using the softmax function receives the deep features vector.

A refined approach to deep transfer learning makes it a powerful artificial intelligence tool. With the use of source datasets, CNN architecture is trained for customised transfer learning (Ds).Using a fresh data set called the target data-set, the trained model is improved (Dt). For object detection using outdated or conventional techniques, the issue of feature extraction and selection is critical and difficult. A human-imitated technique is used by deep convolutional networks

to automatically pick up features. The characteristics of each layer represent several data interpretations. Basic components like blobs, edges, or curves make up the first few layers. The deeper and topmost layers hold the features' high-level vision.

Using convolutional neural network models that have already been trained to recognise learned properties like edges and blobs, I run a series of tests on the database in this part. I first train a range of pre-trained CNN models, including VGG16, Inception-v3, ResNet50, and Xception, using the dataset to learn and extract the best visual and optimal signature patterns for verification.

3.4 Training the model

Another technique for enhancing the performance of deep models, notably in CNNs, is transfer learning. Breaking out from the isolated learning paradigm, transfer learning is the idea of using the information acquired for one activity to address related problems. There are three unique transfer learning methods used in CNNs. These models are pre-trained, feature extractor, and fine-tuning. The discovery that early layers of CNNs had more generic properties, such as borders and colours, served as the study's inspiration for its fine-tuning methodology.

3.5 Testing the model

Training data and testing data were created from the selected data. Using the training data, a deep learning model was created, and the model was then verified using the testing data. The model, which was created using training data, was put to the test using test data in order to assess its accuracy-related performance. At this point, the constructed model has been put to the test using the test dataset. The suggested models are evaluated using test data images for categorising signatures. Testing accuracy is the most crucial performance metric for a classification algorithm. Comparing the suggested VGG16 model to the other three models, it is significantly more accurate.

3.6 VGG16 Architecture

Simonyan and Zisserman first described VGG16. The model's top-5 test accuracy in ImageNet, a dataset containing 14 million photographs, is 92.7Conv1 receives a 224x224 RGB picture. When an image is processed using convolutional layers, the filters are adjusted to capture left/right, up/down, and centre with a restricted receptive field of 33. (the smallest size to catch these concepts). In one form, it uses 11 convolution filters, which modify input channels linearly (followed by non-linearity).Stride 2 max-pools a 22-pixel frame.

Following convolutional layers are three FC layers (which has a different depth in different architectures). Finally, soft-max. Every network has the same fully connected tiers. All buried layers are non-linear (ReLU). No networks (save one) use Local Response Normalisation (LRN), which degrades ILSVRC dataset performance while increasing memory use and processing time. Figure 3.2 shows VGG16's layers.

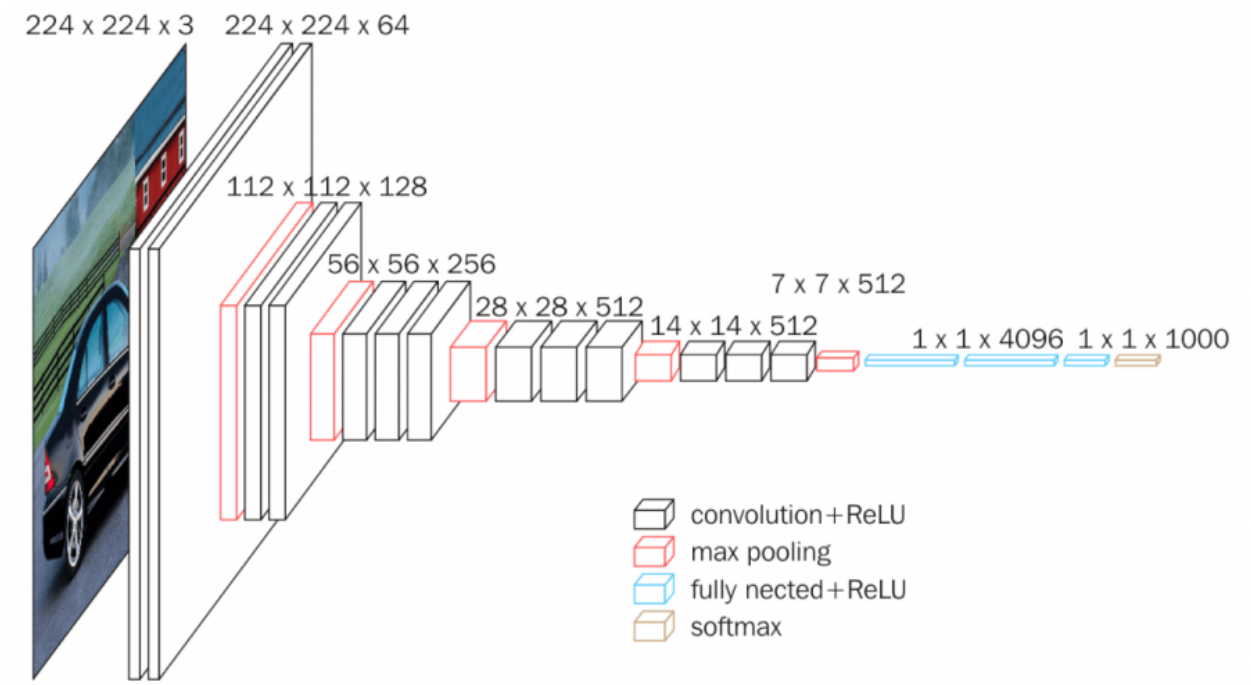

Figure 3.2: Architecture of VGG16

3.7 Inception-v3 Architecture

Label Smoothing, Factorized 7 x 7 convolutions, and an auxiliary classifier help Inception-v3 send label information through the network (along with the use of batch normalisation for layers in the sidehead).

The third iteration of a line of Deep Learning Convolutional Architectures is Google's Inception V3. The Tensorflow version of Inception V3 now contains 1,001 classes as a result of the inclusion of a "background" class that wasn't included in the original ImageNet dataset. To train Inception V3 with 1,000 classes, the original ImageNet dataset was used (see the list of classes here). Inception V3 placed second in the ImageNet Large Visual Recognition Challenge.

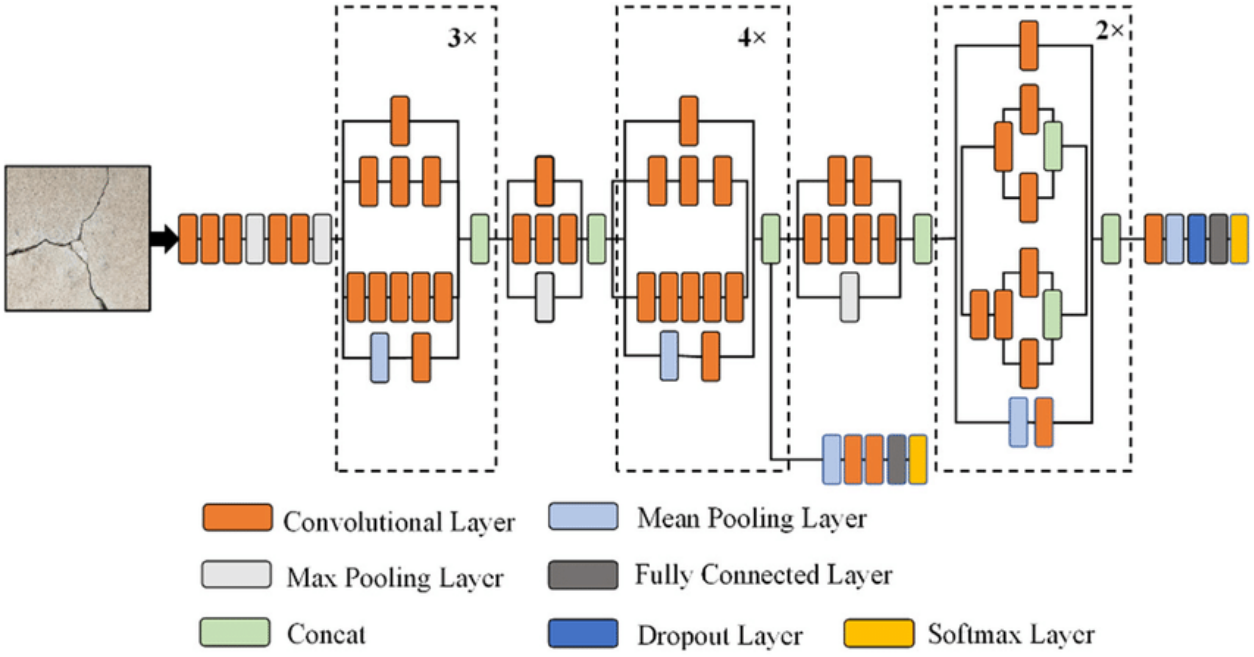

Figure 3.3: Architecture of Inception-v3

3.8 ResNet50 Architecture

50-layer DCN ResNet-50. Many computer vision tasks use ”ResNet” neural networks. ResNet enabled 150-layer neural network training. Deep Residual Learning for Image Recognition was released in 2015. There is a big problem with convolutional neural networks called the ”Vanishing Gradient Problem.” During backpropagation, the gradient value drops a lot, so there isn't much change in the weights. ResNet is used to get around this. It is used to ”SKIP CONNECTION.”

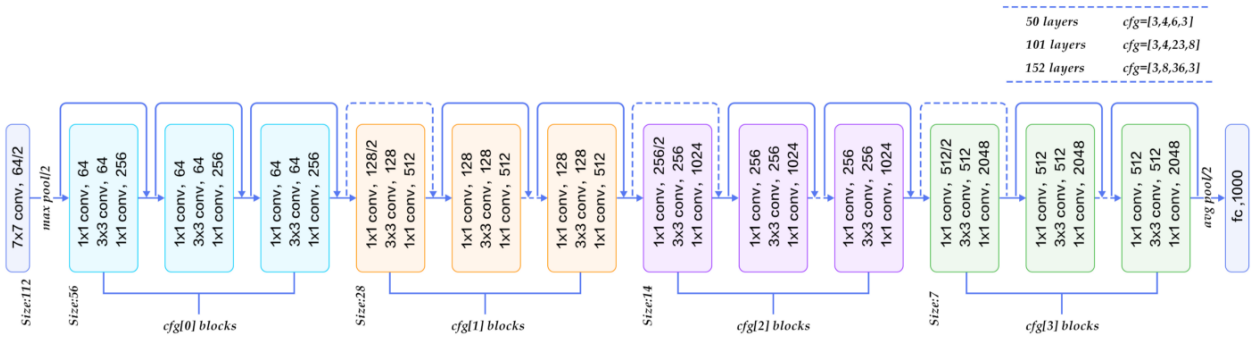

Figure 3.4: Architecture of ResNet50

3.9 Xception Architecture

Depthwise Separable Convolutions are a part of Xception, which is an architecture for deep convolutional neural networks. Scientists at Google were the ones who made it. Inception modules bridged conventional and depth-separable convolution (a depthwise convolution followed by a pointwise convolution). An indefinitely large Inception module is like a depthwise separable convolution. Using this knowledge, they build a new Inception-based deep convolutional neural network. Instead of Inception modules, this architecture uses depth-separable convolutions.

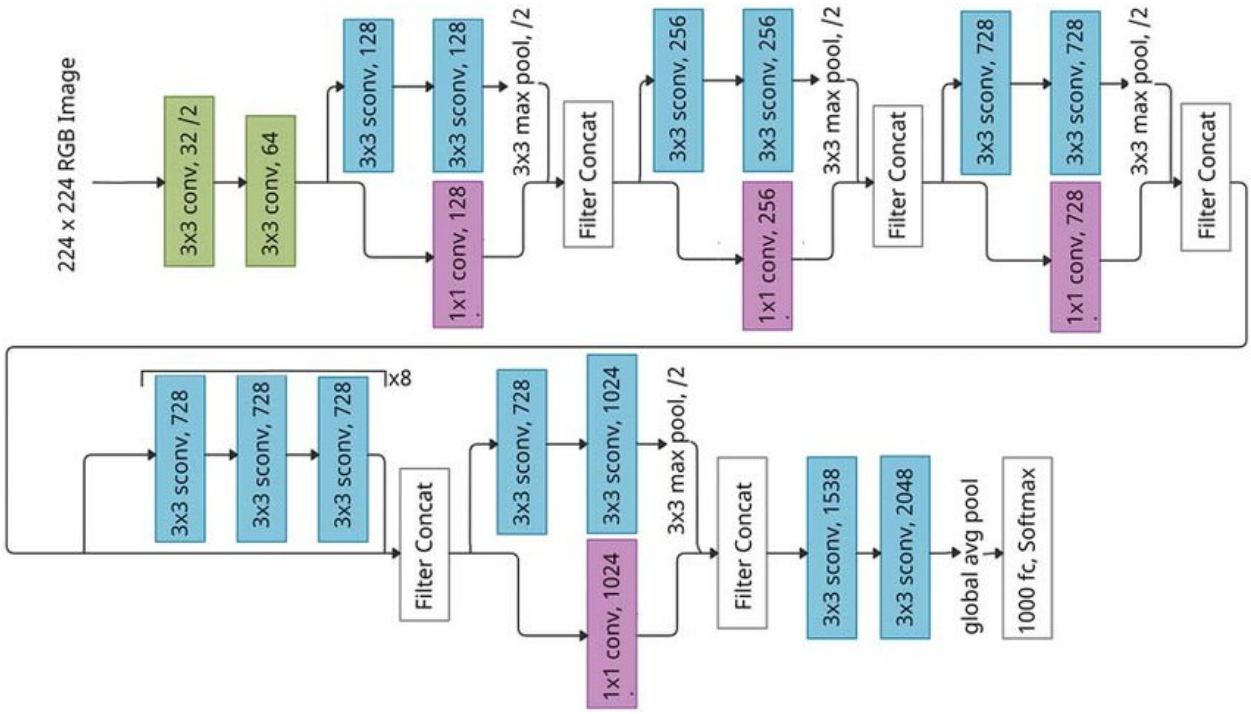

Figure 3.5: Architecture of Xception

3.10 Software Requirement and Specification

The following software was used for the project:

• Python

• Google Colab

• Jupyter Notebook

3.10.1 Python

Python is a high-level computer language. Python was created in 1991 by Guido van Rossum, and its design philosophy emphasises the use of plenty of whitespace to make code simple to understand. Programmers may develop code that is straightforward and simple to comprehend for both small and big projects because to its language architecture and object-oriented approach. Python provides dynamic typing and removes outdated code on its own. It supports procedural, object-oriented, and functional programming. Python is called ”all-inclusive” due to its huge standard

library. Python 2.0, which came out on October 16, 2000, added some important new features, such as a garbage collector that can find cycles and support for Unicode. Python 3.0 came out on December 12, 2008. It was a big change to the language that was only partly compatible with the old one. Many of its most important features have been added to Python 2.6.x and 2.7.x. The 2 to 3 utility, which comes with Python 3, makes it easier to convert Python 2 code to Python 3 (at least in part). Python is meant to be an easy language to learn. It's easy to read and often uses English words in place of punctuation in other languages. Unlike many other languages, it does not use curly brackets to separate blocks, and semicolons are not required to end statements. Compared to C or Pascal, it has fewer exceptions and special cases when it comes to syntax. Guido van Rossum wrote about Python's goals in 1999:

• A straightforward, understandable language that is equally potent as that of the main rival.

• Because it is open source, anyone can help with its development.

• Programming language that is as easy to grasp as plain English.

The goal of Python's design is to be very readable. It substitutes English words for punctuation rather often and has fewer syntactical features than other languages. Students and working adults who wish to become excellent software engineers should study Python, particularly if they work in the web development sector. Some of the key benefits of learning Python include the following:

• Python is Interpreted: Python is processed by an interpreter at runtime. Your software does not need to be compiled before executing. This is similar to PHP and PERL.

• Python is Interactive: You can immediately interact with the interpreter while writing programmes by sitting at a Python prompt.

• Python offers the object-oriented programming method, which encapsulates code into objects.

Python is a great language for novice programmers because it enables users to build a variety of programmes, from simple text processing to World Wide Web browsers to games.

3.10.2 Google Colab

Google Colab was made so that anyone who needs to build a machine learning or deep learning model can use GPUs and TPUs for free. Google Colab is like a better version of Jupyter Notebook. With the help of the Jupyter Notebook app, you can use a web browser or an Integrated

Development Environment to edit and run Notebook files (IDE).

Google Colab has a lot of interesting features that you won't find in any other modern IDE. Here are some of the most interesting traits.

• Machine learning and neural network tutorials that you can do on your own.

• Make Python 3 programmes and run them without a local setup.

• Use the Notebook to send commands to the terminal. bring in information from places like Kaggle.

• Google Drive notebook uploads.

• Import Notebooks from Google Drive.

• Free TPUs, GPUs, and cloud computing.

• Use OpenCV, Tensor Flow, and PyTorch.

• Publish or import directly from/to GitHub.

3.10.3 Jupyter Notebook

Fernando Perez made the Jupyter notebook so that the IPython kernel could be used through the web. Project Notebook is now called Project Jupyter, and it has a front end for the programming environments Juila and R in addition to Python. This is part of an effort to make an integrated interactive computing environment for many languages.

In a notebook document, you can find graphics, equations, and text that is formatted in HTML, among other things. The notebook also works as an executable document and has code blocks for Python or other languages that help it work. Jupyter Notebook is an example of a programme with a server and a client. The application runs a local server and opens the notebook interface in a web browser. After HTML, PDF, or LaTex, the notebook is saved as ipynb.

ADVANTAGES OF JUPYTER NOTEBOOK

• Conveniently located: You may already be aware of Jupyter Notebook, an interactive open-source web environment that combines code, text, photos, videos, mathematical equations, charts, maps, and widgets into a single page.

• Convertibility: Users of Jupyter Notebook can convert their notebooks into other formats like HTML and PDF. Additionally, it takes use of online resources and nbviewer, which enables you to

render a publicly accessible notebook directly in a browser.

• Shareable: Because Jupyter Notebooks are stored as structured text files, sharing them is simple (JSON format).

• Jupyter Notebook is cross-platform since it is stored in the language-neutral JSON (JavaScript Object Notation) text-based file format. The notebook's ability to be processed by any computer language and converted to any file type, such as Markdown, HTML, PDF, and others, is another benefit.

• Interactive code: The ipywidgets packages, which offer a variety of standard user interfaces for interacting with code and data, are used by Jupyter notebook.

DISADVANTAGES OF JUPYTER NOTEBOOK

•Testing lengthy asynchronous jobs is quite challenging.

• Lessening Security

• The incorrect order of the cells is used.

• Jupyter notebook does not support IDE integration, code linting, or code style correction.

Chapter 4

RESULTS AND DISCUSSIONS

The results of the developed model will be discussed in this section. The model verifies handwritten signatures and compares accuracies using deep learning models.

Create the model, then assemble the neural network using Adam as the optimizer, a 64-batch size, and a categorical cross-entropy loss function. Several methods are used to avoid overfitting. After all the layers were connected, the dropout method with a 0.4 rate was used first. Second, early stopping and data augmentation are used to avoid overfitting.

4.1 Training and validation results

The model is trained for 20 epochs in batches of 64 for the training and testing phases. Thus, training for 20 such epochs produced optimised results, with high accuracy of 96 percent for VGG16, 71% for Inception-v3, 49% for ResNet50, and 50% for Xception, and corresponding validation losses of 0.11%, 0.75%, 0.77%, and 3.94%, respectively. Training accuracy for VGG16, Inception-v3, ResNet50, and Xception is 99.48%, 99.57%, 98.70%, and 100%, respectively. Table 4.1 displays the training and testing accuracy of the models.

MODEL	TRAINING ACCURACY	TRAINING LOSS	VALIDATION ACCURACY	VALIDATION LOSS
VGG-16	0.9948	0.0152	0.9616	0.1193
Inception-v3	0.9957	0.0167	0.7111	0.7530
ResNet50	0.9870	0.0426	0.4970	0.7776
Xception	1.000	0.0024	0.5051	3.9422

Table 4.1: Accuracy Table

Chapter 5

CONCLUSION

Although there have been many attempts to verify signatures, there are still many difficulties in this field, making it a crucial duty for security concerns to prevent document falsification in a variety of legal, financial, governmental, and other lucrative situations. In this paper, I used the VGG16, Inception-v3, ResNet50, and Xception architectures to talk about identifying and verifying signatures. On the ICDAR 2011 dataset, I employed honed feature extraction techniques with the VGG16 classifier for the classification and verification task, and it outperformed cutting-edge methods. Using a fine-tuned approach, the proposed system achieved an accuracy of 96.16 percent. The proposed deep learner classifier has the advantage of automatically extracting hidden features from signature images, resulting in a significant increase in classifier performance. Signature verification is carried out in this study using an offline Persian signature database.

5.1 Advantages

The main merits of proposed model are:

- User-friendly.
- Well accepted socially and legally.
- Already acquired in a number of applications.
- Long experience in forensic environments.

- Non invasive

5.2 Future Enhancement

The results of the experiments show that the method improves signature verification performance.Additionally, the suggested method works better than conventional data augmentation and is more resistant to variations in the images than the comparative methods. The focus of future research will be on generative models for signature verification and the application of the technique to other multimedia, biometric, and vision issues.

REFERENCES

[1] Saeeda Naz, Kiran Bibi, Riaz Ahmad, ”DeepSignature: fine-tuned transfer learning based signature verification system”, Springer,2022

[2] Mohammad Saleem, Bence Kovari , ”Online signature verification using signature down-sampling and signer-dependent sampling frequency”, Springer, 2021

[3] Md Ajij,Sanjoy Pratihar ,Soumya Ranjan Nayak,Thomas Hanne ,Diptendu Sinha Roy , ”Off-line signature verification using elementary combinations of directional codes from boundary pixels”, Springer, 2021

[4] Luiz G. Hafemann, Robert Sabourin, and Luiz S. Oliveira, ”Characterizing and evaluating adversarial examples for Offline Handwritten Signature Verification”, IEEE, 2019

[5] Huan Li , Ping Wei ,and Ping Hu, ”AVN: An Adversarial Variation Network Model for Handwritten Signature Verification”, IEEE, 2022

[6] Luiz G. Hafemanna, , Robert Sabourina , Luiz S. Oliveirab, ”Learning Features for Offline Handwritten Signature Verification using Deep Convolutional Neural Networks”, IEEE, 2017.

[7]] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, “Writer-Independent feature learning for offline signature verification using deep convolutional neural networks,” in Int. Joint Conf. Neural Netw., 2016, pp. 2576–2583.

[8] L. G. Hafemann, R. Sabourin, and L. Soares de Oliveira, “Meta-learning for fast classifier adaptation to new users of signature verification systems,” IEEE Trans. Inf. Forensics Secur., vol. 15, pp. 1735–1745, 2020.

[9] S. Lai and L. Jin, "Learning discriminative feature hierarchies for off-line signature verification," in Proc. Int. Conf. Front. Handwriting Recognit., 2018, pp. 175–180.

[10] S. D. Das, H. Ladia, V. Kumar, S. Mishra, "Writer independent offline signature recognition using ensemble learning," in Proc. Int. Conf. Data Sci., Mach. Learn. Appl., 2019.

APPENDIX

A Screenshots

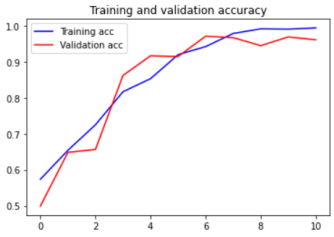

Figure A.1 : Accuracy graph of VGG16

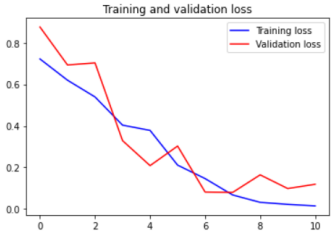

Figure A.1 : Loss graph of VGG16

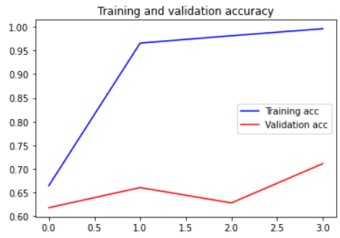

Figure A.1 : Accuracy graph of Inception-v3

Figure A.1 : Loss graph of Inception-v3

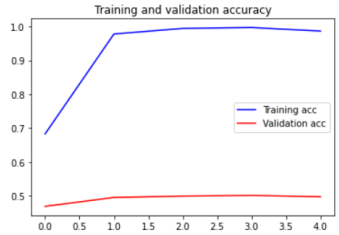

Figure A.1 : Accuracy graph of ResNet50

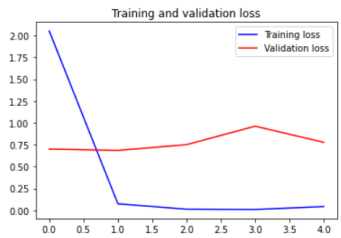

Figure A.1 : Loss graph of ResNet50

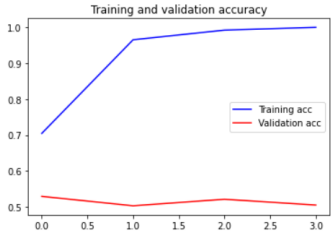

Figure A.1 : Accuracy graph of Xception

Figure A.1 : Loss graph of Xception