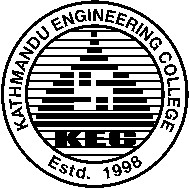
Tribhuvan University

**Institute of Engineering**

Kathmandu Engineering College

Department of Computer Engineering



Minor Project Report

On

SIGNATURE VERIFICATION USING CNN

[Code No: CT654]

By

Prajwol Shrestha – 76050

Rujal Shrestha – 76062

Saurav Khanal – 76071

Suprim Shakya – 76087

Kathmandu, Nepal

2079

Tribhuvan University

**Institute of Engineering**

Kathmandu Engineering College

Department of Computer Engineering

SIGNATURE VERIFICATION USING CNN

[Code No: CT654]

PROJECT REPORT SUBMITTED TO

THE DEPARTMENT OF COMPUTER ENGINEERING

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE BACHELOR OF ENGINEERING



By

Prajwol Shrestha – 76050

Rujal Shrestha – 76062

Saurav Khanal – 76071

Suprim Shakya – 76087

Kathmandu, Nepal

Ashad 2079

TRIBHUVAN UNIVERSITY

Kathmandu Engineering College

Department of Electronics and Computer Engineering

# ABSTRACT

A signature serves as a visual representation of the signer's consent, dedication, inclination, obligation, etc. The signature is unique to every individual. Ideally no other person or medium should be able to recreate a person’s signature. But the cases of signature forgery are not new.

Manual verification has a lot of subjectivity, and its efficacy tends to be inconsistent based on the expertise of the verifier as well as their mood, fatigue levels, and distractions. As a result, manual signature review can lead to an uncomfortable number of false rejections. To tackle such problem new efficient tool is needed. This project offers a signature verification tool that can aid humans in making accurate decisions in the authentication of handwritten signatures to address such a problem.

The users will be able to upload sample signature to train model and also scan signature to be verified to get the accuracy as result.

# TABLE OF CONTENTS

[**ABSTRACT iii**](#_Toc108511971)

[**TABLE OF CONTENTS iv**](#_Toc108511972)

[**LIST OF FIGURES vi**](#_Toc108511973)

[**LIST OF ABBREVIATION vii**](#_Toc108511974)

[**CHAPTER 1: INTRODUCTION 1**](#_Toc108511975)

[1.1 Background 1](#_Toc108511976)

[1.1.1 Neural Network 2](#_Toc108511977)

[1.1.2 Convolutional Neural Network 2](#_Toc108511978)

[1.2 Problem Statement 2](#_Toc108511979)

[1.3 Objective 3](#_Toc108511980)

[1.4 Scope and Application of the Project 3](#_Toc108511981)

[**CHAPTER 2: LITERATURE REVIEW 4**](#_Toc108511982)

[**CHAPTER 3: METHODOLOGY 6**](#_Toc108511983)

[3.1 PROCESS MODEL 6](#_Toc108511984)

[3.2 Operations in CNN 7](#_Toc108511985)

[3.2.1 Convolution step: 7](#_Toc108511986)

[3.2.2 Activation Function 8](#_Toc108511987)

[3.2.3 The Pooling Step 8](#_Toc108511988)

[3.2.4 Classification Step 9](#_Toc108511989)

[3.3 Block Diagram 10](#_Toc108511990)

[3.4 Project Workflow 11](#_Toc108511991)

[3.5 TOOLS TO BE USED 13](#_Toc108511992)

[3.5.1 Python 13](#_Toc108511993)

[3.5.2 TensorFlow 13](#_Toc108511994)

[3.5.3 Keras 13](#_Toc108511995)

[3.5.4 Pandas 13](#_Toc108511996)

[3.5.5 NumPy 13](#_Toc108511997)

[3.5.6 Inception V3 14](#_Toc108511998)

[3.5.7 Tkinter 14](#_Toc108511999)

[**CHAPTER 4: EPILOGUE 15**](#_Toc108512000)

[4.1 EXPECTED OUTPUT 15](#_Toc108512001)

[4.2 GANTT CHART 15](#_Toc108512002)

[**REFERENCES 16**](#_Toc108512003)

[**BIBLOGRAPHY 16**](#_Toc108512004)

# LIST OF FIGURES

Figure 3.1: Block Diagram of Incremental Process Model

Figure 3.2.1: Convolution of 5\*5 matrix image with 3\*3 kernel

Figure 3.2.2: Activation Functions

Figure 3.2.3: Max Pooling and Average Pooling

Figure 3.2.4: Schematics of Basic CNN

Figure 3.3: Block Diagram of the Purposed System

Figure 3.4: Project Workflow

Figure 4.2: Gantt Chart

# LIST OF ABBREVIATION

AI: Artificial Intelligence

CNN: Convolutional Neural Network

GUI: Graphical User Interface

ML: Machine Learning

ReLU: Rectified Linear Unit

# CHAPTER 1: INTRODUCTION

## 1.1 Background

A signature is a handwritten and frequently stylized representation of a person's name and nickname that a person puts on documents as proof of identity and genuine intent The handwritten signature of a person is commonly accepted as a means of verifying the legality of documents such as certificates, checks, drafts, letters, approvals, visa, passport, etc. It is indispensable in countering the forgery and falsification of such documents in diverse financial, legal, bureaucratic, academic, and other commercial settings. Take for instance, in any bank at the point when the cashier gets a cheque from a client, the cashier compares the received signature with a stored record of a genuine signature before continuing with any legitimate transaction.

The convention of using signatures to confirm the authenticity of documents has been followed in the past and is still followed in the present and will continue in the future. Authentication of the signature is very crucial in legal, and financial cases. Fraudulent signatures in such cases may have severe damage to a person's image and assets. So, to avoid this kind of fraud, a methodical approach to signature verification is essential.

Traditionally, authentication of signature is done manually by comparing with previously acquired genuine data. In the context of Nepal especially in the banking sector, signature verification is important in various transaction and approvals processes. The banks have signature capture software. This software records the customer's signature at the time of account opening. Bank verifies the signature based on the sample signature retained in the bank’s record. But such a simple approach may not be sufficient in all cases as various advanced forgery and falsification techniques are emerging.

Software used for signature verification compares signatures and verifies their authenticity. This reduces the possibility of fraud during the authentication process, saves time and resources, and helps to prevent human error during the verification process. The software compares from a vast database and confirms the validity score for the signature. Too low of a validity level suggests the signature is most likely a fraud.

### 1.1.1 Neural Network

A neural network is a method in AI that teaches computers to process data in a way that is inspired by the human brain. It is a type of machine learning process, called deep learning, that uses interconnected nodes or neurons in a layered structure that resembles the human brain. It creates an adaptive system that computers use to learn from their mistakes and improve continuously. Neural networks can help computers make intelligent decisions with limited human assistance. This is because they can learn and model the relationships between input and output data that are nonlinear and complex. For instance, they can Make generalizations and inferences, reveal hidden relationships and patterns, Create autonomous self-learning systems etc.

### 1.1.2 Convolutional Neural Network

In deep learning, a convolutional neural network (CNN/ConvNet) is a class of deep neural networks, most commonly applied to analyze visual imagery. Convolutional Neural Networks are a category of Neural Networks that have proven very effective in areas such as image recognition and classification. CNNs have been successful in identifying faces, objects and traffic signs apart from powering vision in robots and self-driving cars. CNN takes in an input image, assigns importance (learnable weights and biases) to various aspects/objects in the image and differentiates one from the other.

Every image is made up number of pixels. For example, if we take 28\*28 black and white image then it has total 768 number of pixels. The CNN takes individual pixels as an input, applies filters to it, derives some patterns within the pixels of data to give an output.

## 1.2 Problem Statement

The verification of one’s identity via signature is a risky task. The person verifying the signature must be absolutely certain of the decision. The validation of signature in many cases is highly critical and any inaccuracy in the authentication may result in serious consequences and damages.

In order to avoid such scenarios and prevent potential damage, a modern robust approach must be adopted to verify the genuineness of the signature. Adopting such an approach will assist a person in making decisions over the authenticity of signatures and prevent mistakes.

## 1.3 Objective

The objective of this project is:

* To develop an application that can assist and improve the verification of the authenticity of a signature by implementing convolutional neural network.

## Scope and Application of the Project

Software applications for verifying the genuineness of handwritten signatures can be applicable in various sectors and activities. Some conceived applications are:

1. Customer signature validation on banks
2. Government services document verification
3. Academic and professional certification
4. Contracts and legal statements

# CHAPTER 2: LITERATURE REVIEW

The area of Handwritten Signature Verification has been broadly researched in the last decades and still remains as an open research problem. This project focuses on offline signature verification, characterized by the usage of static (scanned) images of signatures, where the objective is to discriminate if a given signature is genuine (produced by the claimed individual), or a forgery (produced by an impostor). We present an overview of how the problem has been handled by several researchers in the past few decades and the recent advancements in the field.

Article published in International Journal of Scientific & Engineering examined signature verification using neural network approach and analyzed its strengths and weakness [1]. Paper presented method which uses geometric features extracted from preprocessed signature images, which trained neural network using error back propagation training algorithm for verification of signature. They used a feature vector of dimension 60 to uniquely characterize a candidate signature. Article used different technique for verification analyzing different error rate: False Acceptance Rate (FAR), False Rejection Rate (FRR) and Correct Classification Rate (CCR). Result of research was 12% FAR, 16.7% FRR and Correct Classification Rate is 85.7%.

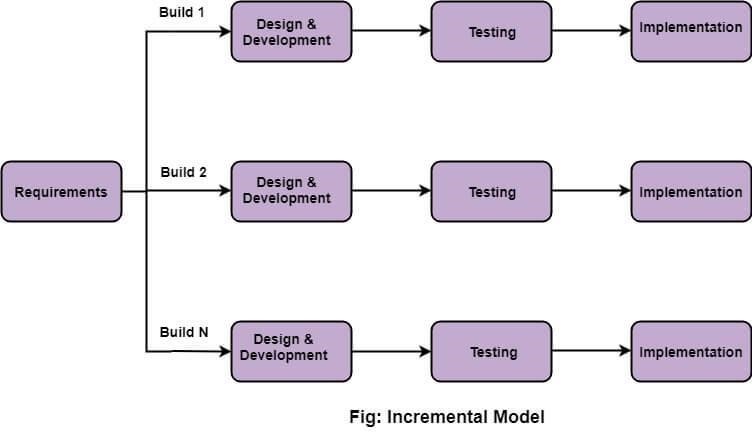
A research conducted in Stanford University used convolutional neural network for offline signature verification based on the VGG-16 architecture and ICDAR 2011 SigComp dataset to train their model with transfer learning [2]. The dataset includes both online and offline signatures (of which we only use the latter) for both Chinese and Dutch signers. The dataset was split into a training set and testing set of non-overlapping IDs. The Dutch training set included a total of 366 images for 10 IDS, with about 25 genuine signatures and 11 forged signatures for each ID. The result from research was validation accuracy of 67.1%, FAR of 33.0%, FRR of 33.0 %. The main limitation of research was limited dataset and with large data, the result would have been better.

Research paper of topic “offline handwritten signature retrieval using curvelet transform” proposed a new method for offline handwritten signature retrieval based on curvelet transform [3]. It focused on applications of image processing with similarity retrieval of an image from large collections of images. In such case image indexing become important for efficient organizational and retrieval of images. The proposed system used a curvelet based texture feature extraction. The performance of the system was tested with an image database of 180 signatures. The result obtained indicated that the proposed system was able to identify signatures with greater accuracy even when part of signature was missing.

# CHAPTER 3: METHODOLOGY

## 3.1 PROCESS MODEL

As we are trying to develop a signature verification system, we have to use training data to the system and make changes to the system as per our need to get a more efficient architecture with a minimum number of neurons and maximum efficiency within a small period of time, the software development model best for us was found to be Incremental model.



#### Figure 3.1: Block Diagram of Incremental Process Model

Incremental Model is a process of software development where requirements divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, implementation and testing phases. Every subsequent release of the module adds function to the previous release. The process continues until the complete system achieved. [4]

Incremental software development is a fundamental part of agile approaches, is suitable for the projects where system requirements rapidly during the development. Incremental development reflects the way that we solve the problems, i.e., move toward a solution of a problem is a series of steps, backtracking when we realized that we have made a mistake.

Customers can evaluate the system at a relatively early stage in the development to see if it delivers what is required. If not, then only the current incremental has to be changed and possibly, new functionality defined for later increments.

**Advantages of Incremental Development:**

* The cost of accommodating changing customer requirements is reduced.
* It is easier to get customer feedback on the development work that has been done.
* More rapid delivery and deployment of useful software to the customer is possible, even if all of the functionality has not been included

**Disadvantages of Incremental Development**

* Not suitable for large, complex, long-lifetime systems, where different teams develop different parts of the system.
* Incremental delivery and deployment is not always possible as experimenting with new software can disrupt normal business processes.
* Not suitable for a large system development. A large system needs a stable framework or architecture and the responsibilities of the different teams working on parts of the system need to be clearly defined with respect to that architecture. This has to be planned in advance rather than developed incrementally.

## 3.2 Operations in CNN

The four main operation in CNN are:

1. Convolution
2. Activation Function
3. Pooling or Sub Sampling
4. Classification (Fully Connected Layer)

### 3.2.1 Convolution step:

The convolution step is feature extraction step. The image matrix is convoluted with a filter matrix to extract certain features of the input image simultaneously reducing the size of input matrix. The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. As we move deeper into the network it can identify even more complex features such as objects, faces, etc.

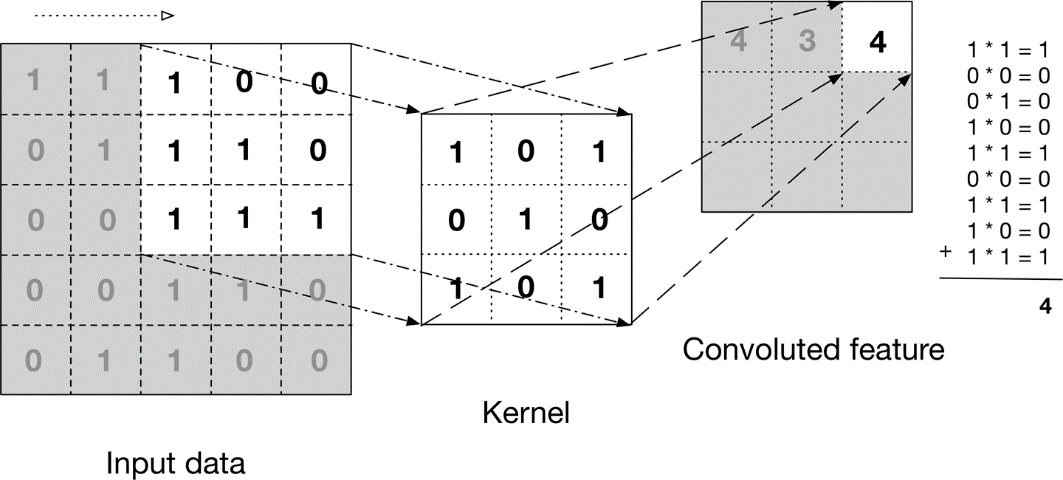


Figure 3.2.1: Convolution of 5\*5 matrix image with 3\*3 kernel

### 3.2.2 Activation Function

The activation function is a node that is put at the end of or in between Neural Networks. They help to decide if the neuron would fire or not. The activation function is the non-linear transformation that we do over the input signal. This transformed output is then sent to the next layer of neurons as input

ReLU function is the most widely used activation function in neural networks today. One of the greatest advantages ReLU has over other activation functions is that it does not activate all neurons at the same time.

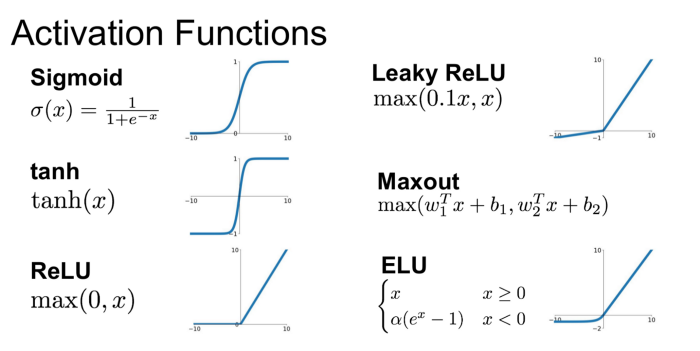


Figure 3.2.2: Activation Functions

### 3.2.3 The Pooling Step

Spatial Pooling (also called subsampling or down sampling) reduces the dimensionality of each feature map but retains the most important information. Spatial Pooling can be of different types: Max, Average, Sum etc.



Figure 3.2.3: Max Pooling and Average Pooling

### 3.2.4 Classification Step

This is the final step. The output from the convolution and pooling layers represents high-level features of the input image. The purpose of the Fully Connected layer is to use these features for classifying the input image into various classes based on the training dataset. The term “Fully Connected” implies that every neuron in the previous layer is connected to every neuron on the next layer.



Figure 3.2.4: Schematics of Basic CNN

## 3.3 Block Diagram

Sampled dataset

Signature to be verified

Pre-Processing

Pre-Processing

Signature Validation

Trained Model

System

Figure 3.3: Block Diagram of Signature Verification process

Firstly, signature sample is collected from user then it is pre-processed and a model is created on the basis of that sample.

Next signature to be verified is collected from the user it is also preprocessed and fed to that trained model which gives the signature validation as output.

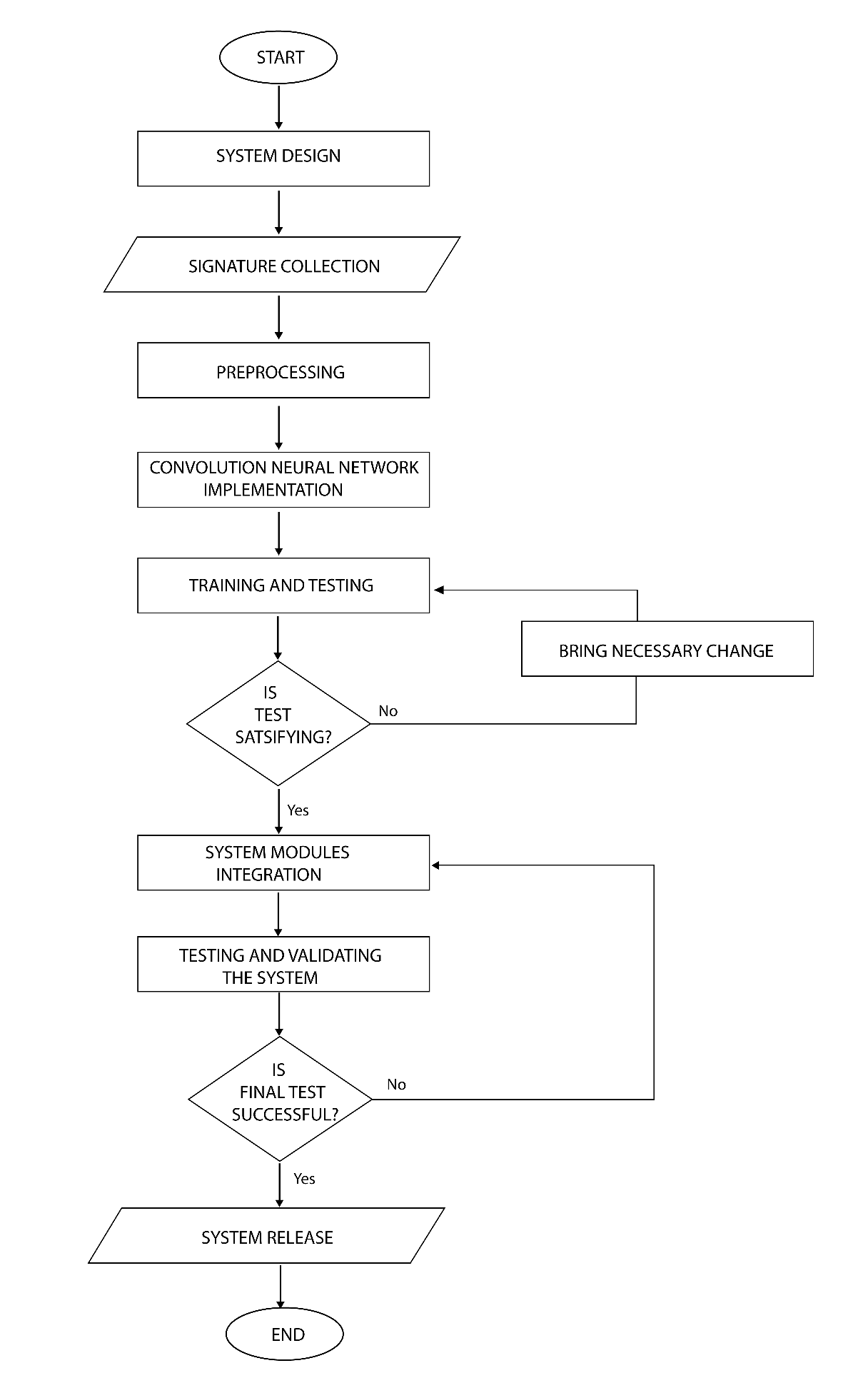
3.4 Project Workflow

Figure 3.4: Project Workflow

**System Design**

To build our final system, we perform two tasks. One is to develop a CNN model which will be used to verify the signature. And next is developing GUI and integrating it with the model.

**Signature collection and Preprocessing**

In Data collection part, we plan to collect 100 signatures from an individual. The collected sample will be scanned, cropped and converted into grey scale image. The samples will be kept in labelled directory so as to import easily when training.

**Convolution Neural Network Implementation**

In this phase a CNN model is created using tensorflow, keras and inception v3, which will process our sample signature to validate the authenticity of signature.

**Training and testing**

The system will be trained with 80% of the signature data and tested with rest. If expected accuracy meets, we conclude that the model is ready and we can now use it to verify the signature where the user can import signature to verify.

**System Modules Integration**

Once the Training and testing is done, we now integrate the GUI platform and the trained model for the user. This will be the final software that a user can use to train the model and then validate the signature after it is trained.

**Testing and validating the system**

Here we test the overall functionality and the accuracy of the application. If it meets desired criterion it will be finalized else necessary modifications will be performed.

**System release**

If everything works properly then we conclude that our system is working as expected. Now we can release this system for users who will be training and also validating their signatures.

## 3.5 TOOLS TO BE USED

### 3.5.1 Python

Python is a high-level, general purpose programming language. Python is popular these days as it is easy to understand and is close to our English language. Also, it has various modules and frameworks which can be used to build entire system such as frontend interface, database and backend. For machine learning, python is one of the popular languages being used because of available libraries and frameworks which supports for data analysis, machine learning and neural networks.

### 3.5.2 TensorFlow

TensorFlow is a free and open-source software library, developed by google for machine learning and artificial intelligence. It can be used across a range of tasks and has a particular focus on training and inference of deep neural networks. We will be using TensorFlow 2.0 which is the latest version of TensorFlow.

### 3.5.3 Keras

Keras is a deep learning API written in Python, running on top of the machine learning platform TensorFlow. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to doing good research.

### 3.5.4 Pandas

Pandas is one of the open-source python packages built on top of NumPy. Pandas has been one of the most commonly used tools for Data Science and Machine learning, which is used for data cleaning and analysis. Here, Pandas is the best tool for handling this real-world messy data.

### 3.5.5 NumPy

NumPy stands for Numerical Python. It is a Python library used for working with arrays. It also has functions for working in domain of linear algebra, Fourier transform, and matrices along with a large collection of high-level mathematical functions to operate on these arrays.

### 3.5.6 Inception V3

Inception V3 is a convolutional neural network for assisting in image analysis and object detection. The design of Inceptionv3 was intended to allow deeper networks while also keeping the number of parameters from growing too large: it has "under 25 million parameters".

### 3.5.7 Tkinter

Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's de facto standard GUI. The name Tkinter comes from Tk interface. This will be used to create a user interface in our system.

# CHAPTER 4: EPILOGUE

## 4.1 EXPECTED OUTPUT

The system will be a simple computer software with two options. First one is importing the samples of signature to train the model. Another option is importing the signature to be authenticated. After the user imports the signature, it is supplied as input to the trained model. Then the output will be the accuracy of the input signature compared with respect to the trained sample signatures.

The decision of accepting or rejecting the signature can be performed manually by the user. For machined verification certain criterion can be set depending upon the match of signature. The criterion may be tight for critical needs and loose for general needs.

## 4.2 GANTT CHART



Figure 4.2: Gantt Chart

# REFERENCES

[1] A. Pansare and S. Bhatia, "Off-line Signature Verification Using Neural Network," International Journal of Scientific Engineering Research , vol. 3, no. 2, 2012.

[2] G. Alvarez, B. Sheffer and M. Bryant, "Offline Signature Verification with Convolutional Neural Networks," Stanford.edu, Stanford, California, 2016.

[3] M. Shirdhonkar and M.Kokare, "Off-Line Handwritten Signature Retrieval using Curvelet Transforms," International Journal of Computer Science and Information Security, vol. 8, 2010.

[4] Software Engineering I. Sommerville. Addison-Wesley, Harlow, England, 9 edition, (2010)

# BIBLOGRAPHY

* <https://www.techopedia.com/definition/30499/signature-verification>
* <https://en.wikipedia.org/wiki/Signature>
* <https://en.wikipedia.org/wiki/Signature#Detection_of_forged_signature>
* Armitage, Susie (2018-11-05). "Handwriting Disputes Cause Headaches for Some Absentee Voters". ProPublica. Retrieved 2020-06-01
* <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53?gi=9059c1ff69b7>
* <https://sqnbankingsystems.com/blog/how-to-improve-forgery-detection-at-your-financial-institution/>