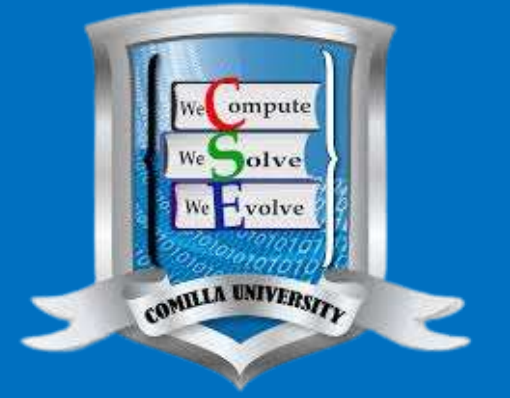# SIGNATURE FORGERY DETECTION

Team Members: Tanmoy Mandol, ID : 11808044 And Hakim Mohammad Insaf, ID : 11808045.

Supervisor: Mahmuda Khatun, Assistant Professor, Dept. of CSE, Comilla university.

INTRODUCTION

➢ Signature verification is an essential aspect of document authentication in various domains, including banking, legal, and government organizations.
➢ In recent years, image processing techniques have shown great potential in the field of signature verification.
➢ These techniques allow for the automatic extraction of features from signature images, which can then be used to verify the authenticity of a signature.

OBJECTIVES

➢ Develop an Intelligent Forgery Detection System
➢ Enhance Forgery Recognition Accuracy
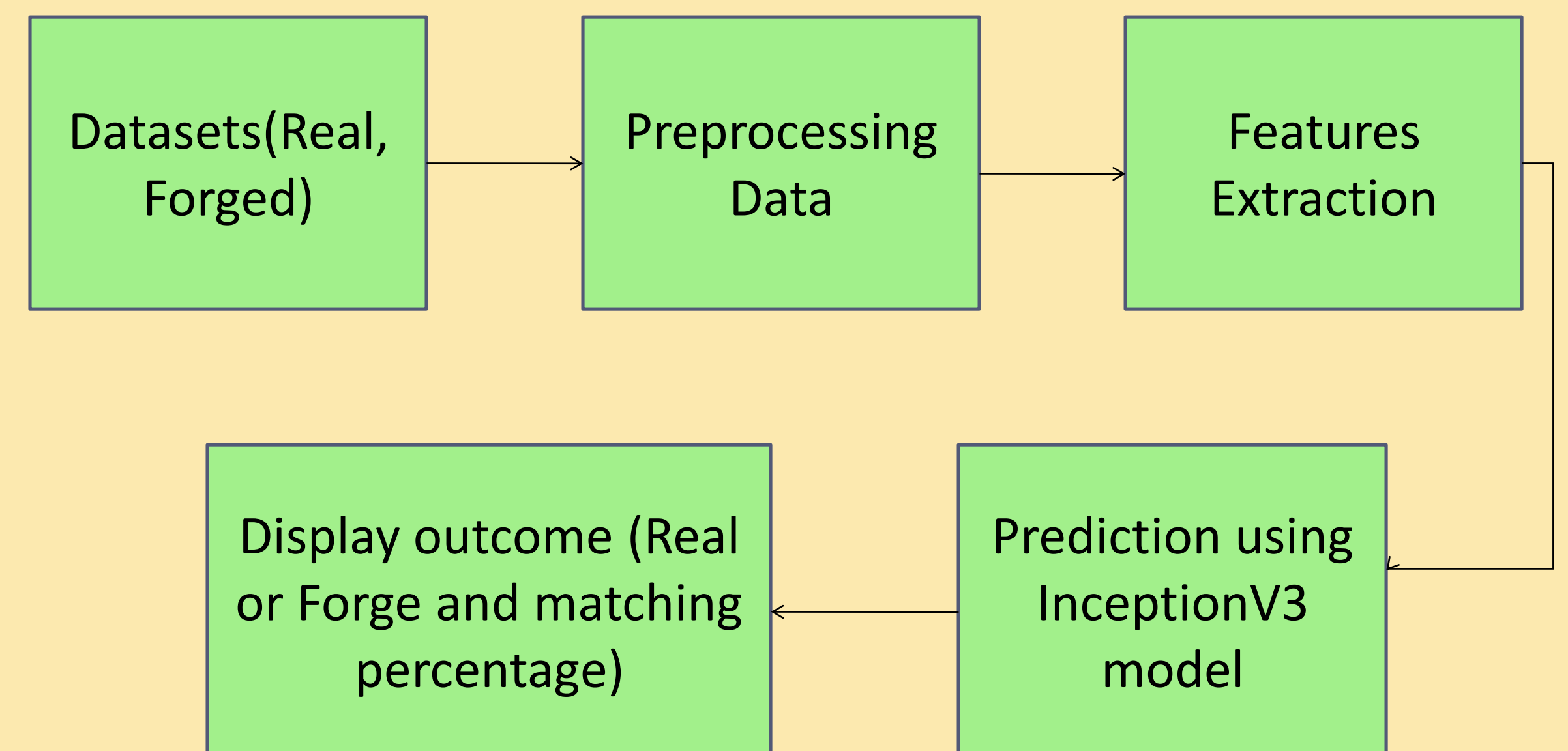➢ Real-World Application and Integration

METHODOLOGY

Datasets(Real, Forged) → Preprocessing Data → Features Extraction

Display outcome (Real or Forge and matching percentage) ← Prediction using InceptionV3 model

Fig: Model training using InceptionV3

LITERATURE SURVEY

SL. no	Paper title	Authors	Description
1	Signature Forgery Detection using machine learning.	Ms. Manjula Subramaniam Teja E , N Arpith Mathew.	We intend to use a Convolutional Neural Network (CNN) .55 individuals contributed 24 signatures each and hence the dataset consists of 1320 genuine signatures.
2	Digital signature Forgery Detection using CNN.	Lakkoju Chandra Kiran, Gorantla Akhil Chowdary, Manchala Shalem Raju, Kondaveeti Gopi Krishna.	We make use of CNNs in this program. We have 12 users, so we have a model that can estimate 60 groups. The highest accuracy we got was 99.7% . The average accuracy is about 97.8%.
3	Deep neural network using CNN for detect forgery signature.	Neha sharma.	Use dretrative model (vgg16, vgg19, ResNet50, Mobile net, Efficient Net). Accuracy 95%. Used Google colab for implementation.
4	Detect Forgery signature using CNN.	S Gldeon, Anurag Kandulna, Aron Kujur, A Diana.	Using Tensorflow backed to build the model. Used kaggle dataset. Used Softmax Activation function in fully connected layer. Accuracy =99.7%.

RESEARCH GAP WITH CONTRIBUTION

Research Gap:
Limited focus on real-time detection and integration into practical applications.
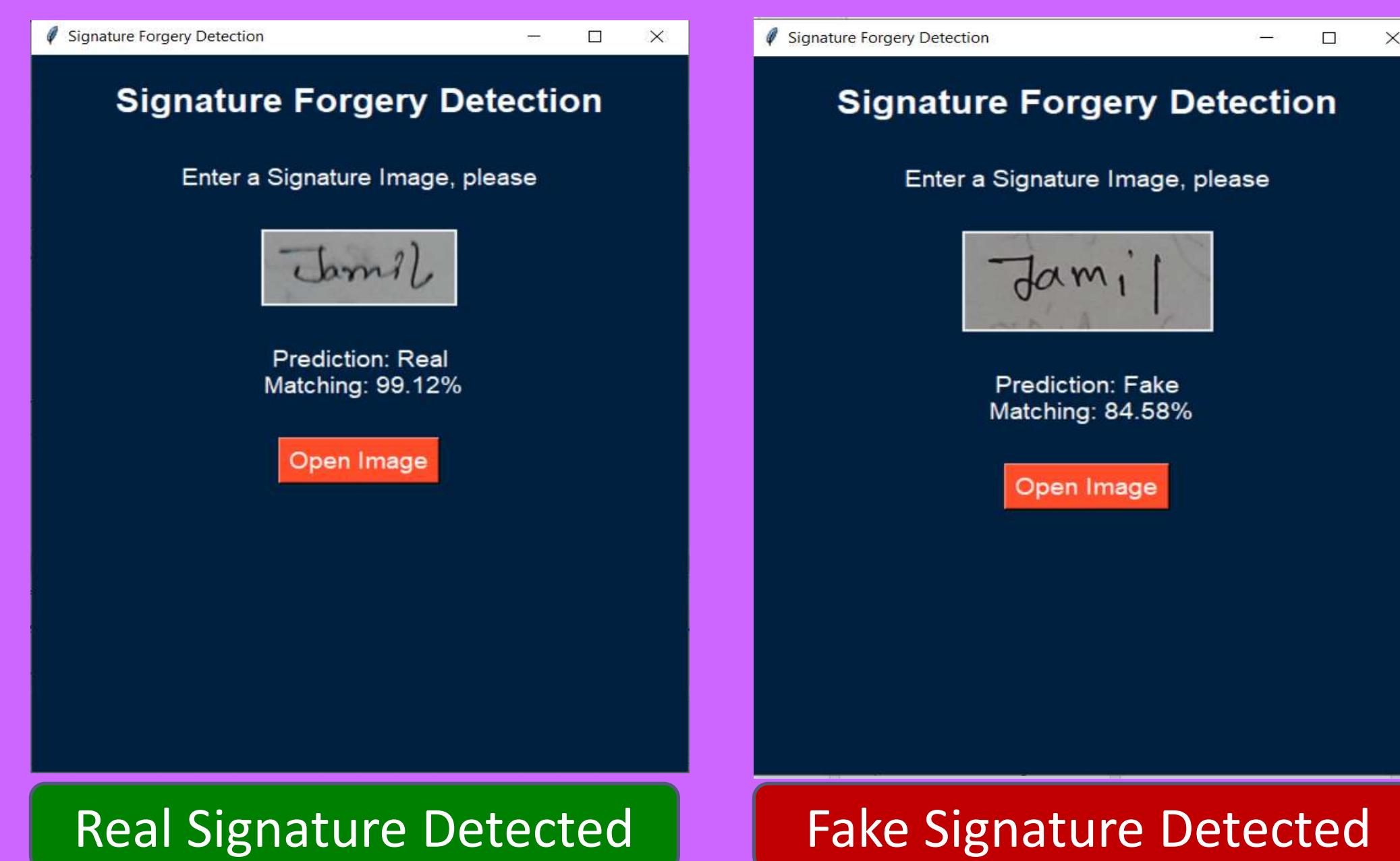**Contribution:**
➢ Optimizing the model for real-time detection.
➢ Development of a user-friendly interface for easy upload and feedback on signature authenticity.
➢ Bridging the gap between research and practical application in various domains.
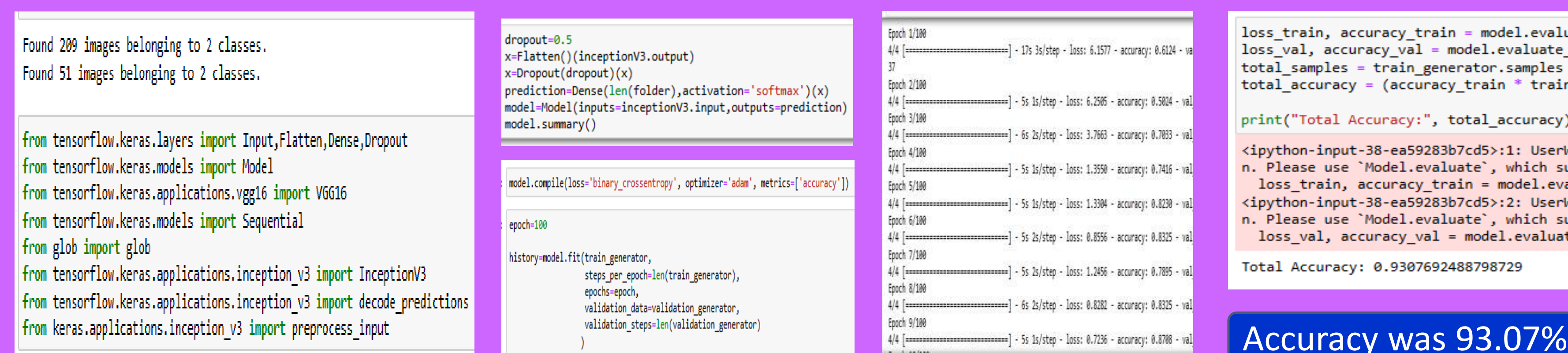
DATASET

Capture3.PNG	Capture4.PNG	Capture5.PNG	Capture6.PNG
Capture11.PNG	Capture12.PNG	Capture13.PNG	Capture14.PNG
Capture19.PNG	Capture20.PNG	Capture21.PNG	Capture22.PNG
Capture51.PNG	Capture52.PNG	Capture53.PNG	Capture54.PNG
Capture59.PNG	Capture60.PNG	Capture61.PNG	Capture62.PNG
Capture67.PNG	Capture68.PNG	Capture69.PNG	Capture70.PNG

RESULTS

Signature Forgery Detection
Enter a Signature Image, please
Prediction: Real
Matching: 99.12%
Open Image

Real Signature Detected

Signature Forgery Detection
Enter a Signature Image, please
Prediction: Fake
Matching: 84.58%
Open Image

Fake Signature Detected

PROPOSED METHOD (FLOW CHART)

Signature Data Acquisition
Training Data — Test Data
Preprocessing
Feature Extraction
Model Training Using InceptionV3
Predict Signature
Accept/ Reject

Fig: Flowchart

CODING & TRAINING ACCURACY

```
Found 209 images belonging to 2 classes.
Found 51 images belonging to 2 classes.

from tensorflow.keras.layers import Input,Flatten,Dense,Dropout
from tensorflow.keras.models import Model
from tensorflow.keras.applications.vgg16 import VGG16
from tensorflow.keras.models import Sequential
from glob import glob
from tensorflow.keras.applications.inception_v3 import InceptionV3
from tensorflow.keras.applications.inception_v3 import decode_predictions
from keras.applications.inception_v3 import preprocess_input
```

```
dropout=0.5
x=Flatten()(inceptionV3.output)
x=Dropout(dropout)(x)
prediction=Dense(len(folder),activation="softmax")(x)
model=Model(inputs=inceptionV3.input,outputs=prediction)
model.summary()

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

epoch=100
history=model.fit(train_generator,
        steps_per_epoch=len(train_generator),
        epochs=epoch,
        validation_data=validation_generator,
        validation_steps=len(validation_generator)
        )
```

```
loss_train, accuracy_train = model.evalua
loss_val, accuracy_val = model.evaluate_g
total_samples = train_generator.samples
total_accuracy = (accuracy_train * train_

print("Total Accuracy:", total_accuracy)

<ipython-input-38-ea59283b7cd5>:1: UserWa
n. Please use `Model.evaluate`, which sup
  loss_train, accuracy_train = model.eval
<ipython-input-38-ea59283b7cd5>:2: UserWa
n. Please use `Model.evaluate`, which sup
  loss_val, accuracy_val = model.evaluate

Total Accuracy: 0.9307692488798729
```

Accuracy was 93.07%

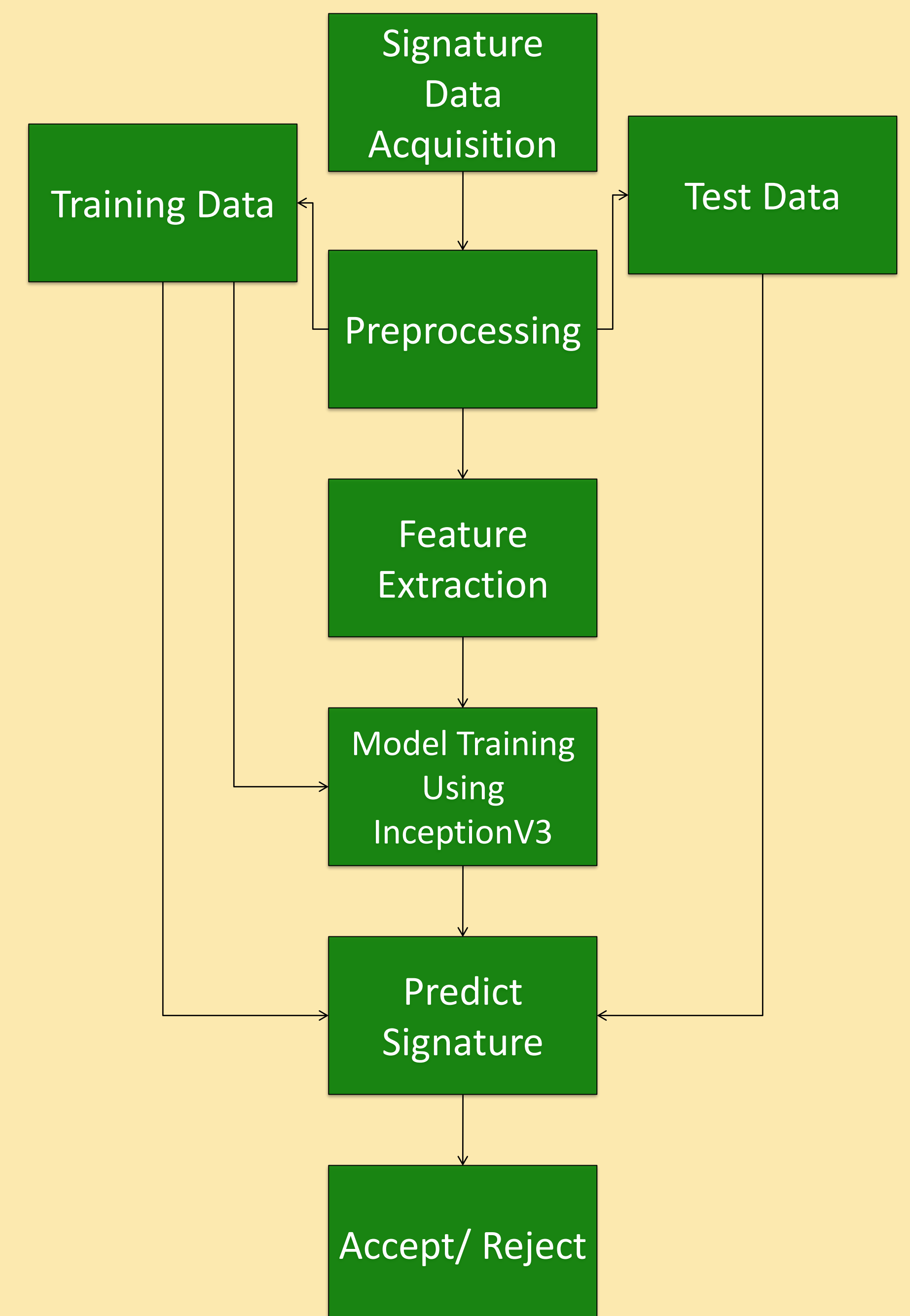## FUTURE WORK

➢ Enhancing Accuracy and Adaptability.
➢ User-Centric Approach.
➢ Real-World Deployment and Collaboration.

CONCLUSION

➢ The project developed an intelligent system for detecting forged signatures.
➢ A user-friendly interface ensures easy access and user trust.
➢ Real-time implementation enhances document authentication and security.
➢ Future work includes multi-modal analysis and collaboration for continuous improvement.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, COMILLA UNIVERSITY