Signature Verification

Capstone Project: XiuTing

PROBLEM STATEMENT

Handwritten signatures are frequently used for personal identification and verification.

Challenges of detecting forged signatures:

- “No two signatures of the same person are exactly the same”
- Variability in pens used
- Noisy background

METHODOLOGY

01 DATASET

Look for usable dataset

02 PROCESS

Pre-process data
Create Lists of Signatures and binary Labels

03 EDA

View and explore images

04 MODEL

Build models

05 TUNE

Fine-tune optimizers and model parameters

05 EVALUATE

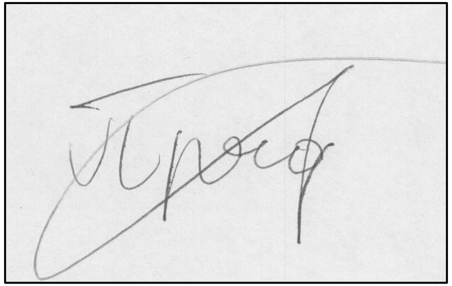
Evaluate models

55 sets (per person) of
20 forged and 20 signatures

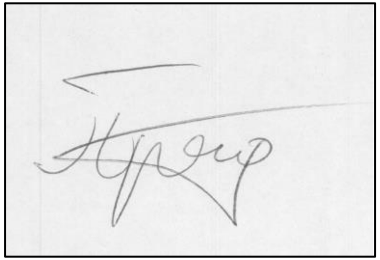
DATASET

GENUINE / FORGED SIGNATURES ?

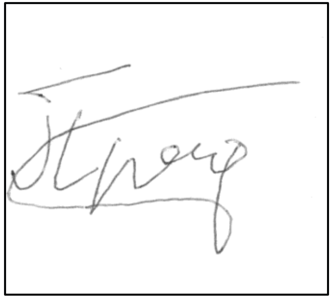
01



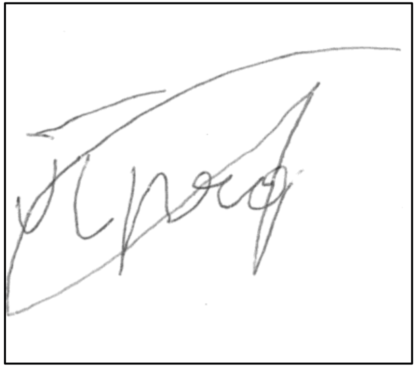
03



02



04

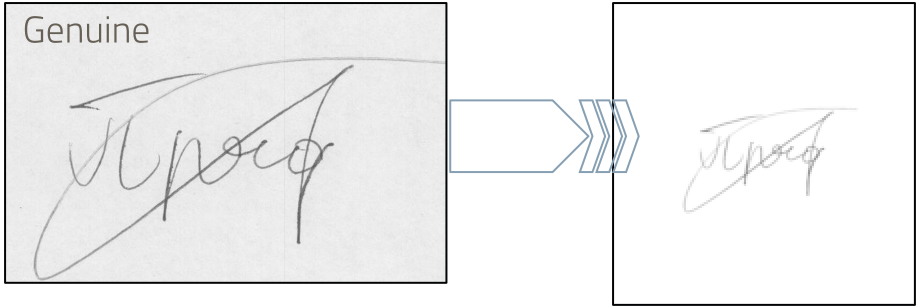


Why pre-processing is required:

- Signature not in the center
- Background noise
- Variability in image size

IMAGE PRE-PROCESSING

01



02

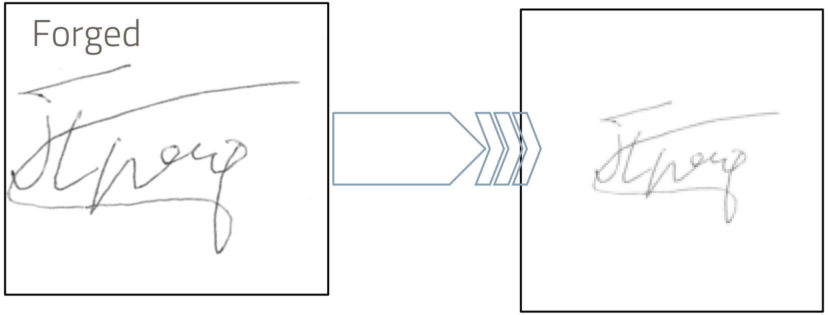
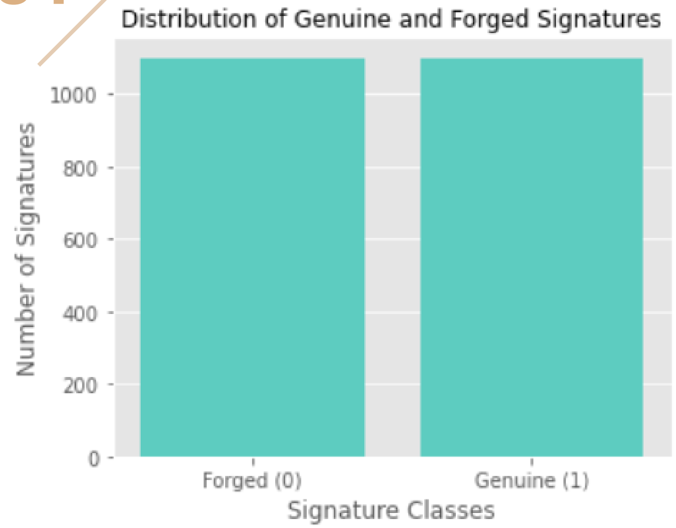


Image pre-processing steps

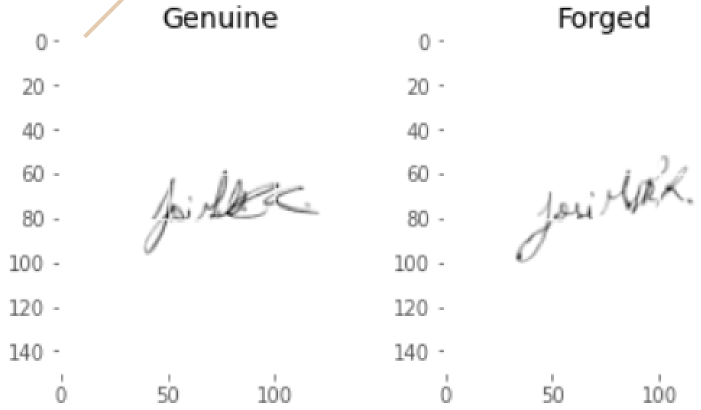
- Image changed to grayscale / single channel.
- Background is removed
- Gaussian filter applied
- Centralized image
- Image is cropped and resized

EDA

01 DATA DISTRIBUTION



02 VISUALIZE SIGNATURES



MODELING: NEURAL NETWORK

A

CNN

Covolutional Neural Network
Insufficient data is likely to cause over fitting of the model

B

SNN

Siamese Neural Networks
Good for training with insufficient data

C

PRE-TRAINED

InceptionV3
Experimented with it
Requires more tuning

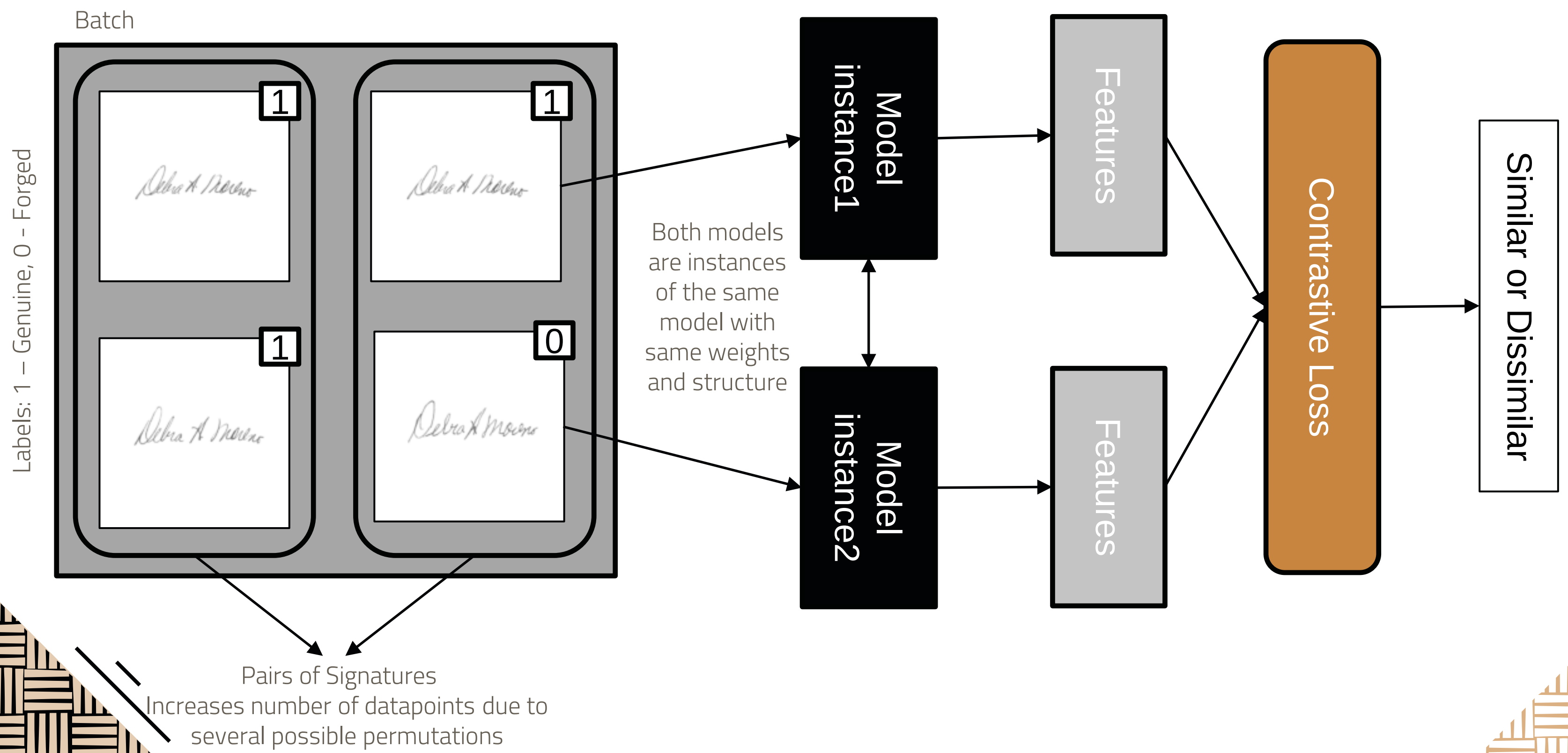
INCREASING DATAPOINTS

Genuine-Genuine Pairs
20 genuine signatures per pax:
20 choose 2 = 190 Genuine-Genuine image pairs for one person

Genuine-Forged Pairs
Pair every 1 genuine signature of a person with 20 randomly sampled Forged signatures of the same person.
20 * 20 = 400 Genuine-Forged image pairs per person

In all we have 55 person's data in the training data.
Total no. of Genuine-Genuine pairs = 55 * 190 = 10450
Total number of Genuine-Forged pairs = 55 * 400 = 22000
Total no. of data points = 10450 + 22000 = 32450

SIAMESE NEURAL NETWORK



MODEL TUNING

01 OPTIMIZER

1. Adam
2. Adagrad
3. SGD
4. RMSprop

02 LEARNING RATE

1. 1e-4
2. 1e-5
3. 1e-6

03 BATCH SIZE

32, 64, 256

04 EPOCHS

EarlyStopping

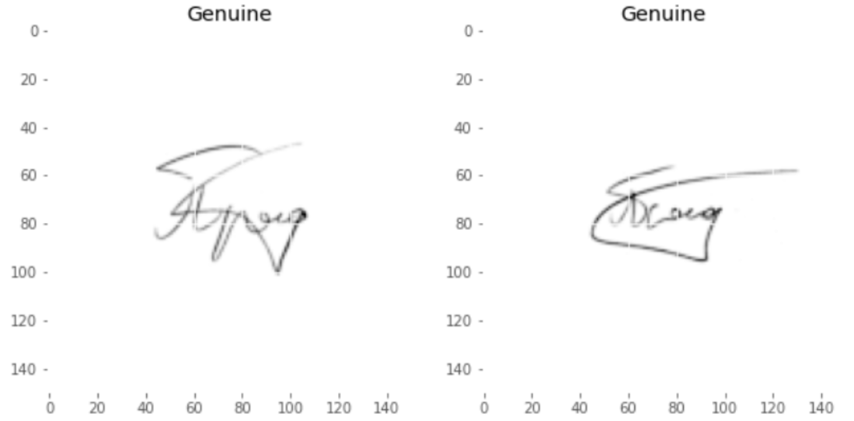
MODEL EVALUATION

Model No.	Optimizer	Batch Size	Learning Rate	Epoch Number	Accuracy (%)
1	Adam	32	0.0001	13	65.6
2	Adam	64	0.0001	24	69.3
3	Adam	256	0.0001	8	64.8
4	Adam	32	1e-05	7	65.9
5	Adam	64	1e-05	79	70.4
6	Adam	256	1e-05	2	71.4
7	RMSprop	32	0.0001	8	72.2
8	RMSprop	64	0.0001	11	69.2
9	RMSprop	256	0.0001	25	72.1
10	RMSprop	32	1e-05	3	69.9
11	RMSprop	64	1e-05	18	68.2
12	RMSprop	256	1e-05	0	70.7

Model 7 has the highest accuracy score.

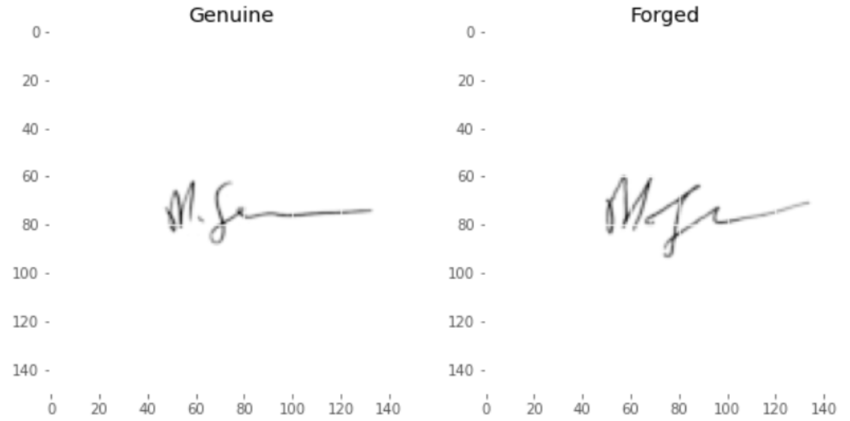
PREDICTIONS

```
1 verify_signature()
```




```
Difference Score =  0.0650725
The signature is genuine
```

```
1 verify_signature()
```




```
Difference Score =  0.55037856
The signature is forged
```

Accuracy: **72.2%,** Threshold: **0.1519**

FUTURE WORKS

1. Build model using triplet loss function on Siamese Neural Networks (Current models built on contrastive loss)
2. Change metrics to precision
3. Look into pre-trained models and fine-tune the model more.
4. Deploy on web application

QUESTIONS