Fingerprint detection

Boros Réka. Czakó Adrienn,
Bárdos-Deák Botond, Kránitz Bence

1. Aim and basics

Fingerprint recognition is one of the most commonly used methods in biometric identification today. In this project, we'll be classifying fingerprints based on different labels. Our goal is to figure out the gender linked to a given fingerprint, identify which finger it belongs to, determine whether it's from the left or right hand, and accurately match it to the person it came from. We'll also explore whether these features can be identified using only the fingerprint itself. [1]

The goal of the project is to identify the following based on a given fingerprint:

1. Female or male
2. Right or left hand
3. Which finger
4. Which person

For implementation, we used the Sokoto Coventry Fingerprint Dataset [2],available on Kaggle . This dataset contains fingerprints from 600 individuals, resulting in approximately 60,000 images after various modifications. These modifications include central rotation, obliteration, and Z-cut.

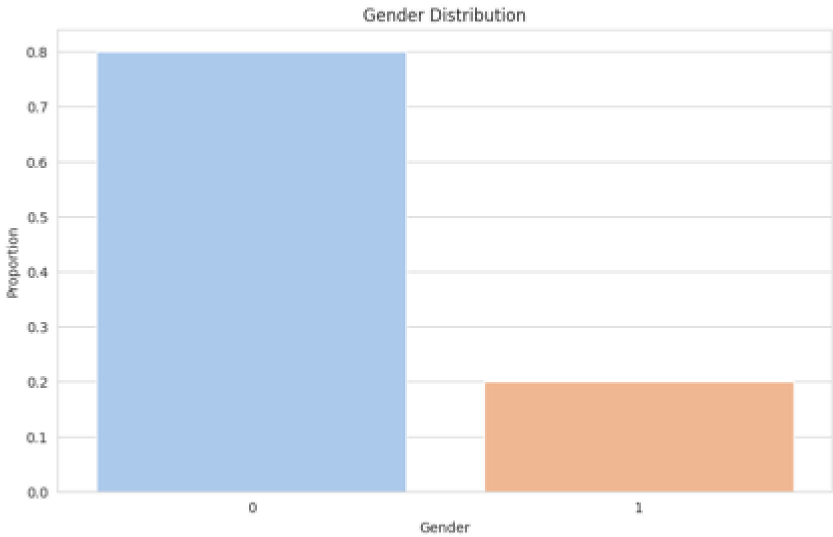
Each image can be identified and its data extracted based on its filename. Specifically, the naming format includes:

- The person's unique ID
- M/F indicating male or female
- LEFT/RIGHT indicating whether the fingerprint is from the left or right hand
- Index/Thumb/Middle/Ring/Little specifying which finger is shown in the image

After loading the data, we quantified these details to enable proper modeling.

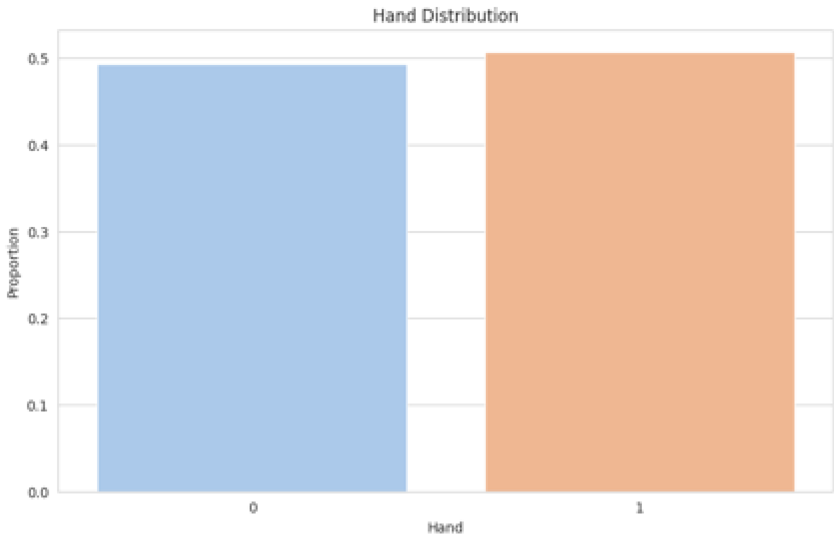
Following the data loading and encoding process, the dataset was organized as follows:

1. Gender ratio



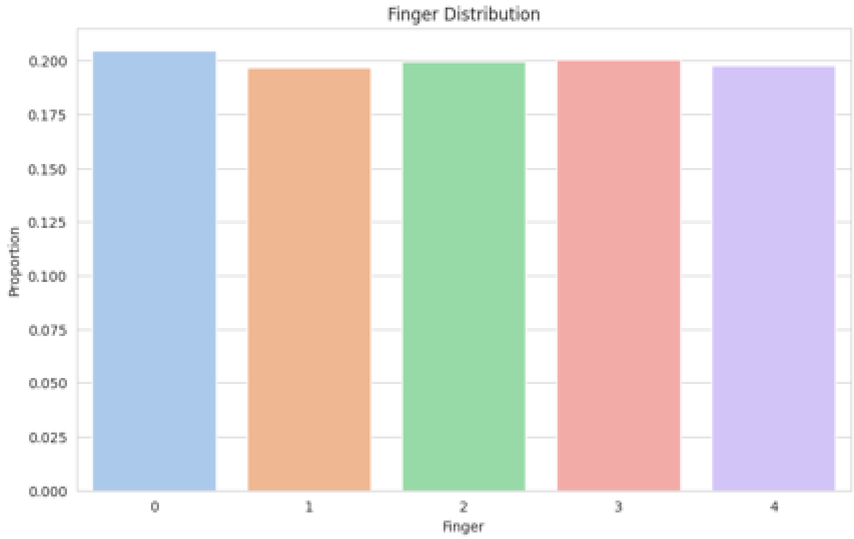
Where 0 represents male and 1 represents female.

2. Hand ratio



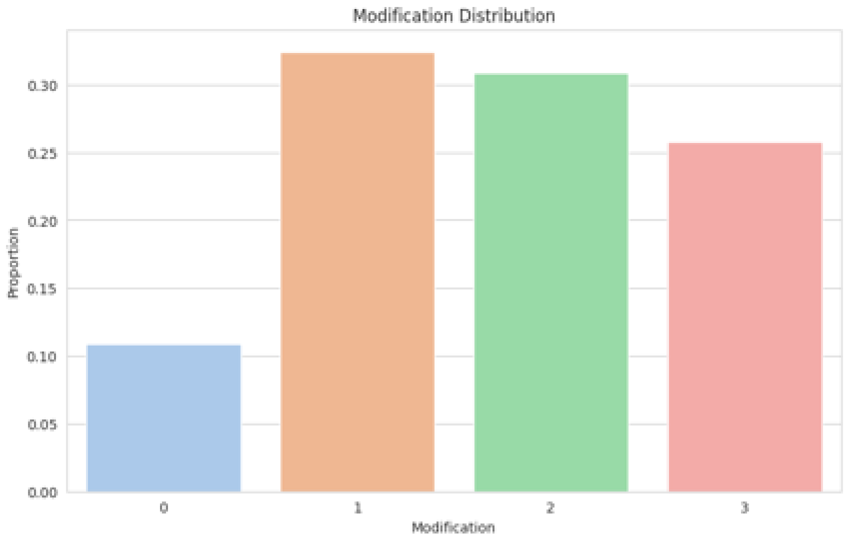
Where 0 indicates the right hand and 1 indicates the left hand.

3. Finger ratio



In the image above, the numbers correspond to the following fingers: thumb, index, middle, ring, and little finger.

4. Distribution of Different Transformations



The numbers represent different types of transformations applied to the fingerprints. A value of 0 indicates that the image is unmodified, while 1 signifies central rotation, 2 denotes obliteration, and 4 corresponds to the Z-cut transformation.

2. Modelling process

During the modeling process, we encountered several issues. One such problem was that the image frame was also rotated during transformations, causing it to appear at the center of the fingerprint. As a result, we had to crop the images before applying any transformations.

Another issue we faced was the gender ratio, as shown in the chart above, where only 20% of the samples were female. Initially, we attempted to address this through data augmentation, but it did not improve the final accuracy.

While searching for a solution, we came across a study suggesting that there is no statistically significant difference between male and female fingerprints. Another article [3] mentioned that men can be identified by their right thumb and women by their left little finger. However, even after training the model using these insights, we did not achieve any significant improvement in accuracy.

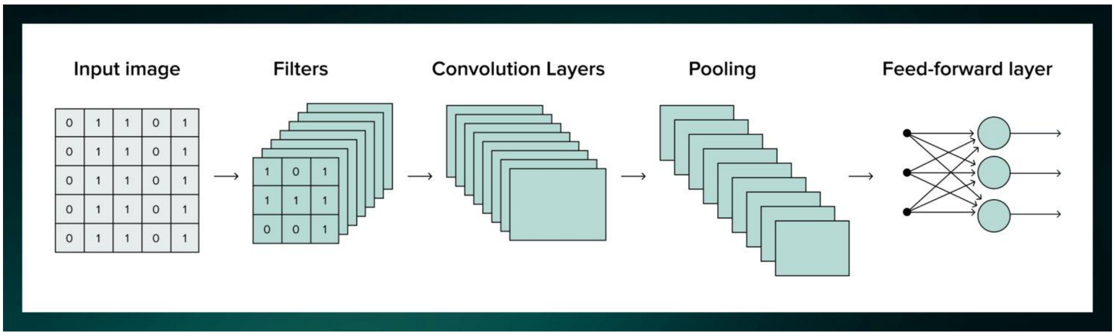
The main issue we worked on improving was the model's accuracy. Initially, we attempted to augment the data using custom transformations, but this proved insufficient since only 11% of the dataset remained original. As a result, we replaced the baseline model with a pre-trained CNN model, experimenting with different parameters.

During the project, we experimented with several methods to improve the model’s performance and accuracy. The first step was data splitting, where we divided the dataset into training, validation, and test sets. This ensured that the model could learn effectively while allowing us to evaluate its performance on new, unseen data.

Next, we tested various algorithms for data analysis and classification. The first method we tried was the k-Nearest Neighbors (KNN) algorithm, which makes classification decisions based on sample similarity. While this approach is simple and often effective for smaller datasets, its results were unsatisfactory in this case.

We then tested the Random Forest algorithm, an ensemble method that uses multiple decision trees to improve predictions. This approach yielded somewhat better results than KNN, but the accuracy still fell short of our expectations.

Finally, we turned to deep learning techniques and built a Convolutional Neural Network (CNN) for the task. This model was better equipped to learn complex patterns from image data and ultimately achieved 70% accuracy in the classification task. Although this was a significant improvement over previous methods, the accuracy was still not high enough for us to fully trust the results.



We aimed to train the model to predict gender and determine whether a given image depicts a left or right hand. To achieve this, we split the data based on individual IDs. The first 400 individuals' data were used for the training set, IDs from 400 to 500 for the validation set, and IDs from 500 to 600 for the test set.

Next, we tested the model's ability to correctly identify individuals by their IDs. For this, we selected one image from each individual in the test and validation sets, while the remaining images stayed in the training set. This approach ensured that the model had enough data for learning while still being tested on new, unseen images. This strategy allowed us to evaluate the model not only in predicting gender and hand sides but also in identifying individuals.

To prepare the data for efficient training, we applied several image preprocessing steps. We applied various transformations to the training set to increase data diversity and improve the model's generalization ability.

The transformations included:

- Horizontal Flip: Applied with a 50% probability to mirror the images.
- Affine Transformation: Applied with a 30% probability to adjust the images' perspective.
- Gaussian Blur: Used with a 20% probability to reduce image sharpness.
- Grid Distortion: Applied with a 40% probability to alter the images' structure.
- Random Brightness and Contrast Adjustments: Applied with a 40% probability to vary the images' lighting conditions.

Additionally, all images were resized to 112×112 pixels to fit the model's input requirements. These preprocessing steps ensured that the training set contained a wide range of image variations, contributing to a more robust model capable of better performance on unseen data.

```
def img_trf_train(code):
    try:
        img=cv2.imread(str(code))#
        #add transform
        transform=A.Compose([
        A.HorizontalFlip(p=0.5),
        A.Affine(scale=(0.9,1.1), shear=(-5,5), rotate=(-10,10), p=0.3,fit_output=False,keep_ratio=True),
        A.GaussianBlur(blur_limit=(3, 7), sigma_limit=0, p=0.2),
        A.GridDistortion (num_steps=5, distort_limit=(-0.3, 0.3), interpolation=1, border_mode=2, p=0.2),
        A.RandomBrightnessContrast(p=0.4),
        A.Resize(112,112),
        ToTensorV2()])
        front_transformed=transform(image=img)["image"]
        return front_transformed

    except:
        z=torch.zeros(3,112,112)
    return z
```

We started by selecting a baseline model to establish initial reference results. For this, we used a simplified ResNet [4] model, a well-known and effective architecture for image classification tasks. This model allowed us to set basic performance benchmarks for evaluating classification accuracy.

For our final model, we used a pre-trained VGG16 [5] architecture from the PyTorch library. Pre-trained models have the advantage of possessing general pattern recognition capabilities learned from large datasets like ImageNet. Leveraging this benefit, we applied transfer learning to adapt the original network to our specific task.

As part of this process, we modified the network's final classifier layer to fit our classification tasks, such as predicting gender and identifying hand types.

This approach enabled us to utilize the strengths of existing deep learning models, allowing us to achieve good results faster and more efficiently. Using VGG16, we developed a stable and well-performing solution for our classification problem.

3. Conclusion and results

We evaluated the model's performance across various classification tasks and compared the results of the baseline model with the final model.

For gender classification, the baseline model achieved 79% accuracy, while our final model improved this to 81.76%. In recognizing the hand type (left or right), the baseline model scored 84% accuracy, which the final model increased to 89.84%. For finger classification, the final model achieved an accuracy of 66.46%, though we had no baseline model for comparison.

Regarding individual identification (ID recognition), the baseline model reached only 70% accuracy, whereas the final model demonstrated a significant improvement, achieving 94.25%.

	Baseline	Final
Gender	79	81.76
Hand	84	89.844
Finger	-	66.46
ID	70	94.25
Gender with two finger	-	82.7

As an additional experiment, we tested whether using only the thumb and little finger could improve gender classification, based on suggestions from the literature. This approach resulted in a slight improvement, reaching 82.7% accuracy, only about 1% higher than the traditional method.

Overall, the final model showed substantial improvements over the baseline in several classification tasks, particularly in individual identification. However, some areas, such as finger classification and gender prediction, still have room for improvement.

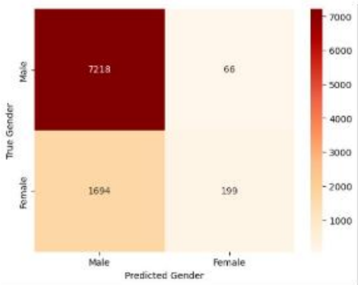
We evaluated the model's performance using confusion matrices, which provided a detailed view of classification accuracy and errors.

For gender recognition, the model classified males accurately (7,218 correct predictions), but made more frequent mistakes with females, misclassifying 1,694 female images as males.

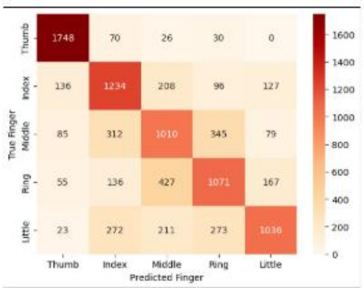
During finger classification, the model performed well in recognizing the thumb, but it often struggled to distinguish between the three middle fingers, leading to more frequent misclassifications.

For hand type recognition, the model performed exceptionally well, correctly identifying 4,147 right hands and 4,098 left hands.

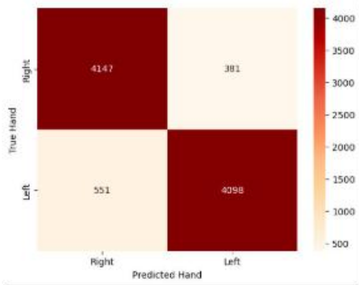
These results indicate that while the model excels in distinguishing hand types, further fine-tuning is needed for gender and finger classification, especially in improving the recognition of female samples.



Gender prediction



Finger prediction



Hand prediction

As a final step, we tested the model on our own fingers to evaluate its performance in real-world conditions. During the test, we analyzed images of different fingers and assessed the model's ability to correctly identify gender, fingers, and hand orientation.

Finger	Boti Right Thumb	Boti Right Index	Boti Right Middle	Boti Right Ring	Boti Right Litle	Bebe Left Thumb	Bebe Left Index	Bebe Left Middle	Réka Left Little	Réka Left Thumb
Gneder	Female	Male	Male	Male	Male	Male	Male	Male	Male	Male
Finger	Thumb	Ring	Ring	Ring	Index	Thumb	Ring	Ring	Thumb	Thumb
Hand	Left	Right	Right	Right	Right	Right	Right	Left	Left	Right

Reference and bibliography:

[1] https://github.com/m-mutti/fingerprint-from-image

[2] https://www.kaggle.com/datasets/ruizgara/socofing

[3] https://pubmed.ncbi.nlm.nih.gov/10423851/

[4] https://arxiv.org/pdf/1512.03385

[5] https://arxiv.org/pdf/1409.1556