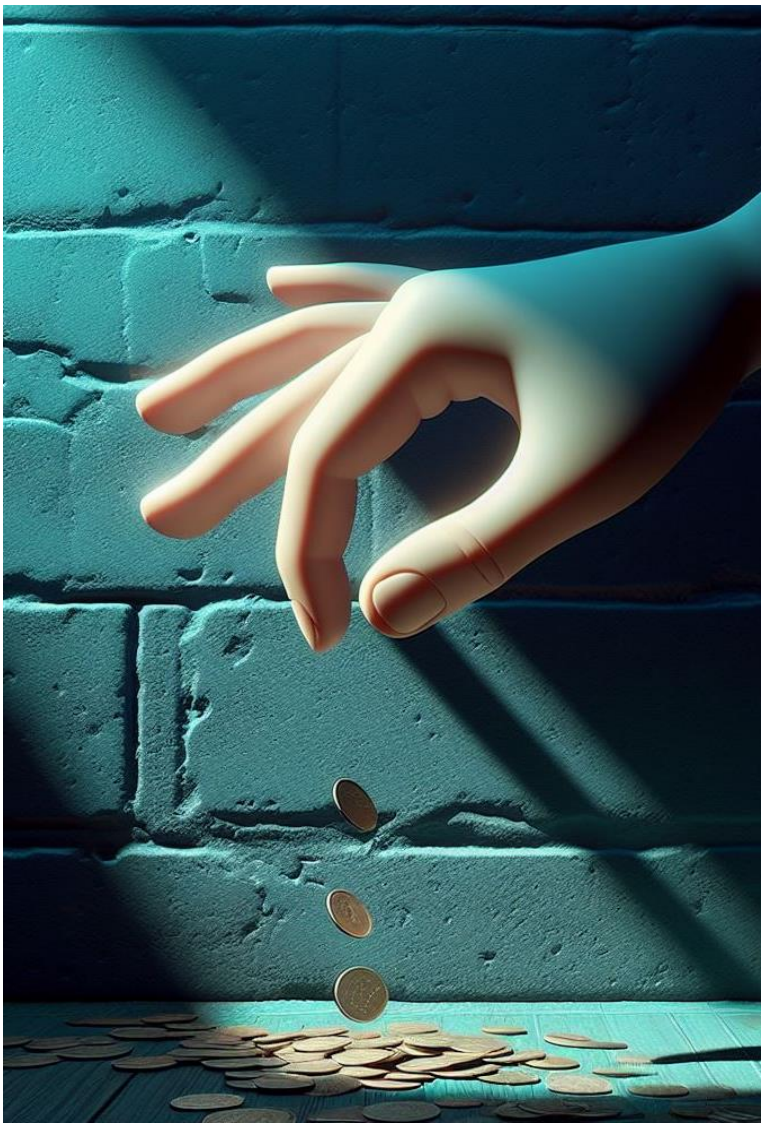

Enhanced with DALL-E

Position Prediction of a Dropping Coin

FINAL OF EHB 420E

Melisa Erce | Artificial Neural Network | 15.01.2024

Data Collection and Model

There are three initial positions for dropping the coin: head up, tail up, and vertical. The final position of the coin is determined by how it makes contact with the floor. Slow-motion footage of coin drops has been gathered, and each contact moment has been documented.

In total, 104 coin drop footage clips have been collected. All positions and the head-tail labels can be observed in the figure 1. Dark blue dots represent heads, while light blue dots represent tails.

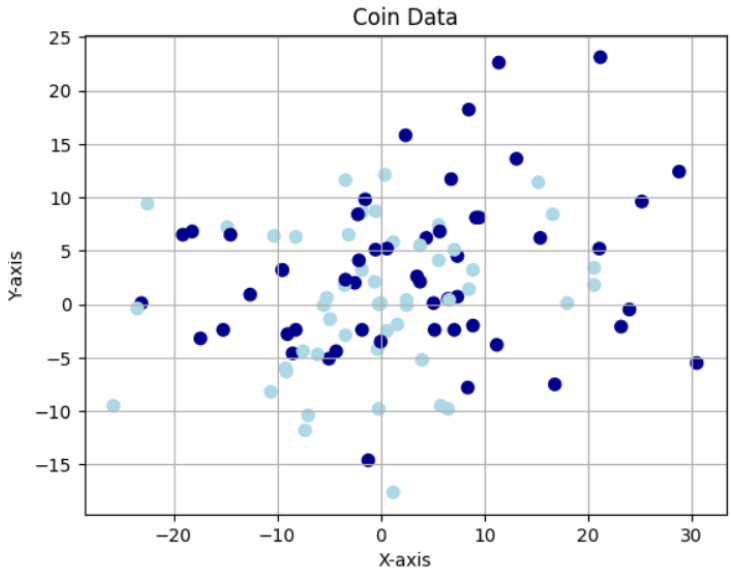

figure 1 Coin positions

If we plot according to the initial positions, we get these three plots.

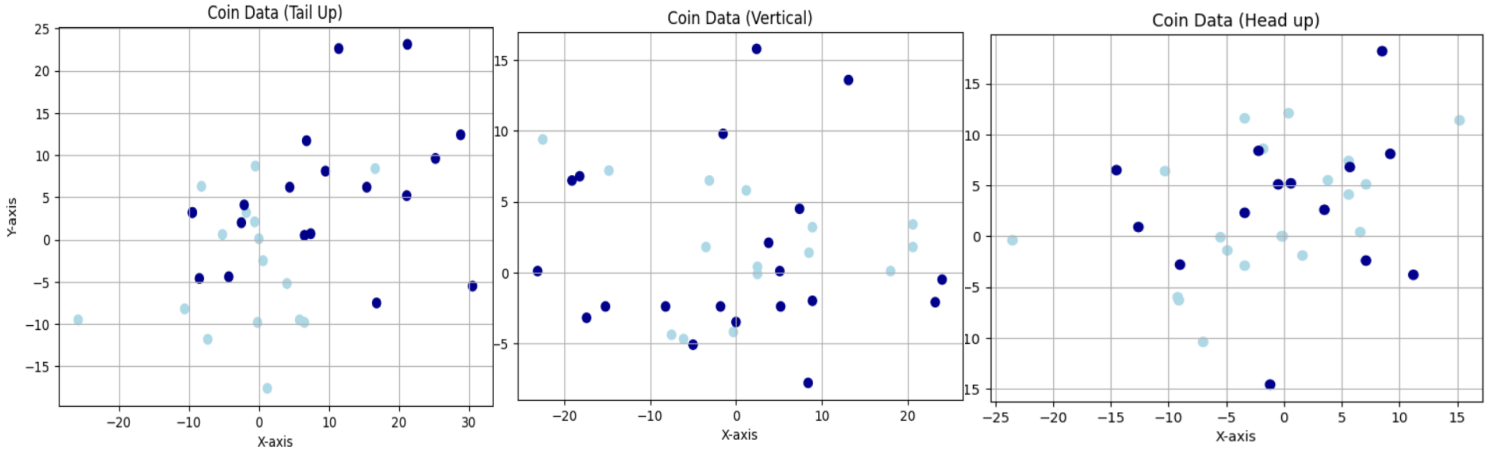

figure 2 Seperate coin drop positions according to initial positions

The initial position fails to provide sufficient information about the object's location. To address this limitation, slow-motion footage has been separately captured at the moment of contact with the floor. These recorded images will play a crucial role in determining the final position of the object. However, the data collection process proves to be challenging and time-consuming. Due to the constraints of time and insufficient data for training a CNN algorithm, data augmentation will be applied to the images. Additionally, enhancements such as increased crispness and light will be implemented to improve the quality of the captured pictures.

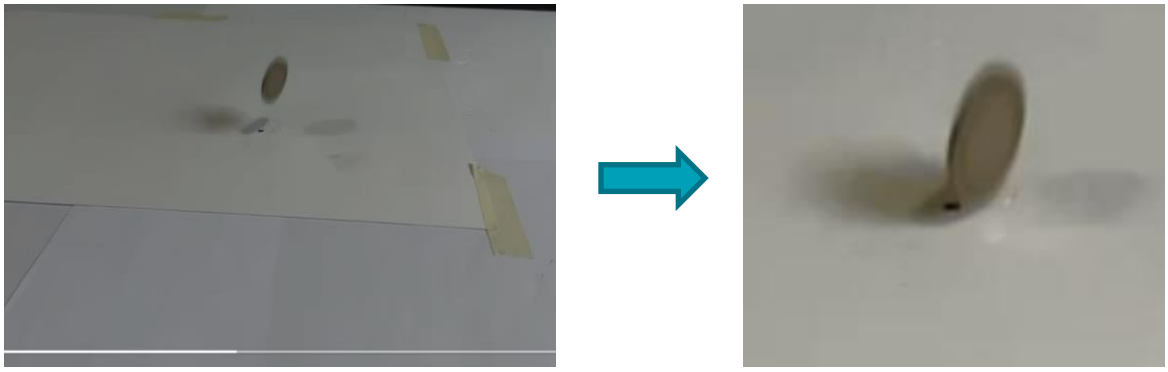

figure 3 Gathering the pictures

We have very little data. To prevent overfitting, we can use data augmentation. The generated data can be seen in Figure 4.

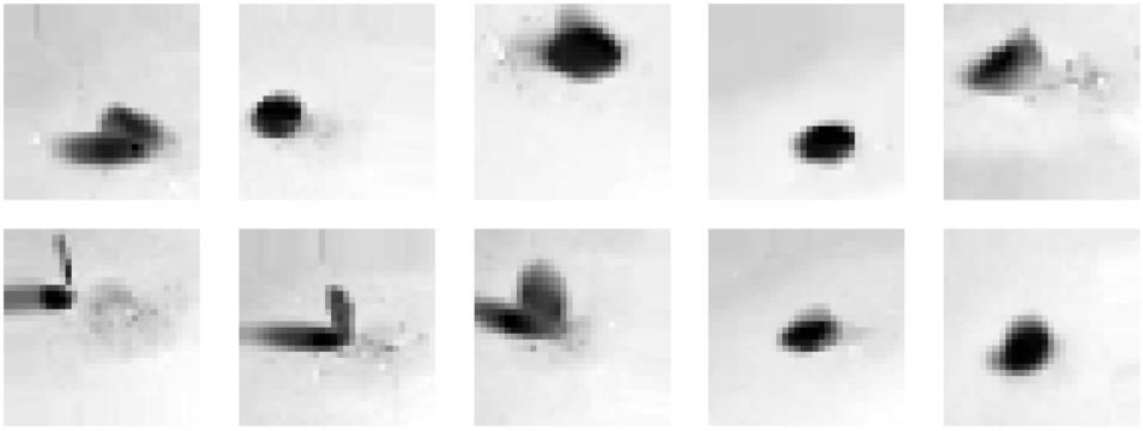

figure 4 Data augmentation

Unfortunately, with only 104 images, predicting the position has proven challenging. Regardless of the model selection, data augmentation, and various accuracy metrics, the results did not improve. It appears that using images may not be a suitable approach. Additionally, due to the small size of the dataset, the accuracy plot is inconsistent.

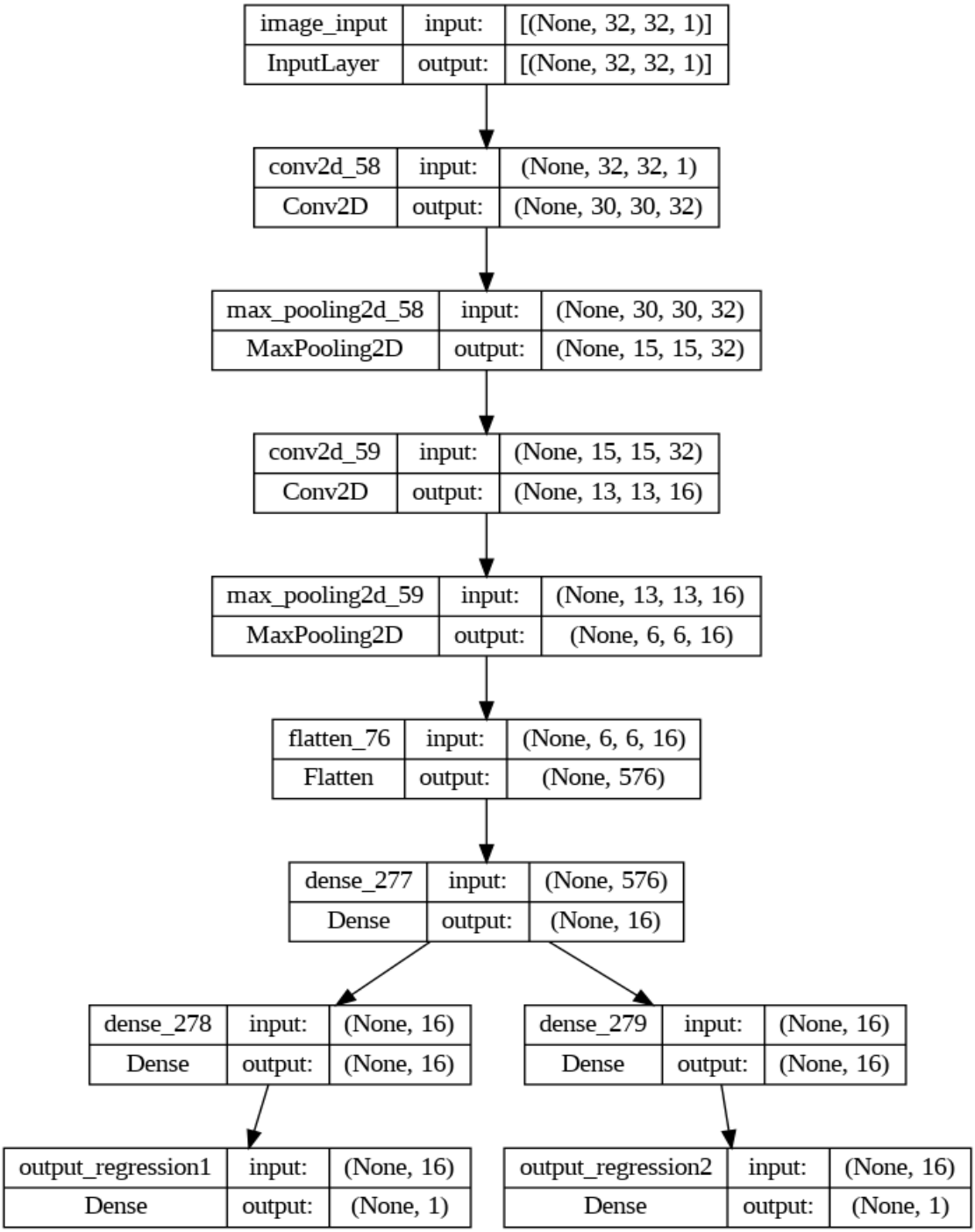

figure 5: The CNN Model

A mutli output CNN was thougth because the x and y data can be generated seperatly.

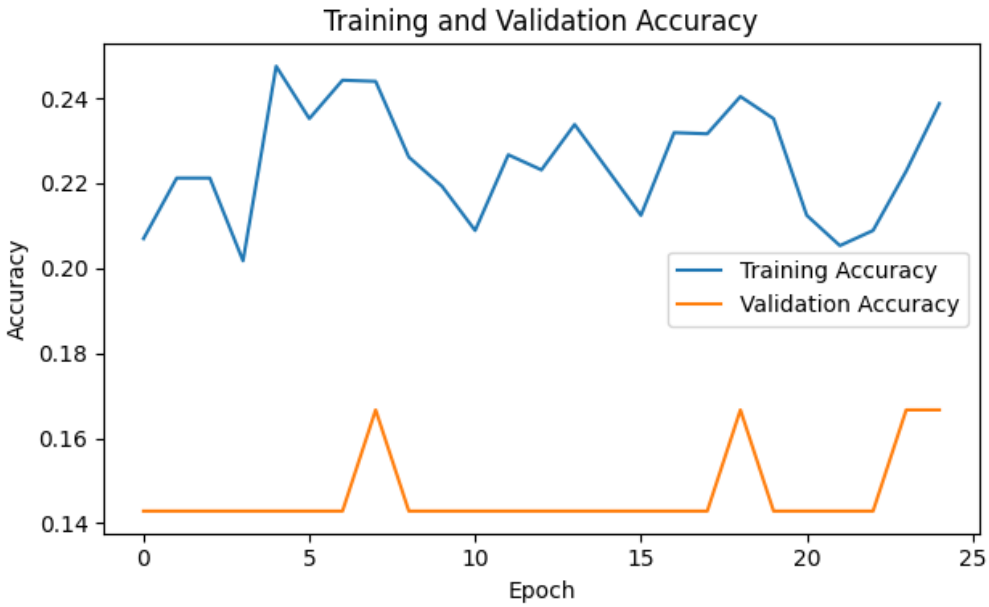

figure 6 Accuracy less than 20%

The training and test accuracy are determined when |y_pred – y_true| < 2. To achieve an 80% accuracy, the accuracy criterion needs to be |y_pred – y_true| < 10.5. This accuracy criterion only captures a few closely aligned data points. An example of such data points is [1.26, 1.35]. This point with a 10.5 radius is where the most of the coins have dropped.

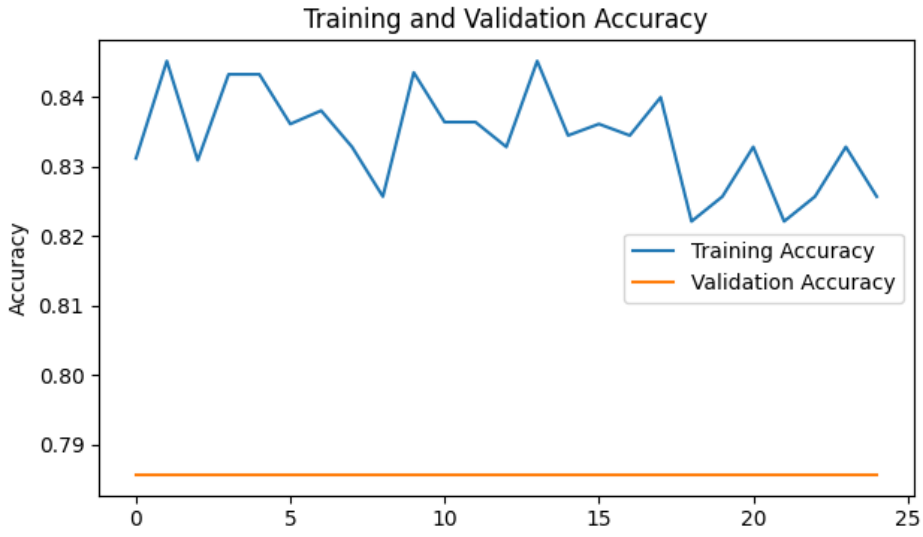

figure 7 Accuracy close to 80%

Also, the head or tail position was not solely determined by the coin's contact with the floor. Neither this model nor other smaller models I tried could even reach 50%. It wasn't even making random predictions. Trying another approach, I considered that the initial position of the coin might influence the end position, whether it is heads or tails.

```
model = Sequential()
model.add(Dense(8, input_dim=3, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(8, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(4, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(4, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))

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

```

figure 7: Model for head or tail prediction

To increase the number of features, I incorporated the coin-dropping position and the initial position. I labeled the tail-up position as 0, the vertical position as 1, and the head-up position as 2. As a result, the coin orientation becomes entirely random. According to probability theory, we understand that the probability of a coin toss, whether it lands as tails or heads, is random and equals 50%.

```
Epoch 25/30
14/14 [==============================] - 0s 3ms/step - loss: 0.6737 - accuracy: 0.5301
Epoch 26/30
14/14 [==============================] - 0s 2ms/step - loss: 0.6869 - accuracy: 0.5422
Epoch 27/30
14/14 [==============================] - 0s 2ms/step - loss: 0.6719 - accuracy: 0.5181
Epoch 28/30
14/14 [==============================] - 0s 2ms/step - loss: 0.6861 - accuracy: 0.5301
Epoch 29/30
14/14 [==============================] - 0s 2ms/step - loss: 0.6727 - accuracy: 0.5422
Epoch 30/30
14/14 [==============================] - 0s 2ms/step - loss: 0.6874 - accuracy: 0.5181
1/1 [==============================] - 0s 62ms/step
Test Accuracy: 47.62%
```

figure 8: Test accuracy

No matter the model, the coin drop head or tail probability found to be random as expected.

Conclusion

The data collection process was challenging, and achieving accuracy was not entirely feasible. I placed a floor covering to minimize the coin's jumping, but after each coin drop, the paper sustained damage, resulting in surface dents. This issue could have led some coins to deviate unexpectedly.

Moreover, the collected data was limited, requiring some adjustments. Initially, I considered using a dropping coin video, but the sample size was small, leading me to narrow down the data to focus solely on the moment of contact between the coin and the floor. However, this approach posed challenges, as slow-motion videos from a phone camera often lacked the desired quality. Some footage appeared blurry, and others failed to precisely capture the contact moment.

Despite experimenting with various model variations and data augmentation techniques, none proved sufficient for accurate predictions.

Other than using the pictures I tried using initial positions as feature ant tried using it to ıdentfy the potition. But only 3 position was not sufficient to guess the position. This can be seen from the figure 2.

Video:

In the video R is taken to be 11.5. İn the demonstration it can be seen that the algorithm only gives one point.

https://www.youtube.com/watch?v=wQpi9C-aae8

Codes:

If the directory of the photo zip and the cvs file correct, the code should run.

After reading the zip file, CoinDrops file (which is inside the zip) directory should be written to train_generator and test_generator

https://github.com/melyneca/Coin-Drop