Weekly Report

Week 6

Nguyen Huu Huy Thinh

2025-03-10

Abstract

This report summarizes the work completed this week, which includes fixing the rotation axis of the robot arm's end effector and applying a custom Convolutional Neural Network (CNN) on the plastering data provided. The report covers the adjustments made to the robot arm's rotation axis, as well as the application of the custom CNN model to the plastering dataset.

1 Robot Arm Adjustments

This week, I changed the rotation axis of the robot arm's end effector to improve its movement and alignment. This adjustment helps the arm operate more smoothly and follow the intended motion more accurately.

- Video Link: https://drive.google.com/video1
- Joint Angle: Rotates around the X-axis (Ox), with an angle range of $0° \leq x \leq 180°$, having a 1:1 motor-to-joint connection for smooth movement.

2 Plastering Data Preparation

The 5-minute video was divided into 36,001 individual frames, corresponding to 36,001 data points in an Excel file. Each data point represents a time interval of 0.008333 seconds. Each data point represents a record taken at an interval of 0.008333 seconds. Each data point consists of two key components: the position of the trowel and the time at which it was recorded.

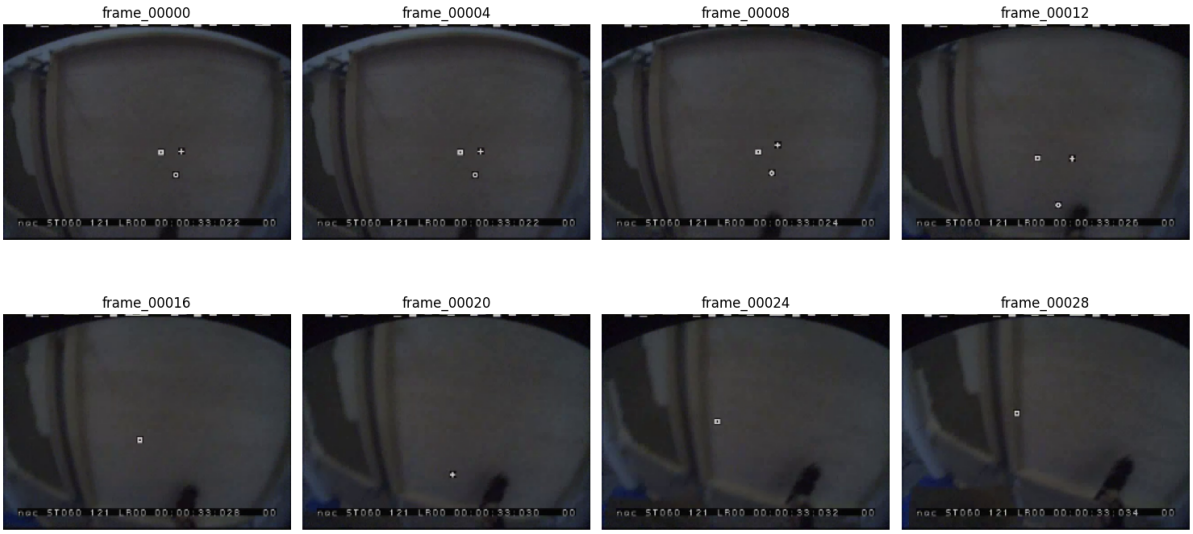

Figure 1: Video Frames

Time[s]	X[mm]	Y[mm]	Z[mm]	xx	xy	xz	yx	yy	yz	zx	zy	zz
0	1.540984	400.174255	-2.351813	0.999682	-0.024465	-0.006109	0.024309	0.999408	-0.024363	0.006702	0.024207	0.999685
0.008333	1.553865	400.180626	-2.4459	0.999688	-0.02434	-0.005666	0.024202	0.999437	-0.023215	0.006228	0.023071	0.999714
0.016666	1.562118	400.221721	-2.643259	0.999691	-0.024419	-0.004636	0.024317	0.999483	-0.021018	0.005147	0.020898	0.999768
0.024999	1.557966	400.350363	-2.88786	0.999683	-0.024968	-0.003358	0.0249	0.999513	-0.018805	0.003826	0.018716	0.999818
0.033332	1.58805	400.504954	-3.239581	0.999669	-0.02567	-0.001544	0.02564	0.999533	-0.016645	0.001971	0.0166	0.99986
0.041665	1.649195	400.7526	-3.628307	0.999638	-0.026913	0.000546	0.026918	0.999516	-0.01562	-0.000125	0.015629	0.999878
0.049998	1.777172	401.067745	-4.097636	0.999586	-0.028609	0.003064	0.028653	0.99947	-0.01542	-0.002621	0.015501	0.999876
0.058331	2.010375	401.449879	-4.696184	0.999509	-0.030717	0.006192	0.030813	0.999396	-0.016094	-0.005694	0.016277	0.999851
0.066664	2.397443	401.828676	-5.414048	0.999411	-0.032864	0.009905	0.033038	0.999292	-0.018015	-0.009306	0.018331	0.999789
0.074997	2.940621	402.185052	-6.207849	0.999301	-0.034754	0.013758	0.035039	0.999165	-0.021036	-0.013016	0.021504	0.999684
0.08333	3.711247	402.440437	-7.202341	0.999188	-0.035891	0.018281	0.036329	0.999047	-0.024193	-0.017395	0.024838	0.99954
0.091663	4.683607	402.587919	-8.317937	0.999081	-0.03618	0.022983	0.036811	0.99894	-0.027635	-0.021959	0.028456	0.999354
0.099996	5.873095	402.574546	-9.590668	0.998997	-0.035226	0.027631	0.036058	0.998893	-0.030211	-0.026537	0.031177	0.999162
0.108329	7.325656	402.364573	-11.057713	0.998953	-0.032671	0.032024	0.033693	0.998923	-0.031899	-0.030947	0.032945	0.998978
0.116662	8.967431	402.045879	-12.562662	0.998941	-0.029198	0.035554	0.030401	0.998966	-0.033791	-0.03453	0.034836	0.998796
0.124995	10.770986	401.696671	-14.138351	0.998956	-0.025272	0.038063	0.026624	0.999017	-0.035451	-0.03713	0.036428	0.998646
0.133328	12.819039	401.283721	-15.807171	0.999001	-0.020898	0.039499	0.022398	0.99903	-0.037921	-0.038668	0.038767	0.9985
0.141661	15.005714	400.861551	-17.468559	0.999081	-0.016566	0.039539	0.018197	0.998983	-0.041255	-0.038816	0.041937	0.998366
0.149994	17.318387	400.514626	-19.179861	0.999179	-0.012893	0.038417	0.014622	0.998876	-0.045077	-0.037793	0.045602	0.998245
0.158327	19.818802	400.21502	-21.020246	0.999286	-0.009883	0.036474	0.01169	0.998698	-0.049658	-0.035936	0.050049	0.9981
0.16666	22.381459	400.015507	-22.893607	0.999396	-0.007991	0.033805	0.009852	0.998425	-0.055237	-0.033311	0.055537	0.997901
0.174993	25.04293	399.82726	-24.943046	0.999501	-0.00662	0.030871	0.008478	0.998135	0.060445	0.030414	0.060677	0.997694
0.183326	27.84392	399.610481	-27.188941	0.999601	-0.005547	0.027706	0.007356	0.997816	-0.065637	-0.027282	0.065814	0.997459
0.191659	30.657355	399.377971	-29.587691	0.999683	-0.004831	0.024729	0.00656	0.997501	-0.070344	-0.024327	0.070484	0.997216
0.199992	33.519776	399.087661	-32.186551	0.999747	-0.004146	0.022106	0.005778	0.997221	-0.074273	-0.021736	0.074382	0.996993
0.208325	36.473493	398.708039	-35.048636	0.999793	-0.003234	0.0201	0.004789	0.996958	-0.07779	-0.019787	0.07787	0.996767
0.216658	39.395081	398.232898	-38.098514	0.999816	-0.001932	0.019092	0.003466	0.996734	-0.080682	-0.018874	0.080733	0.996557

Figure 2: Data Points

The position of the trowel is described by 12 values. The first three values (X, Y, Z) represent the world coordinates of the trowel, with the origin set at the left bottom corner of the wall. The next nine values represent the local coordinates of the trowel, captured as a 3x3 rotation matrix. These values are as follows: (xx, xy, xz), (yx, yy, yz), (zx, zy, zz).

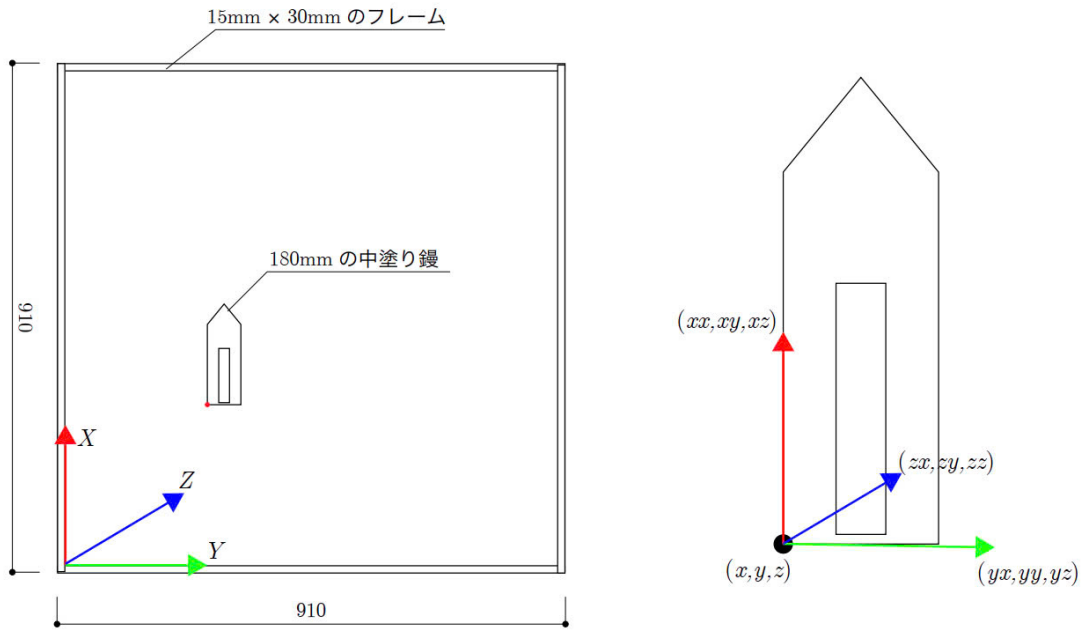

Figure 3: Trowel's position

In the preprocessing step, the images corresponding to each frame were resized to a resolution of 160x120 pixels to facilitate model training. After this, the dataset was split into training and testing sets, with 80% of the data used for training and 20% reserved for testing. This random split ensures a balanced representation of the data for evaluation.

3 Model Architecture

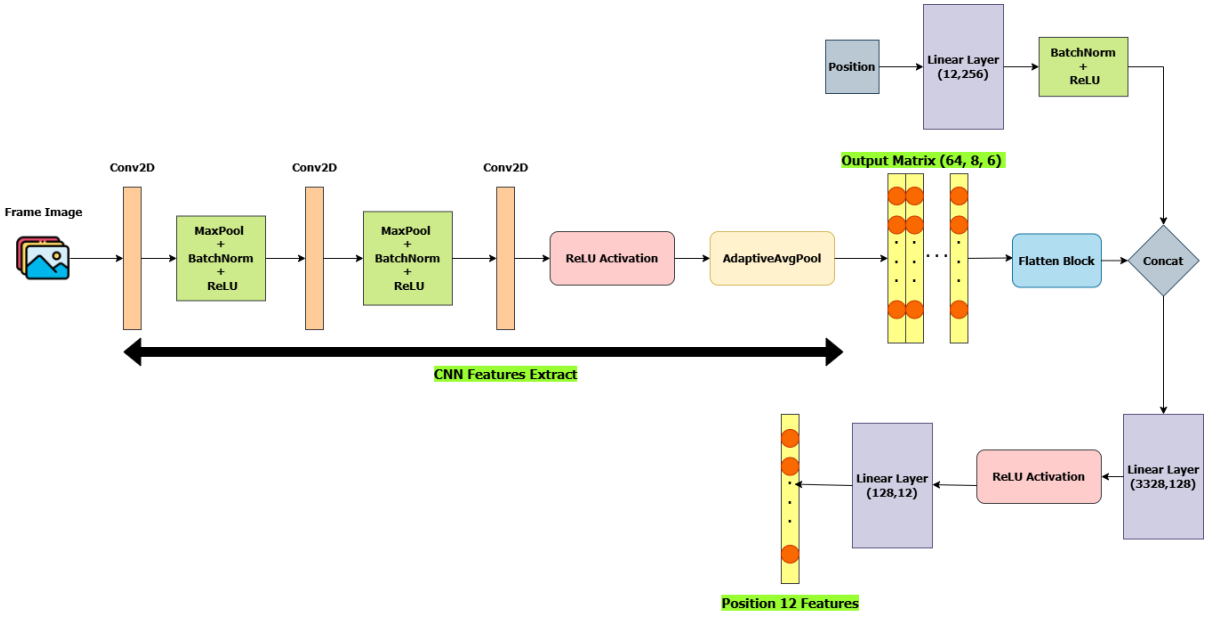

Figure 4: Model Architecture

3.1 Input Components

- Image Input: RGB images of shape (batch_size, 3, 160, 120).
- Position Input: A feature vector containing 12 values per sample, with shape (batch_size, 12).

3.2 Features Extraction

- Convolutional Neural Network (CNN):
  - Three convolutional layers extract spatial features from the segmented image.
  - Each layer uses Batch Normalization and ReLU activation.
  - The final output is reduced to a fixed-size representation via Adaptive Average Pooling, resulting in a feature map of shape (64, 8, 6).
- Position Feature Encoder:
  - A fully connected (FC) layer processes the position features vector, expanding it to a 256-dimensional representation.
- Feature Projection:
  - The extracted CNN features ($64 \times 8 \times 6 = 3072$) and position features (256) are concatenated, forming a 3328-dimensional input
  - This is passed through another fully connected layer with ReLU activation, reducing it to a 128-dimensional feature vector.

3.3 Output Layer

- The resulting 128-dimensional feature vector is passed through a fully connected layer to predict the next position vector.
- The output has the shape (batch_size, position_dim), where position_dim is the dimension of the predicted position

4 Training and Evaluation

4.1 Training Phase

For training, the model was set up with the following parameters:

- Epochs: 50
- Batch size: 32
- Learning Rate: 1e-3
- Weight Decay: 1e-4
- Optimizer: AdamW
- Loss Function: L1 Loss

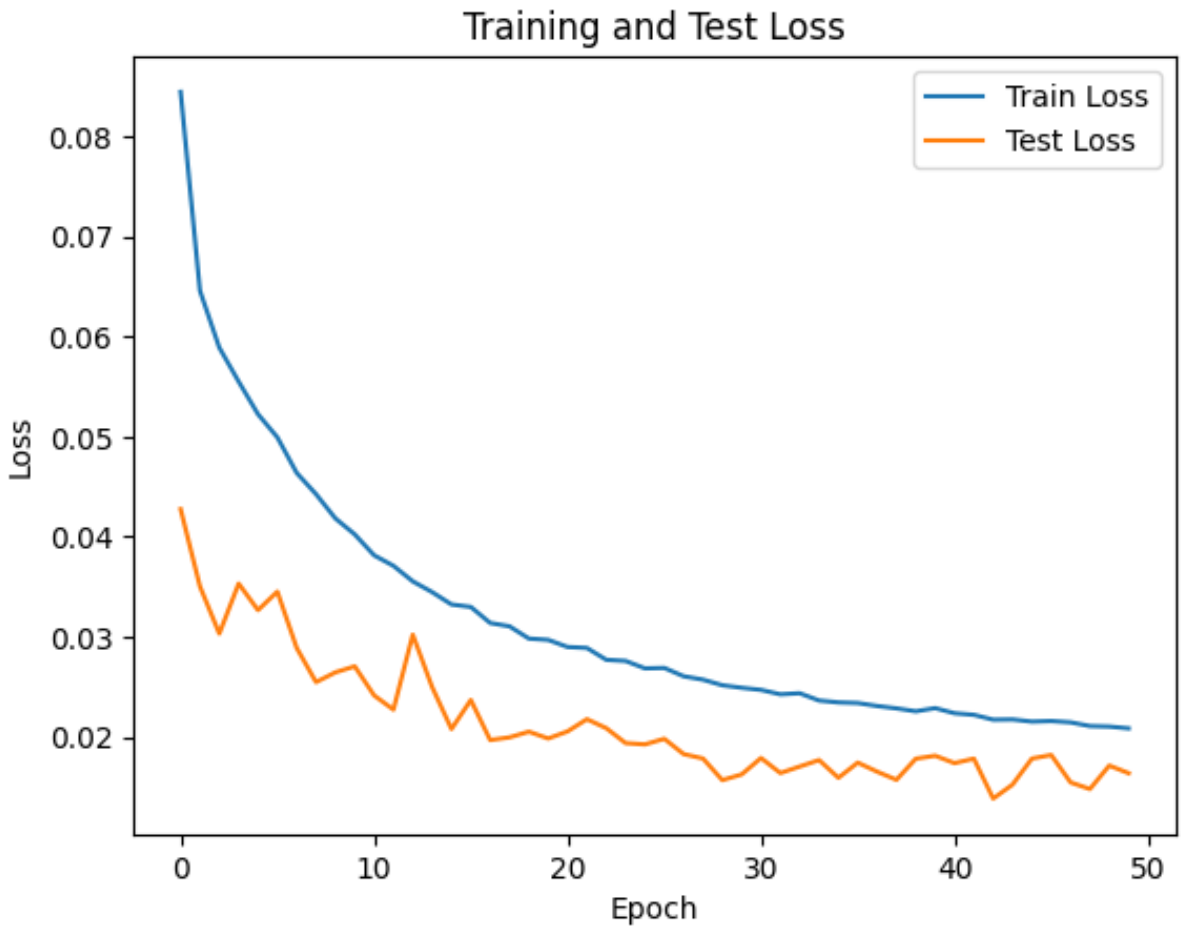

Figure 5: Loss Diagram

The diagram (Figure 5) shows the training and test loss of a model over 50 epochs. The blue line represents the training loss, which starts at around 0.08 and decreases steadily,

leveling off near 0.02 by the end. The orange line represents the test loss, starting slightly higher and fluctuating more, but also decreasing overall to around 0.02-0.03. This suggests the model is learning effectively, with both training and test losses converging, indicating good generalization without significant overfitting.

4.2 Evaluation Phase

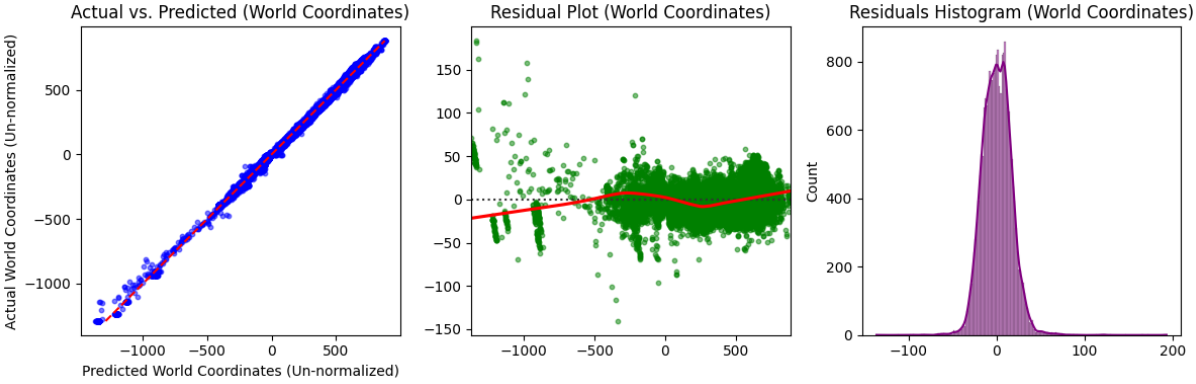

Figure 6: Evaluation Diagrams (World Coordinate)

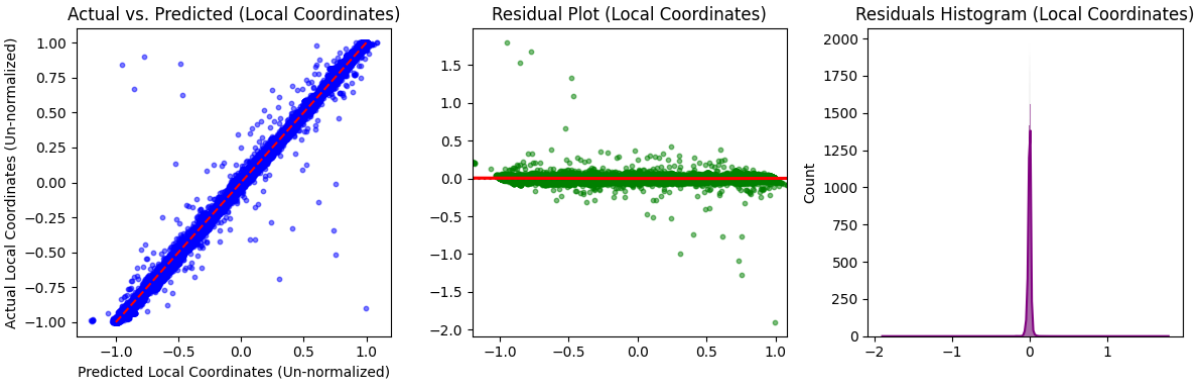

Figure 7: Evaluation Diagrams (Local Coordinate)

- R^2 Value: 0.9677 (world coordinate) and 0.9874 (local coordinate) The R^2 score, or coefficient of determination, measures how well the model's predictions match the actual values. An R^2 score close to 1 indicates that the model captures almost all the variability in the data, leaving very little unexplained.
- Actual vs. Predicted Graph:
  - Each data point represents a pair of values: the actual (true) coordinate and the predicted coordinate for a sample.
  - X-Axis: The predicted coordinates. Y-Axis: The actual coordinates.

  - The blue data points closely follow the red diagonal line (where actual = predicted), indicating a strong linear relationship. This means the model's predictions are very close to the true local coordinates, suggesting high accuracy across the range.

- Residual Plot
  - Each data point shows the residual (error), which is the difference between the actual and predicted local coordinates for a sample. This measures how far off the predictions are from the true values.
  - X-Axis: The predicted coordinates. Y-Axis: The residuals.
  - In both residual graphs, the points are scattered around the red dashed line at zero with no clear trend. This indicates that the errors are random and not systematically related to the predicted values, suggesting that the model has no significant bias.

- Residuals Histogram
  - Each bar in the histogram represents the frequency (count) of residuals falling within a specific range. The residuals are the differences between actual and predicted coordinates, aggregated to show the distribution of errors across all samples.
  - X-Axis: The residual values. Y-Axis: The count of residuals
  - The histogram forms a bell-shaped curve centered near zero. This symmetric distribution suggests that the errors are normally distributed, which is a good sign of a well-fitted model with no systematic over- or under-prediction.

4.3 Inference Phase

This diagram shows how a robot arm uses OAK-1 camera to see its workspace. The camera provides frame for CNN model. The picture and position information go to a custom CNN Model, which decides how the robot arm should move.

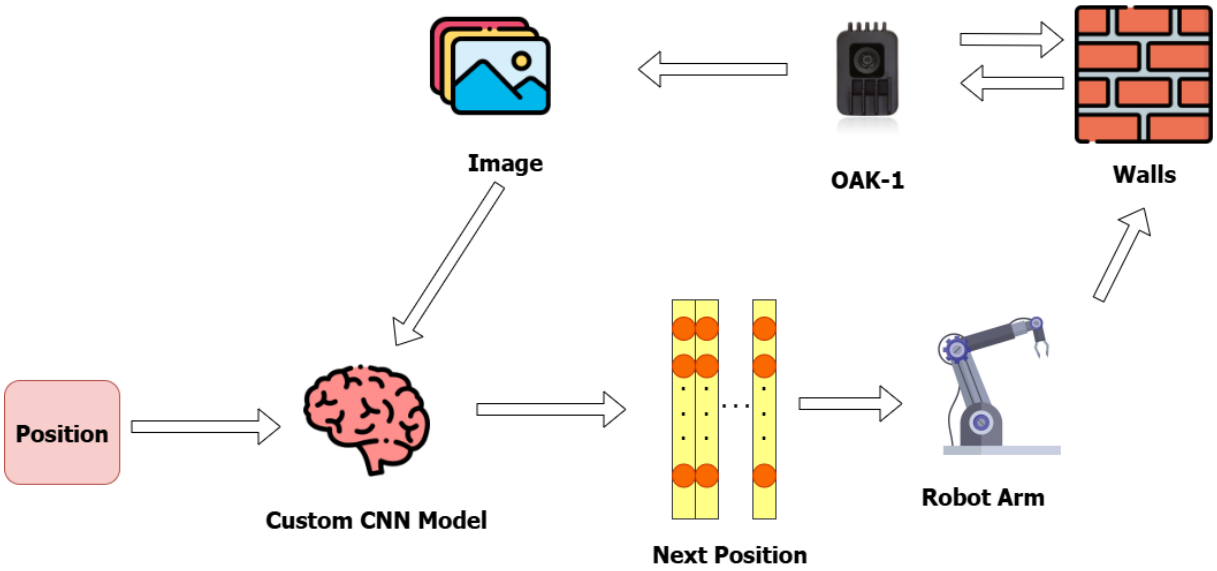

Figure 8: Inference Diagram

Video Link:

- Demo Simple Wall: https://drive.google.com/video2

5 Conclusion

This report summarizes the work completed this week, including fixing the rotation axis of the robot arm end effector and applying a custom Convolutional Neural Network (CNN) to the provided plastering data. The dataset, consisting of 36,001 frames extracted from a five-minute plastering video, was processed and structured for training. From the high R^2 scores and evaluation graphs, the model demonstrates strong performance in predicting the next movement based on the current frame and position.

References