The CNN network used in this project consists of three convolutional layers and two fully connected layers. A sigmoid activation function is used for the last layer and the “BCELoss” function. The following table outlines the characteristics of each layer.

	Input size	Output size	Kernel size
Conv1	1	3	7
Conv2	3	6	5
Conv3	6	12	3
Fc1	12*100*100	256	
Fc2	256	1	

In this stereo image dataset, we can consider one image to be the translated image of another image.

1. Symmetry group

To determine the symmetry group, rotation and translation transformations are used. Training the model with left side images and testing it with right images allows us to determine the translation state. In order to check the rotation state, a 90 degree rotation is used. This network is non-invariant to rotation but invariant to translation as shown in the figure below.

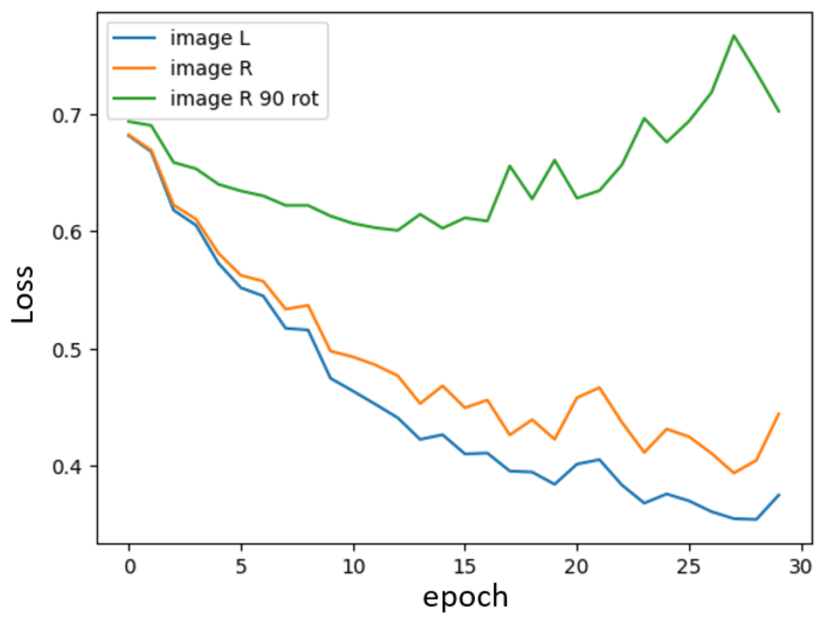

Figure 1. Analysis of test loss results. The basic model is trained with left side images and tested with left side images as well, the second curve is tested with right side images, and the third curve is the result of testing the model with right side image that have been rotate 90 degree.

Based on the results, it can be determined that the network must be invariant over the C4 group.

2. Group-smoothing

The averaging operator is used here to smooth the network. So, we first apply the group transformation to the input and then calculate the layer's output from their average.

By training the network with this technique, it can be seen that the convolution layers are equivalent to the C4 group. However, the whole network is still not invariant from the C4 group and shows different results with rotated input.

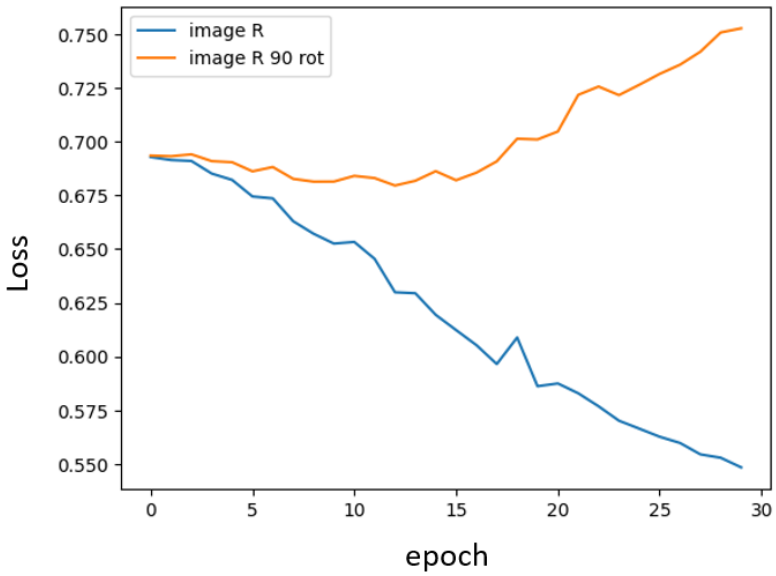

Figure 2. Analysis of test loss results for smoothed model. The smoothed model is trained with left side images and test with right side images and rotated images.

3. Group-equivariant neural network and implementation

In order to add group-equivarant property to convolutional layers, instead of applying all the group transformations on the input, we can apply all the group transformations on the kernels. In this way, for the C4 group, four parallel convolution operators are used. It is important to note that each kernel belongs to the C4 group. Finally, we consider the maximum value (using max pooling) of the output features as input to the next layer.

Using the smoothing technique, the layers are equivariant, as shown in the second part. Our first convolution layer uses the group equivariant approach, and our second layer uses the smoothing approach. In the third layer, we use the usual 2D convolution. In order to make the network invariant, we applied an invariant pooling function before the flatten layer, which produces the sum of all group transformations on the extracted features. Therefore, due to the transformation of the C4 group, the features of the input to the fc1 layer will not change.

In the following figures you can see the extracted features as input of the fc1 layer without the invariant pooling function in the basic network. In Figure 3, the input of the network is the right side image, and in Figure 4, the input is the right side image rotated 90 degrees.

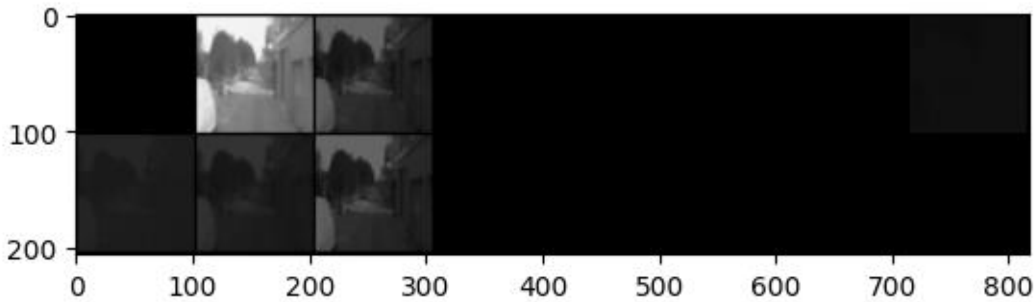

Figure 3. Extracted features by using the right side image as input to the basic network.

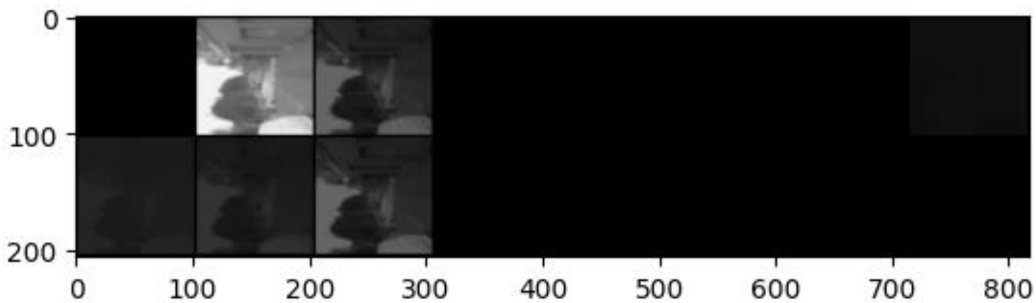

Figure 4. Extracted features by using the right side image rotated 90 degrees as input to the basic network.

The following figures also illustrates how the invariant pooling function impacts the extracted features, which form the input for the fc1 layer. In Figure 5, the input of the network is the right side image, and in Figure 6, the input is the right side image rotated 90 degrees.

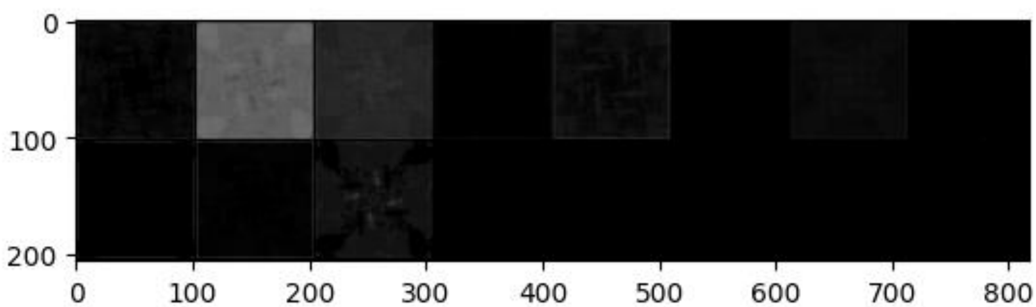

Figure 5. Extracted features by using the right side image as input to the group-invariant network.

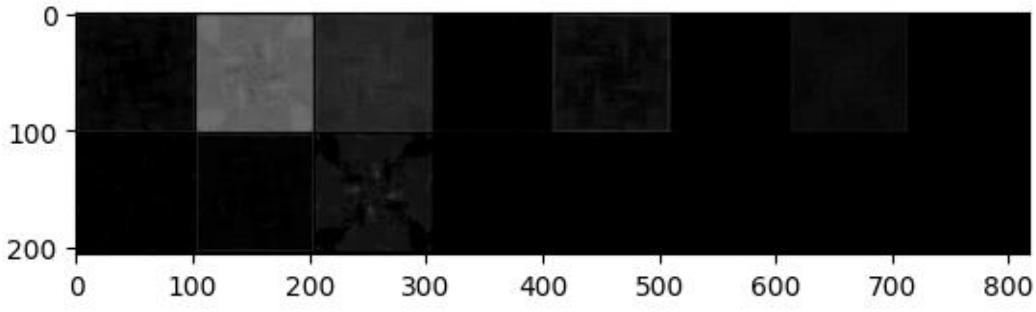

Figure 6. Extracted features by using the right side image rotated 90 degrees as input to the group-invariant network.

As shown in these figures, by using the invariant pooling function, the input of the fc1 layer remains unchanged under the C4 group transformations, and as expected, the network is invariant to the C4 group.

As shown in Figure 7, the test loss results for group-invariant networks are shown for both the right side images and the rotated version. In this network, the left side images are used to train the network. The results of testing on right side images and rotated images confirm that this network is invariant to translations as well as rotations.

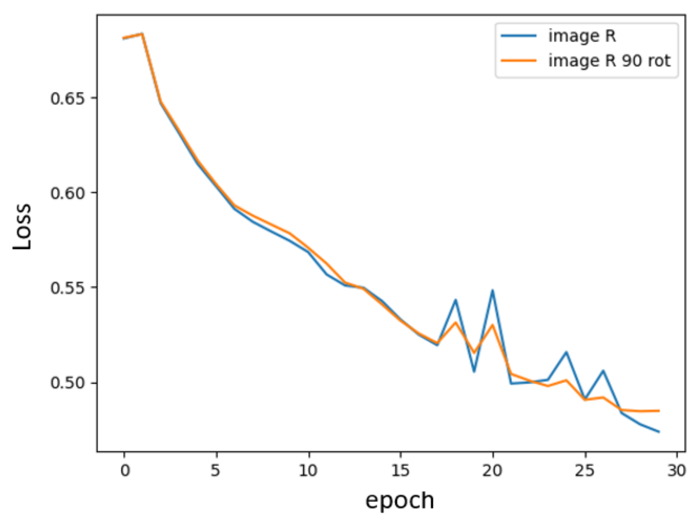

Figure 7. Analysis of test loss results for invariant model. The invariant model is trained with left side images and test with right side images and rotated images.

In the following table, we present the results of three networks (basic, smoothed, and invariant networks). We trained these networks using 30 epochs.

Table1 . Comparison of loss value for three models.

Model	**Rotated input (C4 Group)**	**Translated input (Right side image)**
Basic model	0.7021	0.4443
Smoothed model	0.7527	0.5484
Group Invariant	0.4849	0.4739