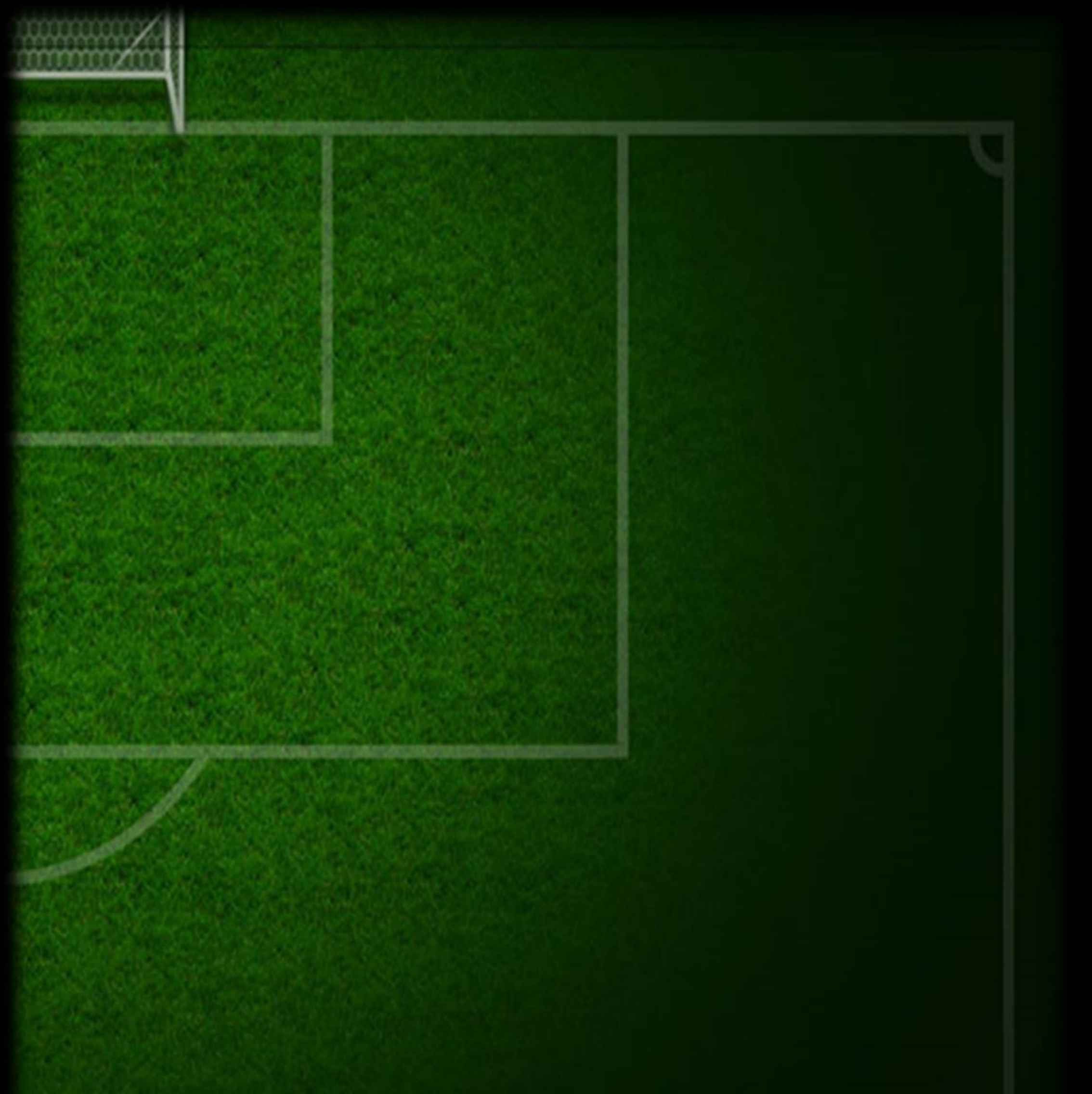
Real to Synthetic Image Translation for Pose and Image understanding in Robocup



Serena Bono Msc in Robotics Systems and Control sebono@student.ethz.ch
Dr. Ajad Chhatkuli ETF D113.2 ajad.chhatkuli@vision.ee.ethz.ch
Prof. Luc Van Gool ETF C117 vangool@vision.ee.ethz.ch

Introduction

Image understanding in Robocup:

Understanding the image of the Robocup field means detecting the main features of the field, namely robots, filed lines, field.

We want to design the method for unpaired data

Real data: robot camera view of robot matches
Synthetic data: generated data

Cycle-GAN

«Cycle-GAN learns to translate an image from a source domain X to a target domain Y in the absence of paired examples. Its goal is to learn a mapping G : X → Y such that the distribution of images from G(X) is indistinguishable from the distribution Y using an adversarial loss»

$$\mathrm{L_{GAN}}\left(\mathrm{G}, \mathrm{D_y}, \mathrm{X}, \mathrm{Y}\right) = \\ \mathbb{E}_{\mathrm{y} \sim \mathrm{P_{data}(y)}}\left[\log \mathrm{D_y}(\mathrm{y})\right] + \\ \mathbb{E}_{\mathrm{x} \sim \mathrm{P_{data}(x)}}\left[\log(1 - \mathrm{D_y}(\mathrm{G}(\mathrm{x})))\right]$$

$$\mathrm{L_{cyc}}(\mathrm{G}, \mathrm{F}) \\ = \mathbb{E}_{\mathrm{y} \sim \mathrm{P_{data}(y)}}\left[||\mathrm{G}\left(\mathrm{F}(\mathrm{y})\right) - \mathrm{y}||_1\right] \\ + \mathbb{E}_{\mathrm{x} \sim \mathrm{P_{data}(x)}}\left[||\mathrm{F}\left(\mathrm{G}(\mathrm{x})\right) - \mathrm{x}||_1\right]$$

X: Real Data
Y: Synthetic Data

Cycle-GAN in Robocup

Take Unlabeled Input Data → Learn Density Distributions → Translate To Synthetic

Cycle-GAN in Robocup

Take Unlabeled Input Data → Learn Density Distributions → Translate To Synthetic

Cycle-GAN in Robocup

Take Unlabeled Input Data → Learn Density Distributions → Translate To Synthetic

- Introduction
- Strategies
- Datasets and Synthetic data generation
- Domain Translation
- Pose Evaluation
- Real to synthetic translation by Homography
- Conclusions

Strategies

In the following sections methods are investigated to perform image translation for image and pose understanding.

First Strategy

Image Translation

Posenet (on translated data)

Second Strategy

Posenet (on real data)

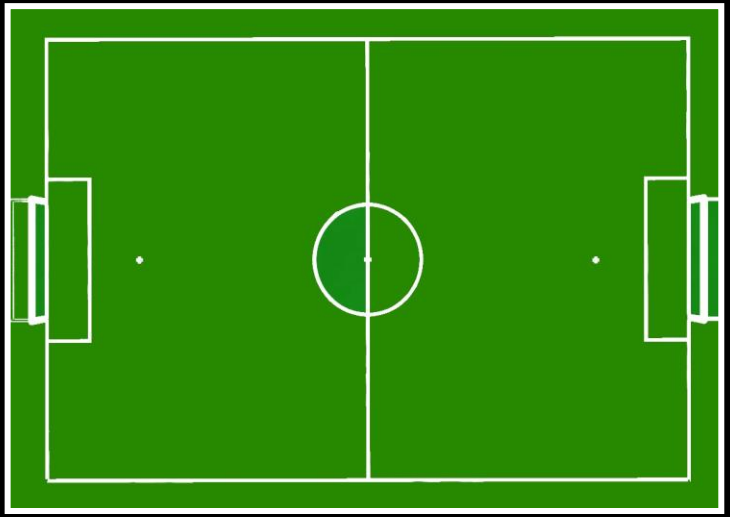
Self-supervised Learning

Datasets and Synthetic data generation

https://github.com/serenabono/Semester-Project/generate synthetic images/

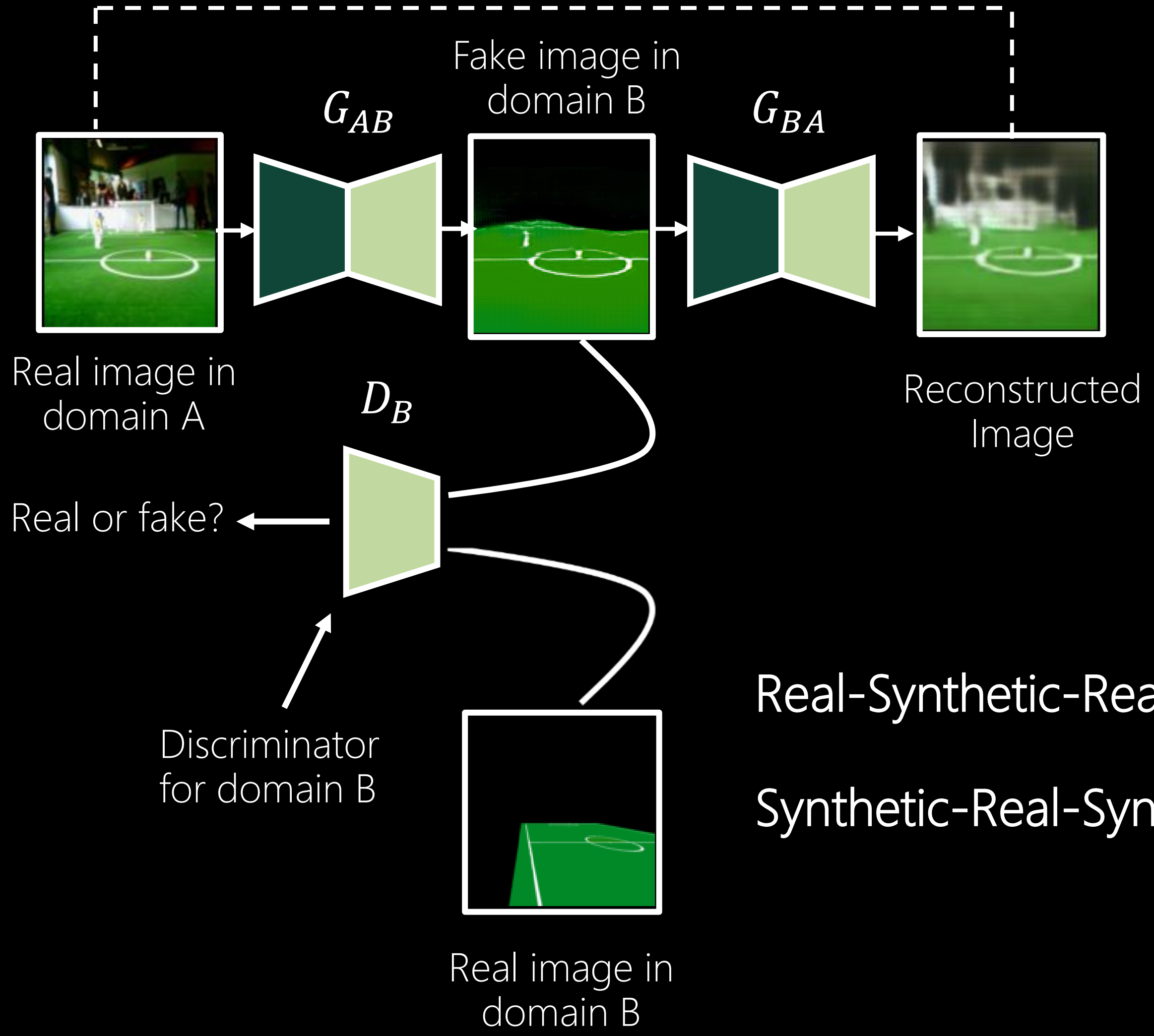
The soccer fields can be accurately defined by a 2D image and then projected using an image warp.

Generate around 3300 images to be divided into :
~3000 training
~300 testing



Domain translation

https://github.com/serenabono/Semester-Project/cycleGAN-PyTorch



Real-Synthetic-Real

Synthetic-Real-Synthetic

Domain translation

More information about the GAN:

Generator:
- three convolutions,
- several residual blocks,
- two strided convolutions,
- one convolution that maps features to RGB

Discriminator:
- 70 × 70 PatchGANs

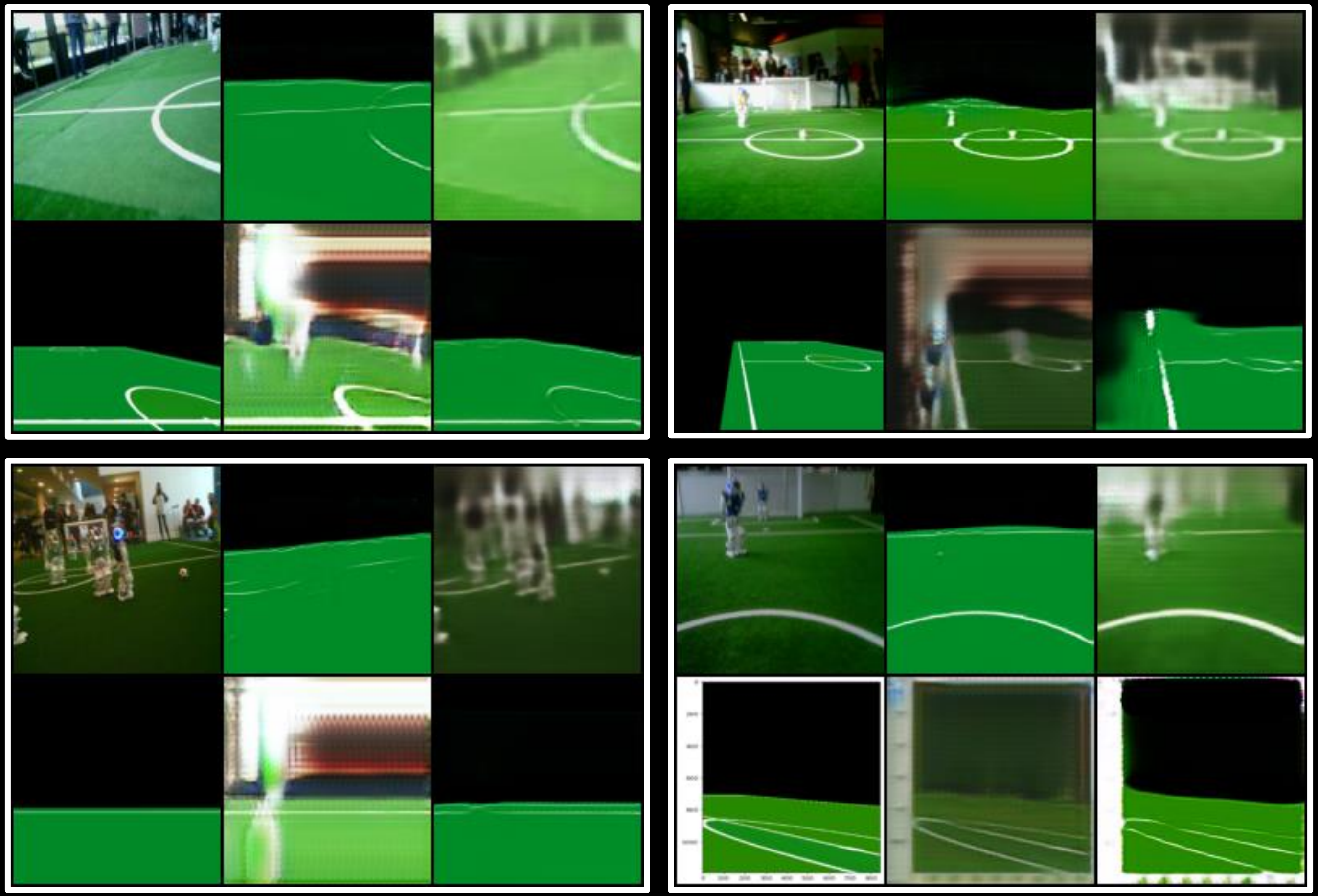
Losses:

Generator:
- Identity Loss: $L1_{loss} * \lambda * idt_{coeff}$
- Adversarial Loss: MSE_{loss}
- Cycle Consistency Loss: $L1_{loss} * \lambda$

Discriminator:
- MSE_{loss}

λ=10
idt_{coeff}=0.5

Domain translation



Fighting Mode Collapse

Mode Collapse: the generator start producing only a single type of output or a small set of outputs.

Plausible Cause:
- the variability of the synthetic image was very low. Therefore, the loss gradient most of the time was close to zero.

Solution:
- Injecting some variability by gradually increasing the intensity bottom to top.

Fighting Mode Collapse

https://github.com/serenabono/Semester-Project/generate_synthetic_images/trainBmain_w_robots.py

Mode Collapse: the generator start producing only a single type of output or a small set of outputs.

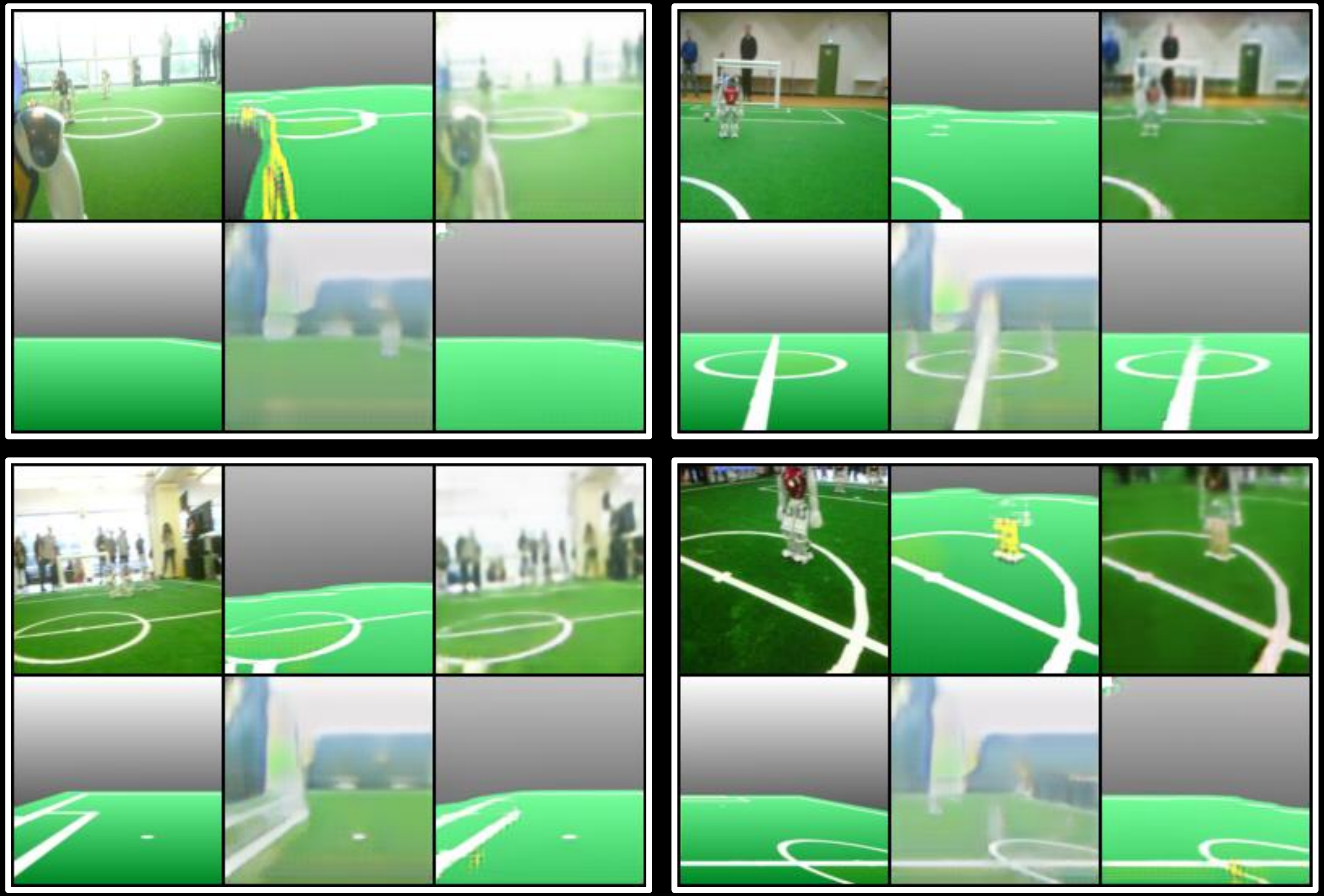
Plausible Cause:

- Synthetic images lack a representation for robots, this might be a problem when finding the mapping between real and synthetic images

Solution:

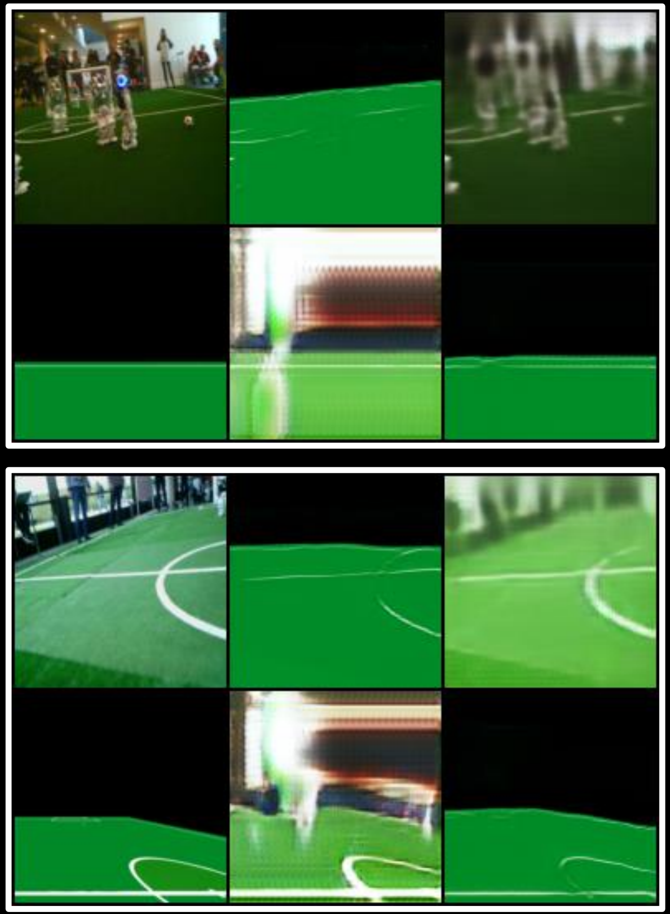
- Generating robot representation

Domain translation

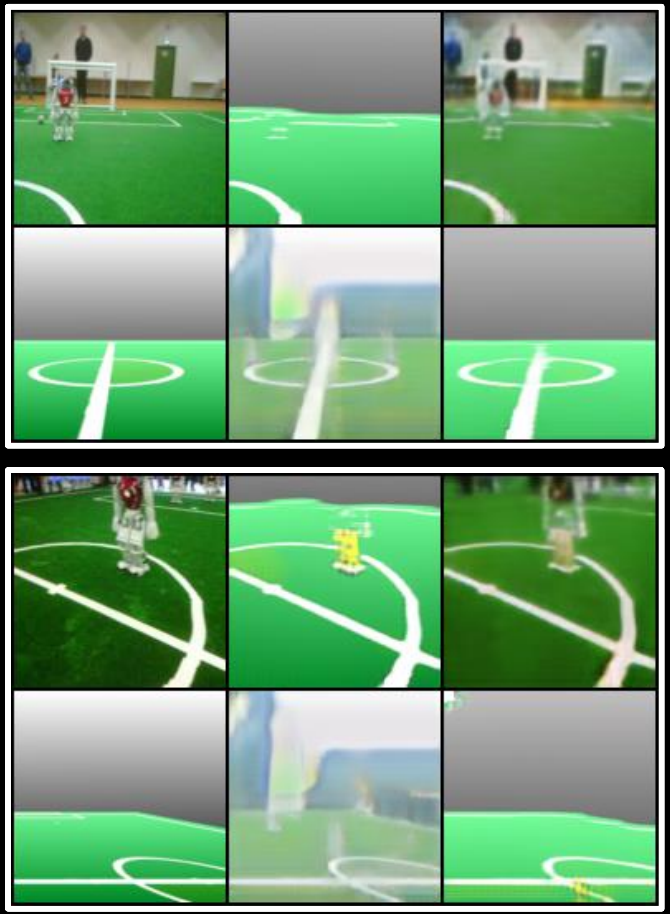


- Introduction
- Strategies
- Datasets and Synthetic data generation
- Domain Translation
- Pose Evaluation
- Real to synthetic translation by Homography
- Conclusions

Domain translation



Before



After

- Introduction
- Strategies
- Datasets and Synthetic data generation
- Domain Translation
- Pose Evaluation
- Real to synthetic translation by Homography
- Conclusions

A Metric for Evaluation

The Frechet Inception Distance (FID) is well-known metric for Cycle GAN evaluation: it calculates the distance between feature vectors of real and generated images. The score represents the dissimilarity of the two images, therefore lower the score higher the accuracy.

$$FID = |\eta - \mu_w|^2 + \mathrm{tr}(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{\frac{1}{2}})$$

The FID requires the features to be computed meaningfully for both the distributions. While for natural images, ImageNet pre-training provides meaningful feature vectors, this is not as reliable for the images of the Robocup field.

Pose Evaluation

Pose Evaluation is fundamental for image understanding.

Model for pose regression:

$$(\partial x, \partial y, \partial z, q1, q2, q3, q4)$$

- The first 3 values are <u>the displacements</u> in the respective directions
- The last 4 values are measure of rotation: "<u>quaternions</u>" 4D values easily mappable to legitimate rotations by normalization to unit length.

Those values can be deduced from the rotation matrix and the displacement of the homography.

PoseNet

https://github.com/serenabono/Semester-Project/visloc-apr

"Posenet is a deep convolutional neural network camera pose regressor".

The Architecture:

- Modified ResNet 34 core module
- Replace all three softmax classifiers with affine regressors.
- Each final fully connected layer was modified to output a pose vector of 7-dimensions
- Insert another fully connected layer before the final regressor of feature size 2048 to form a localization feature
- Normalize the quaternions to unit-length.

PoseNet

https://github.com/serenabono/Semester-Project/visloc-apr

"Posenet is a deep convolutional neural network camera pose regressor".

The Loss Function:

The convnet was trained on Euclidean loss using stochastic gradient descent with the following objective loss function:

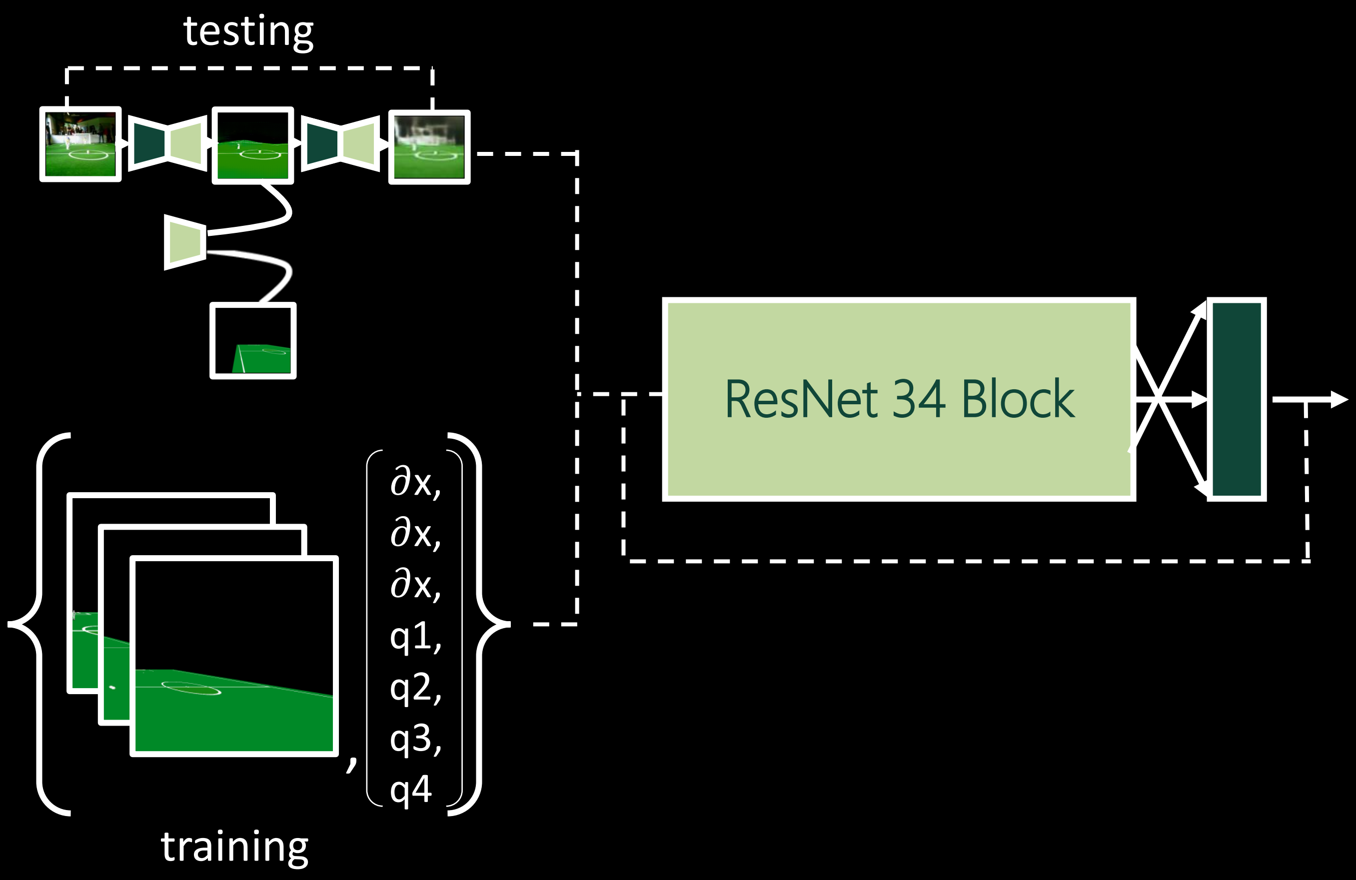
$$loss(I) = ||\hat{x} - x||_2 + \beta||\hat{q} - \frac{q}{||q||}||_2$$

where β is a scale factor chosen to keep the expected value of position and orientation errors to be approximately equal.

PoseNet

https://github.com/serenabono/Semester-Project/visloc-apr

"Posenet is a deep convolutional neural network camera pose regressor".

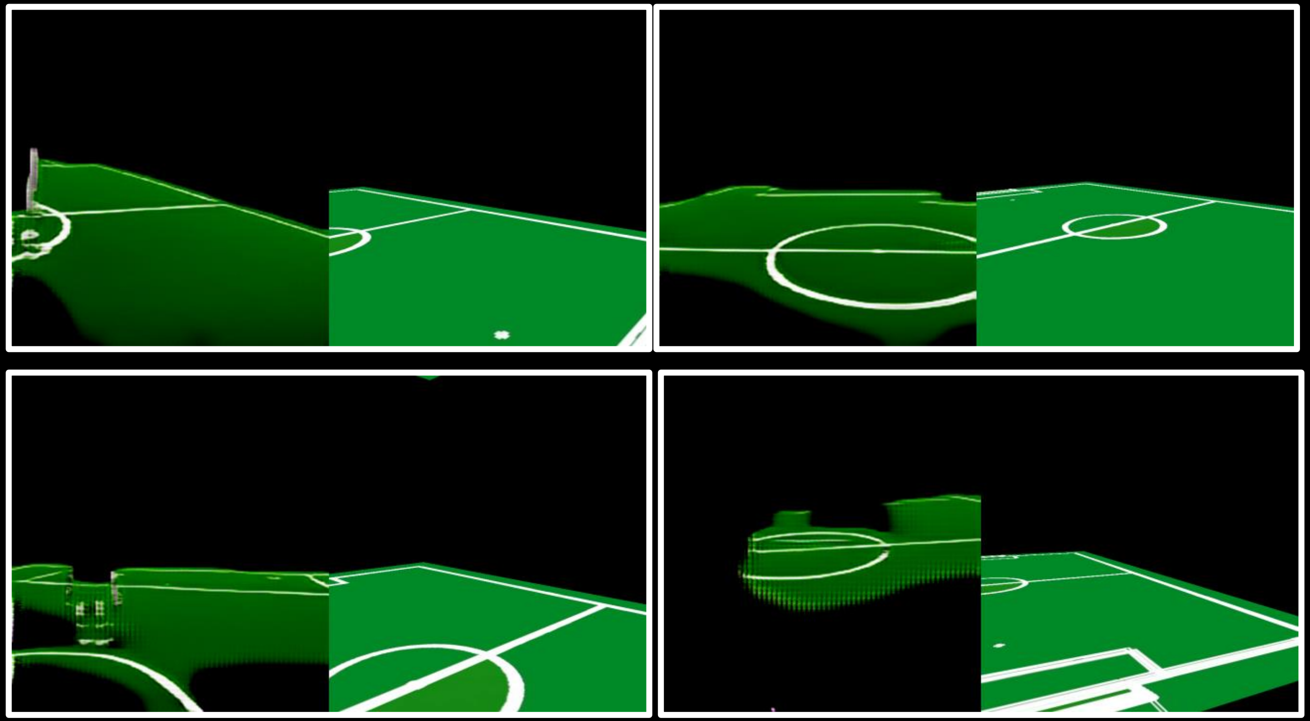


PoseNet

https://github.com/serenabono/Semester-Project/visloc-apr

"Posenet is a deep convolutional neural network camera pose regressor".

Results:



Strategies

In the following sections methods are investigated to perform image translation for image and pose understanding.

First Strategy

Image Translation

Posenet (on translated data)

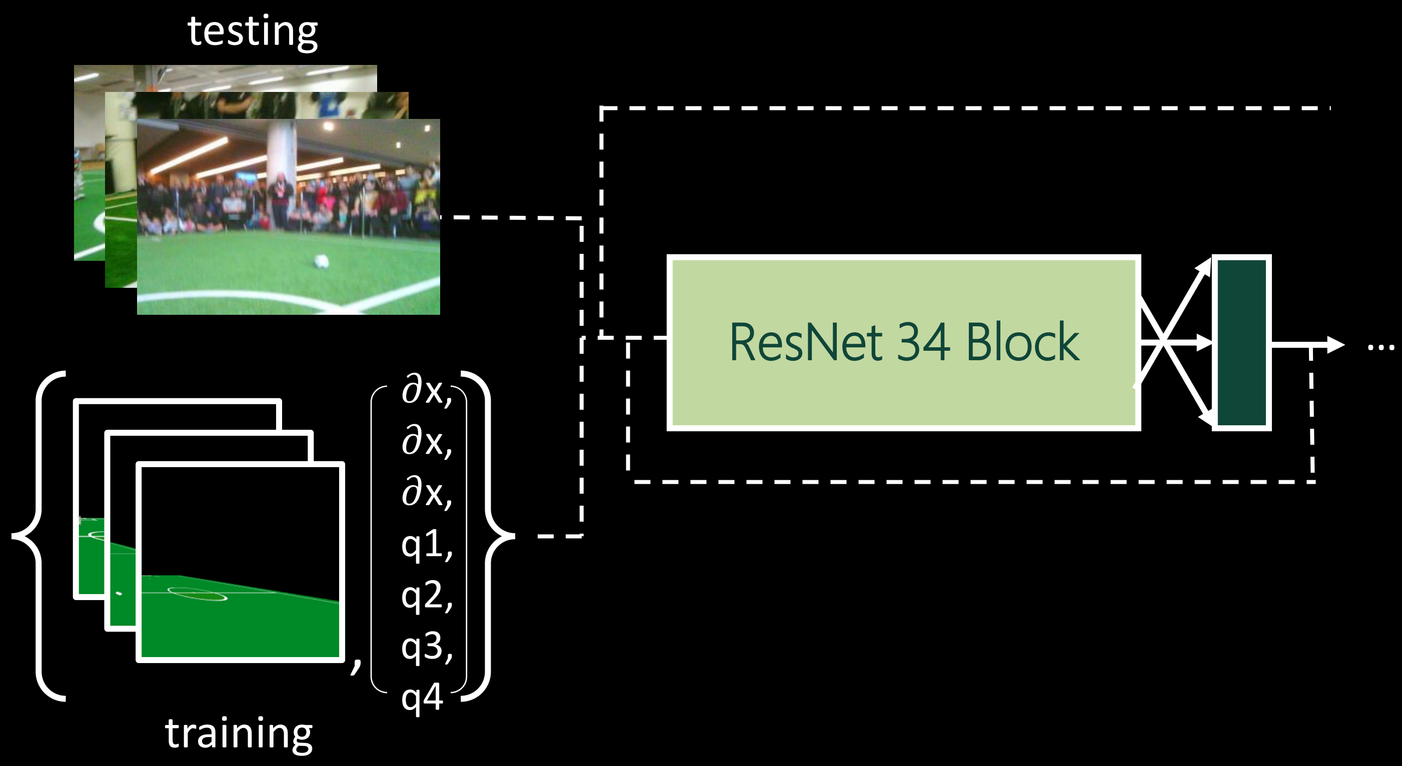
Second Strategy

Posenet (on real data)

Self-supervised Learning

Real-synthetic translation by Homography

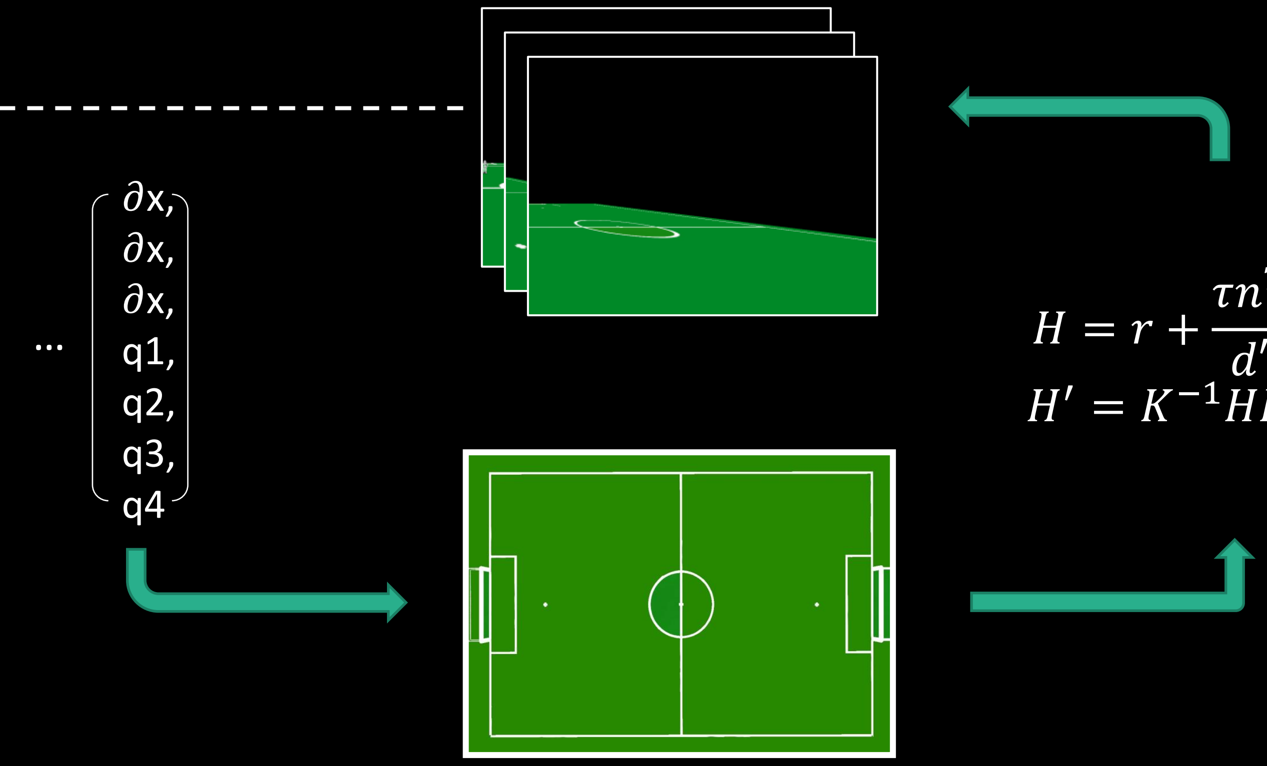
https://github.com/serenabono/Semester-Project/new-architecture



- Introduction
- Strategies
- Datasets and Synthetic data generation
- Domain Translation
- Pose Evaluation
- Real to synthetic translation by Homography
- Conclusions

Real-synthetic translation by Homography

https://github.com/serenabono/Semester-Project/new-architecture



$$H = r + \frac{\tau n^T}{d'}$$

$$H' = K^{-1} H K$$

Real-synthetic translation by Homography

https://github.com/serenabono/Semester-Project/new-architecture

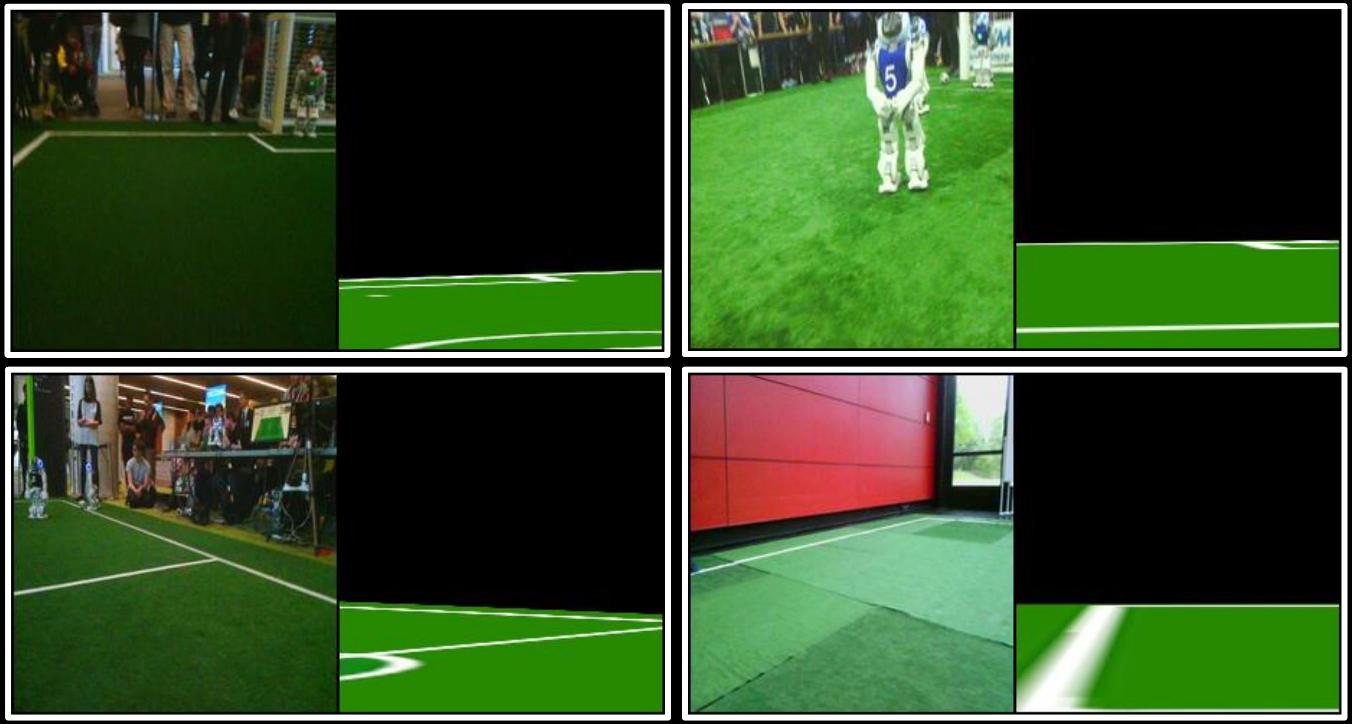
Loss Functions:

- L1 Loss
- L1 Loss + 2D Gaussian Blur + Sobel Filter: The gaussian blur smoothed out the real images and made them more uniform, while the Sobel filter underlined the contour of the figures and therefore the lines of the field.

Real-synthetic translation by Homography

https://github.com/serenabono/Semester-Project/new-architecture

Results L1:



Real-synthetic translation by Homography

https://github.com/serenabono/Semester-Project/new-architecture

Results L1 + filters:

Conclusions

The suggested strategies, if correctly implemented, would be able to extract the exact position of the robot in the field.

Nevertheless, the results of the translation only could be used for lower-level image understanding:

- Lines could be extracted from translated images directly by thresholding
- Data could be annotated to train smaller networks to be used inside the robot.