Deep Homography Estimation Report

Centro de Investigación en Matemáticas A.C

Esteban Reyes Saldaña

CIMAT

June 1, 2023

Summary

- It was introduced a deep convolutional neural network for estimating homography in a relative way between a pair of images.
- The problem was formulated in two ways: a regression and a classification problem.
- It outperforms the SOTA in 2016.

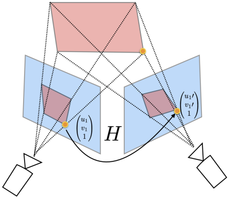
Classic Homography Estimation

- To perform the estimation of a homography, the basic material are correspondences among the two images

$$p_k, p_k'$$

that sould correspond to the same 3D point.

- Tradictional homography estimation is composed of two stages: corner estimation and robust homography estimation using RANSAC algorithm.


$\begin{pmatrix}u_1\\v_1\\1\end{pmatrix}$
H
$\begin{pmatrix}u_1'\\v_1'\\1\end{pmatrix}$

$$\begin{pmatrix}u'\\v'\\1\end{pmatrix} \sim \begin{pmatrix}H_{11} & H_{12} & H_{13}\\H_{21} & H_{22} & H_{23}\\H_{31} & H_{32} & H_{33}\end{pmatrix}\begin{pmatrix}u\\v\\1\end{pmatrix} \tag{1}$$

The 4-Point Homography Parametrization

Suppose that

$$U_i = (x_i, y_i)^T \tag{2}$$

for $k = 1, 2, 3, 4$ are 4 fixed point in the image i and

$$U_j' = (x_i', y_i')^T \tag{3}$$

for each pair of corresponding points, we have

$$\begin{aligned} h_1 x_i + h_2 y_i + h_3 - x_i x_i' h_7 - y_i x_i' h_8 - x_i h_9 &= 0 \\ h_4 x_i + h_5 y_i + h_6 - x_i y_i' h_7 - y_i y_i' h_8 - y_i' h_9 &= 0 \end{aligned}$$

and if we know 4 pair of points, considering $h_9 = 1$, the system can be solved using Direct Linear Transoformation.

Corner Location Parameterization

Why the 3×3 is not so useful for a Neural Network?

- Balancing the rotational and translational terms as part of an optimization problem is difficult.
- Rotation components tend to have a much smaller magnitude than the translation component.
- Impact in the $L2$ loss will be minimal.
- The 4-point parameterization does not suffer from these problems since there are all magnitudes refering to a position rate.

Corner Location Parameterization

Let

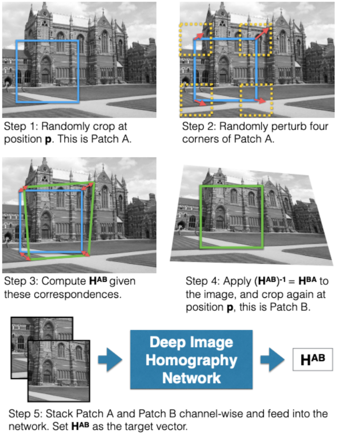
$$\Delta u_1 = u_1' - u_1 \tag{4}$$

be the u-offset for the first corner. The 4 point parametrization represents a homography as follows

$$H_{4points} = \begin{pmatrix} \Delta u_1 & \Delta v_1 \\ \Delta u_2 & \Delta v_2 \\ \Delta u_3 & \Delta v_3 \\ \Delta u_4 & \Delta v_4 \end{pmatrix} \tag{5}$$

once displacement is known, one can easily convert $H_{4points}$ to H using several methods, such as DLT or the function $getPerspectiveTransofrmation()$ in OpenCV.

xData Generation for Homography Estimation



Step 1: Randomly crop at position **p**. This is Patch A.

Step 2: Randomly perturb four corners of Patch A.

Step 3: Compute $\mathbf{H}^{\mathbf{AB}}$ given these correspondences.

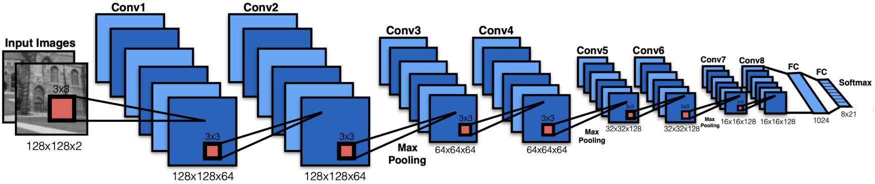
Step 4: Apply $(\mathbf{H}^{\mathbf{AB}})^{-1} = \mathbf{H}^{\mathbf{BA}}$ to the image, and crop again at position **p**, this is Patch B.

Step 5: Stack Patch A and Patch B channel-wise and feed into the network. Set $\mathbf{H}^{\mathbf{AB}}$ as the target vector.

The process is list as follow

1. Randomly crop a square patch from a large image I at position p.
2. Randomly perturbed corners by values within the range$[-\rho, \rho]$.
3. The four correspondences define a homography H_AB.
4. Compute $H^{BA} = (H_{AB})^{-1}$.
5. Apply H^{BA} to the large image.
6. A second patch I'_p is cropped from I' at position p.
7. The training example (I_p, I'_p) and the second element is $H_{4points}$.

Data Generation for Homography Estimation

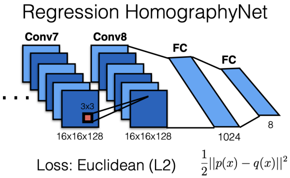


The networks use 3×3 convolutional blocks with BatchNorm and ReLUs activations. filtering

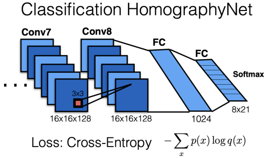
$$64, 64, 64, 64, 128, 128, 128, 128 \tag{6}$$

and finally it is connected to a two fully connected layers.

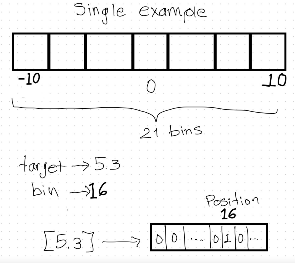
Networks Configuration



it produces directly 8 real values numbers and Euclidean L2 loss is applyed during training.



Use a quantization scheme, has a softmax at the last layer and we use the cross entropy loss function during training. 21 bins was chosen for each of the 8 outputs, which results in a final layer with 168 output neurons.



Networks Configuration

- Optimizer : **SGD** with a learning rate of 0.005 and momentum of 0.9. The learning rate was decrease by a factor of 10 after every 30,000 iterations.
- The network are trained for 90,000 total iterations using a batch size of 64.

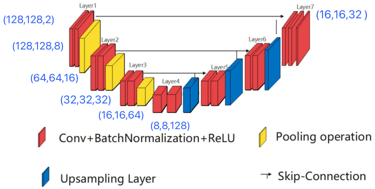
For creating the dataset, MS coco 2017 was used. The main configuration is

Item	Value
Img_size	$(320, 480)$
patch_size	$(128, 128)$
ρ	32
num_samples	500000
Iterations	90000

Item	Value
Img_size	$(640, 480)$
patch_size	$(256, 256)$
ρ	64
num_samples	5000

Unet for regression

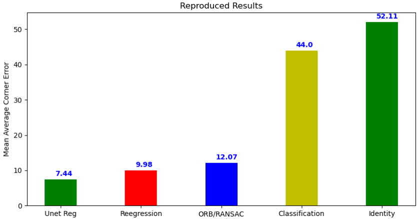
In the aim of reduce the number of parameters using a more complex network architecture, it is use a UNet for feature extracting.



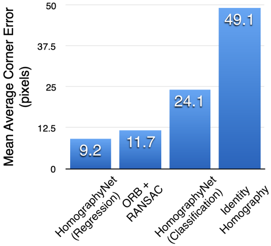
Model	n params	TPE
Reg	34,1 m	1.63 min
Class	34,4 m	1.62 min
Unet Reg	8,9 m	1.21 min

Metrics

According to the Mean Average Corner Error, we get

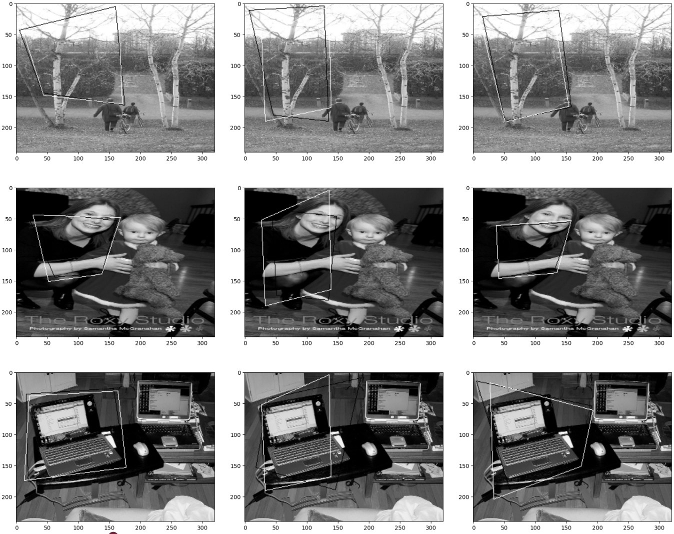


that is similar to the results of the original paper



Visual Results

Regression / Classification Model/ Unet Reegression



Conclusions

- it was introduced a new method for estimating homography between two gray scales images.
- it was introduced an algorithm for creating a synthetic dataset which is a tuple of two image gray-scale images and the homography given 4 corners.

Many suggesting can improve the model

- Introduce a metric that takes into account spatial features.
- Convolutions only sees local context, maybe some attention mechanism could work.
- The dataset is syntetic, it must be compare with real life examples.
- Investigate another metrics.