Performing Particle Image Velocimetry using Artificial Neural Networks

Jean Rabault & Jostein Kolaas

jean.rblt@gmail.com

University of Oslo, Department of Mathematics

Artificial Neural Networks (ANNs) are increasingly successful at solving image analysis problems. Image analysis is widely used in Fluid Mechanics when performing Particle Image Velocimetry (PIV), and therefore it is natural to test the ability of ANNs to perform such tasks. We report for the first time the use of Convolutional Neural Networks (CNNs) and Fully Connected Neural Networks (FCNNs) for performing end-to-end PIV. Realistic synthetic images are used for training the networks and several test cases are used to assess the quality of each network predictions and compare them with state-of-the-art PIV softwares. While the ANNs we present have slightly higher Root Mean Square (RMS) error than state-of-the-art cross-correlation methods, they perform better near edges and allow for higher spatial resolution than such methods.

Introduction

Experimental Fluid Mechanics relies on using image processing for measuring flow velocities. Particle Image Velocimetry (PIV) is one such method which usually relies on computing the cross-correlation of a spatial window between two frames for finding a correlation peak, indicating the most probable displacement of the flow. PIV algorithms have been refined and complexified with time so that several correlation techniques can be used, subpixel accuracy can be achieved, and outliers can be automatically detected and interpolated.

On many aspects, the methods currently used for performing PIV rely on complex feature engineered algorithms. Based on the trends observed in other branches of image processing, and in particular the increasing performance of ANNs, one can expect that well designed ANNs should become better at performing PIV than the algorithms used today.

Image generation

Synthetic data are used to create arbitrarily big labeled training dataset and therefore issues common with ANN training, such as overfitting, are avoided. A dataset used for training the ANNs is composed of a collection of pairs of 32×32 pixels images and their labels. Second order polynomials are created for the u and v components of the velocity as:

$$\begin{cases} u(x,y) = U_0 + \bar{J}_u \cdot \bar{r} + \bar{r}^T \cdot \bar{\bar{H}}_u \cdot \bar{r} \\ v(x,y) = V_0 + \bar{J}_v \cdot \bar{r} + \bar{r}^T \cdot \bar{\bar{H}}_v \cdot \bar{r} \end{cases} \tag{1}$$

where the position of the point considered is $\bar{r} = (x, y)$, (U_0, V_0) is the velocity at the centre of the image, and the Jacobian and Hessian tensors of the velocity component i at the centre of the image are $\bar{J}_i$ and $\bar{\bar{H}}_i$, respectively. U_0, V_0, $\bar{J}_i$ and $\bar{\bar{H}}_i$ are randomly generated from the uniform distribution. U_0 and V_0 are in the range $\pm 4 pixels/frame$, J_i in the range $\pm 0.05/frame$, and H_i in the range $\pm 0.001/pixels\ frame$.

Once the random velocity field has been generated, a set of initial positions for the tracer particles is drawn from a random uniform distribution. Those are advected as:

$$\frac{d\bar{x}_p}{dt} = \bar{u}(\bar{x}_p(\bar{x}_0, t), t), \tag{2}$$

with $\bar{x}_p(\bar{x}_0, t)$ the position of the particle p initially at the position $\bar{x}_0$ after a time t. Equation (2) is integrated using the Runge-Kutta 4 method for generating the particles positions in both frames of the image pair.

Sufficiently small particles imaged by a camera form circular patterns known as Airy disks. The central lobe of the Airy disks are normally well approximated by Gaussian bell curces:

$$I(x,y) = I_0 \exp\left(\frac{-(x-x_0)^2 - (y-y_0)^2}{(1/8)d_\tau^2}\right), \tag{3}$$

where I_0 is the particle luminosity, d_τ the effective particle diameter, and (x_0, y_0) the position of the particle centre. I_0 and d_τ are random and drawn from a uniform distribution independantly for each particle. As a final step a 1% Gaussian white noise is added to the images. Typical results are presented in Fig. 1.

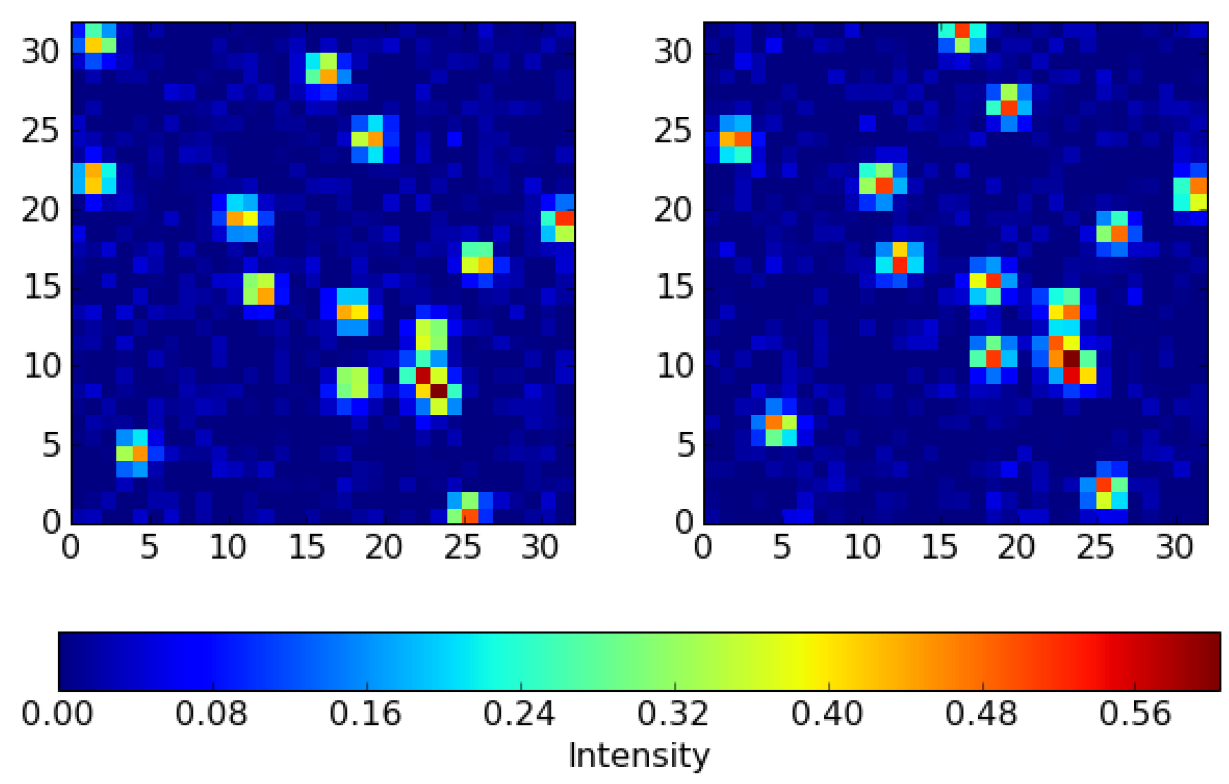

Figure 1: Example of a couple of synthetic images used for training the ANNs.

ANNs and PIV codes

The CNN used is composed of a single convolution layer featuring 512 kernels of size 16×16 pixels and depth 2 applied with a stride of 8 pixels using zero padding. The sizes of the fully connected layers are 8192, 4096, 2048 and 6 neurons going downwards in the network. The first three layers use Rectified Linear Units (ReLUs) of slope 0.1 for negative x values. The last layer uses linear activation function for producing the output prediction of the network. The whole network is trained using the Adam optimizer. The cost function to be minimized is the absolute norm of the prediction error.

The FCNN has a total of 6 layers. The first five layers feature 4096 ReLUs, and the last layer has 6 linear units that are used to produce the network prediction. ReLUs have a slope 0.1 for negative x values, the Adam optimizer is used to minimize the absolute norm of the prediction error and no regularization is imposed on the network.

Comparisons were performed against LaVision, a market leading commercial code, and HydrolabPIV, an in-house PIV code developped at the University of Oslo. Traditional PIV codes were used on extended images of size 128x128 pixels so that mutlipass can be applied. The velocity from the center 32x32 subwindow was extracted for generating the predictions of the ANNs.

Results and conclusion

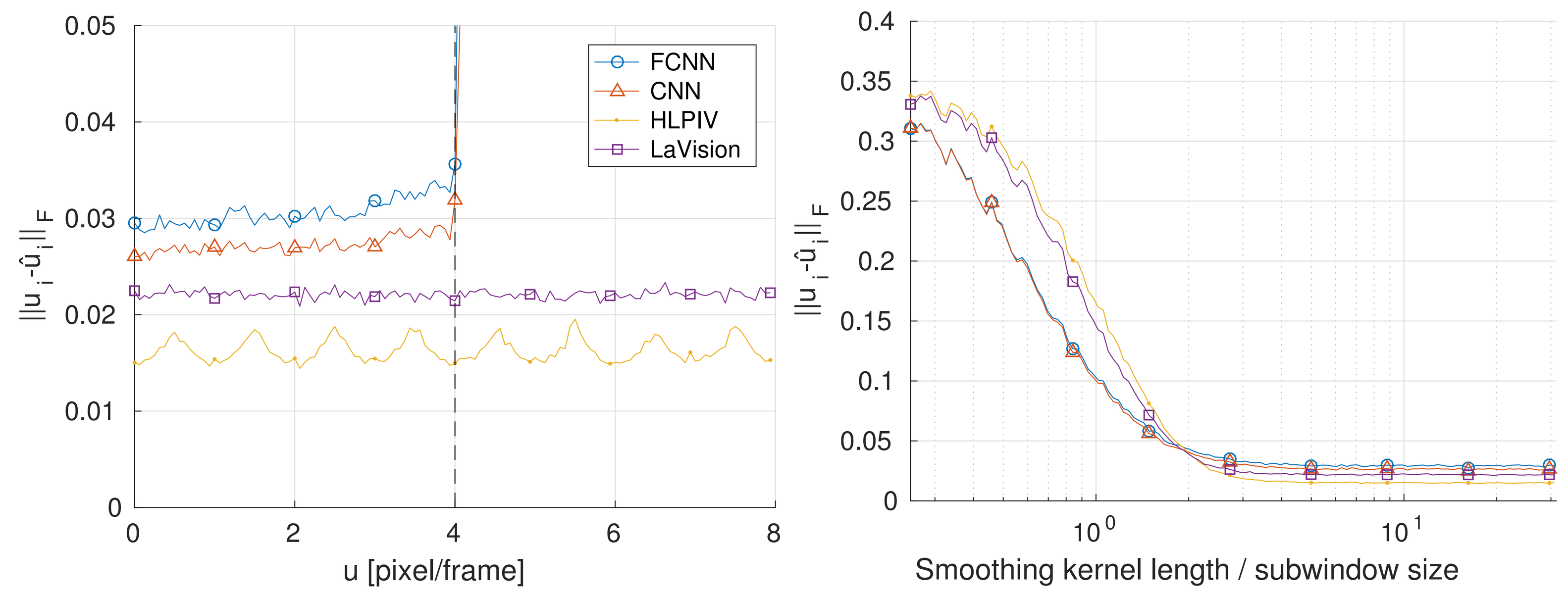

Figure 2: Left: RMS of the velocity error estimates in the Translation test case. The vertical line indicates the maximum value of the parameters selected when generating the pictures used for training the networks. Right: investigation of the spatial resolution of each method, performed by assessing the quality of the predictions obtained on velocity fields featuring a range of typical autocorrelation length.

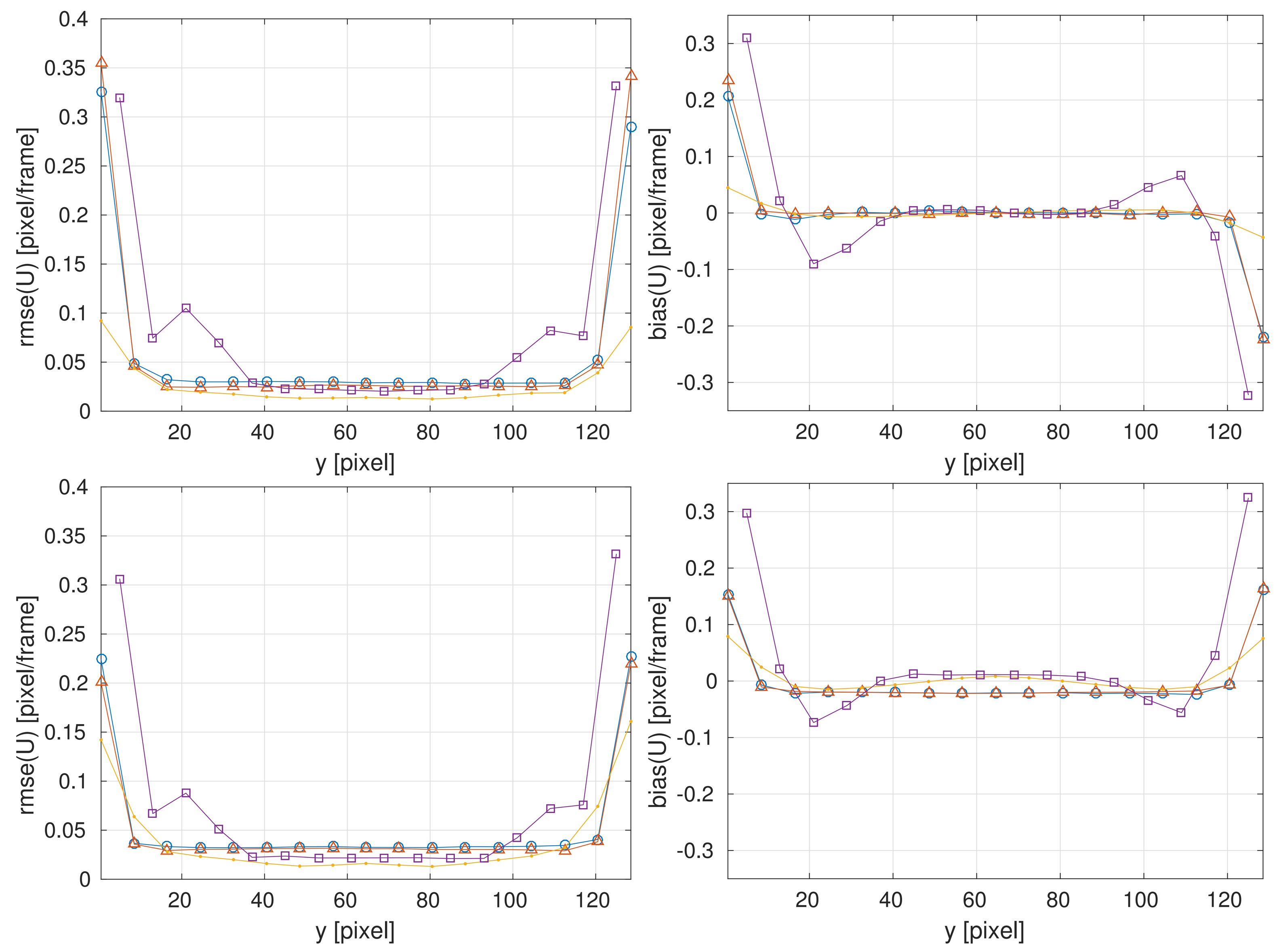

Figure 3: RMS error (left) and bias error (right) of the X component of the velocity estimates obtained with realistic velocity fields. Top: pure gradient, bottom: pure curvature.

	CNN		FCNN		LaVision	
	GTX970	GTX980TI	GTX970	GTX980TI	CPU	GTX970
Loading to RAM from HDD	8.2	5.4	7.5	6.0	-	-
Generating slice from RAM	0.4	0.3	0.2	0.2	-	-
Computation on GPU	22.9	7.2	15.9	5.1	-	-
Writing result to HDD	1.5	0.9	1.5	1.0	-	-
Total time	33.0	13.8	25.1	12.3	70.6	55.0

Table 1: Computational times (in seconds) for 128000 vectors measured using CNN, FCNN and LaVision.

The level of Root Mean Square (RMS) error between ANNs predictions and the velocity values used for generating the images is slightly higher than for state-of-the-art PIV codes. However ANNs may have several advantages over more traditional 2D PIV methods. Benchmarking shows that ANNs are better at using efficiently GPUs. ANNs have better resolution compared with traditional PIV methods, which could be of interest in cases when high local flow variations are expected. We also observed good boundary performance of ANNs compared with more traditional PIV methods.

Acknowledgements

This work was partially financed by the research project DOMT - Developments in Optical Measurement Technologies (project number 231491) funded by the Research Council of Norway.