A deep-learning model for displacement measurement in photomechanics

MécaMat: demi-journée *Apprentissage* et *Mesures de champs*

27 mai 2025

Frédéric Sur

https://members.loria.fr/FSur

LORIA Tangram / Université de Lorraine, CNRS, Inria / Nancy

with colleagues from Institut Pascal (Université Clermont-Auvergne):

experimental mechanics: Benoît Blaysat and Michel Grédiac

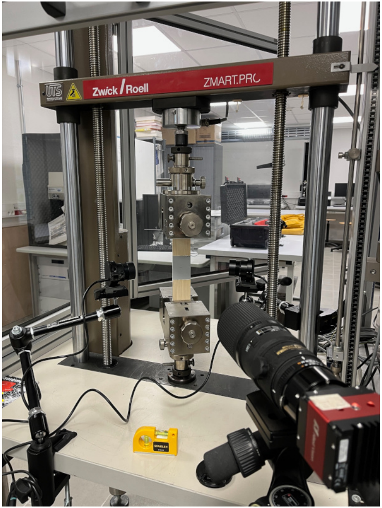
hardware and software architecture for signal processing:

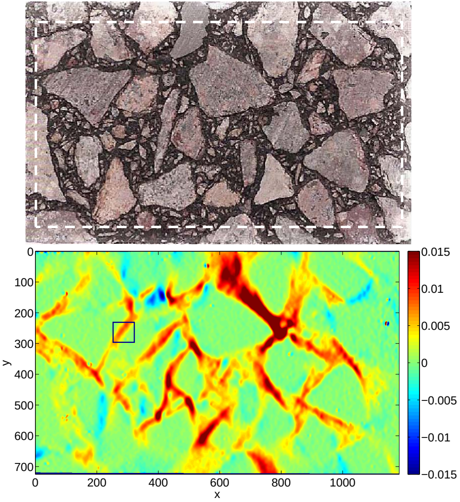
Kamel Abdelouahab, Anas Bahou, François Berry, Seyfeddine Boukhtache

S. Boukhtache, K. Abdelouahab, F. Berry, B. Blaysat, M. Grédiac, F. Sur. *When deep learning meets digital image correlation*. Optics and Lasers in Engineering, vol. 136, p. 106308, 2021

S. Boukhtache, K. Abdelouahab, A. Bahou, F. Berry, B. Blaysat, M. Grédiac, F. Sur. *A lightweight convolutional neural network as an alternative to DIC to measure in-plane displacement fields*. Optics and Lasers in Engineering, vol. 161, p. 107367, 2023

Strain (displacement derivatives) measurement





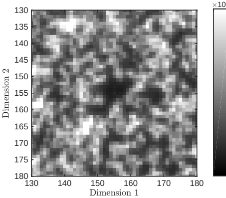
→ in these maps, a 1mm element deforms (at most) to 1.015mm

Full-field photo-extensometry: experimental process

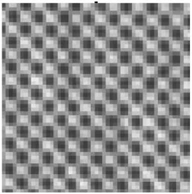
1. surface of the specimen marked with a contrasted pattern
2. take a picture, mechanical deformation, take another picture
3. subpixel displacement field retrieved by registering patches

Until now, two families of methods were available:

- *random speckle*: **Digital Image Correlation**
  light implementation, but slow
  → most widely used method in photomechanics: large literature, dedicated conferences, commercial and free software...



- *periodic pattern*: **Localized Spectrum Analysis**
  (see also GPA, WGPA, Sampling Moiré...)
  heavy implementation but fast
  (direct in Fourier domain)



B. Blaysat, F. Sur, T. Jailin, A. Vinel, M. Grédiac. *OpenLSA: an open-source toolbox for computing full-field displacements from images of periodic paterns*, SoftwareX, vol. 27, p. 101826, 2024

Looks like optical flow estimation in computer vision!

Source: `https://cs.brown.edu/courses/csci1290/2011/results/final/psastras/`

Applications: autonomous driving, traffic monitoring, crowd monitoring, sports analytics, visual effects for video production…

State-of-the-art: deep-learning approaches

FlowNet, Proc. International Conference on Computer Vision, 2015

FlowNet: Learning Optical Flow with Convolutional Networks

Alexey Dosovitskiy[*], Philipp Fischer[†*], Eddy Ilg[*], Philip Häusser, Caner Hazırbaş, Vladimir Golkov[†]
University of Freiburg   Technical University of Munich
{fischer,dosovits,ilg}@cs.uni-freiburg.de, {haeusser,hazirbas,golkov}@cs.tum.edu

Patrick van der Smagt
Technical University of Munich
smagt@brml.org

Daniel Cremers
Technical University of Munich
cremers@tum.de

Thomas Brox
University of Freiburg
brox@cs.uni-freiburg.de

Abstract

Convolutional neural networks (CNNs) have recently been very successful in a variety of computer vision tasks, especially on those linked to recognition. Optical flow estimation has not been among the tasks CNNs succeeded at. In this paper we construct CNNs which are capable of solving the optical flow estimation problem as a supervised learning task. We propose and compare two architectures: a generic architecture and another one including a layer that correlates feature vectors at different image locations. Since existing ground truth data sets are not sufficiently large to train a CNN, we generate a large synthetic Flying Chairs dataset. We show that networks trained on this unrealistic data still generalize very well to existing datasets such as Sintel and KITTI, achieving competitive accuracy at frame rates of 5 to 10 fps.

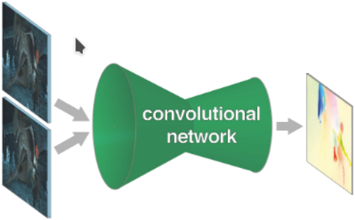
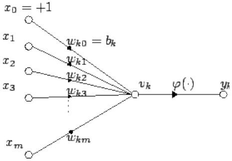


Figure 1. We present neural networks which learn to estimate optical flow, being trained end-to-end. The information is first spatially compressed in a contractive part of the network and then refined in an expanding part.

Artificial neural networks

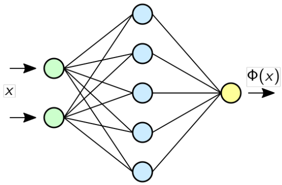
Artificial neuron



output of neuron k: $y_k = \varphi\left(w_{k0} + \sum_{i=1}^{m} w_{ki} x_i\right)$

$(x_i)_{i=1\ldots m}$: **input**
φ: activation function (non-linear)
$(w_{ki})_{i=0\ldots m}$: **weights** (free parameters)

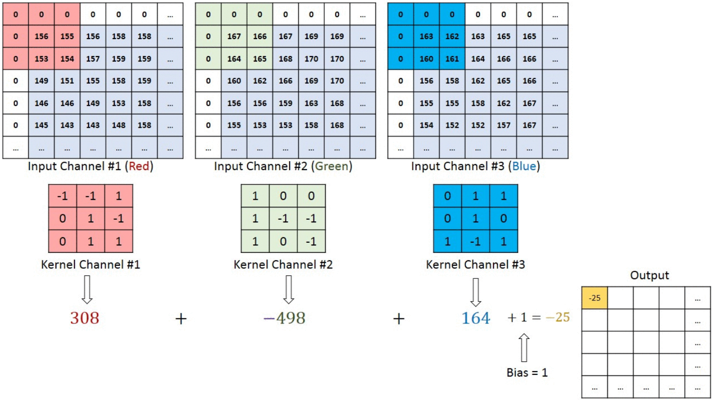
(Feedforward) neural network / Multilayer perceptron



Training: requires a dataset $\mathcal{T}$ made of samples (x, y)
1) $(x, y) \in \mathcal{T}$, compute prediction error $\ell_w(\Phi(x), y)$
2) tune weights w to reduce the error
3) back to 1)
ℓ_w: **loss** function

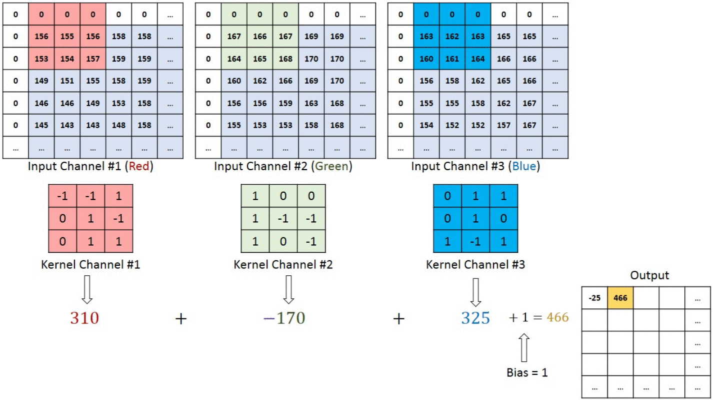
Inference: for an unseen x, predict $\Phi(x)$

Convolutional layer (here: three entry layers)



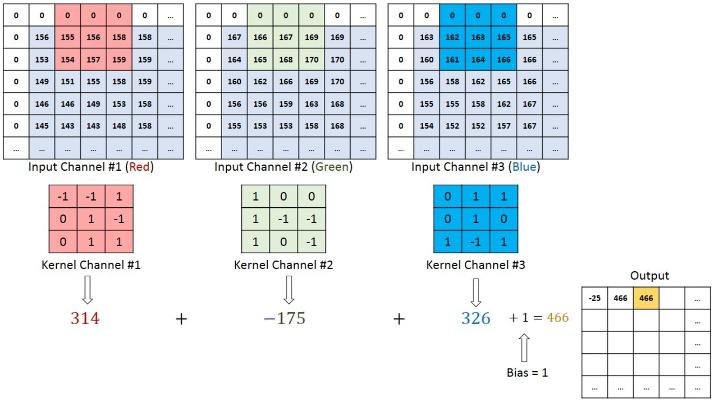
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Convolutional layer (here: three entry layers)



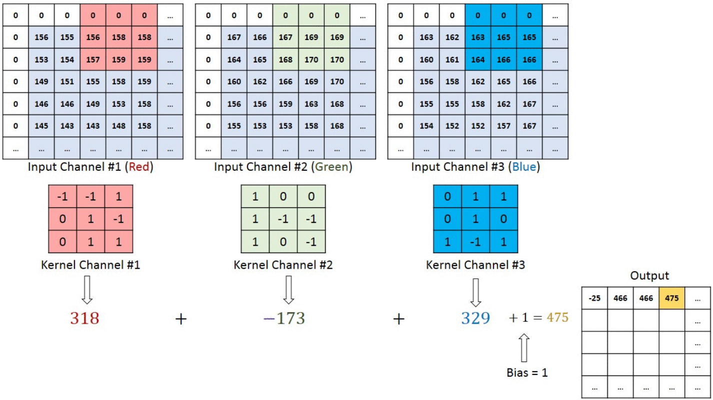
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Convolutional layer (here: three entry layers)



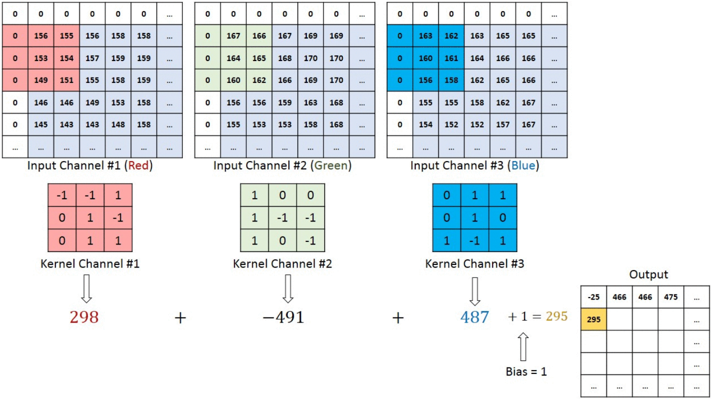
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Convolutional layer (here: three entry layers)



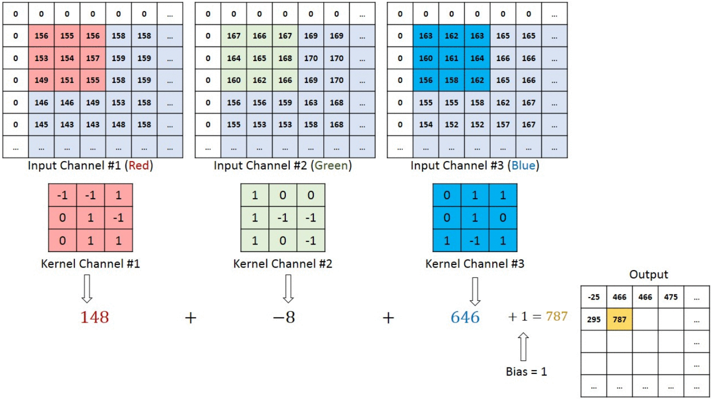
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Convolutional layer (here: three entry layers)



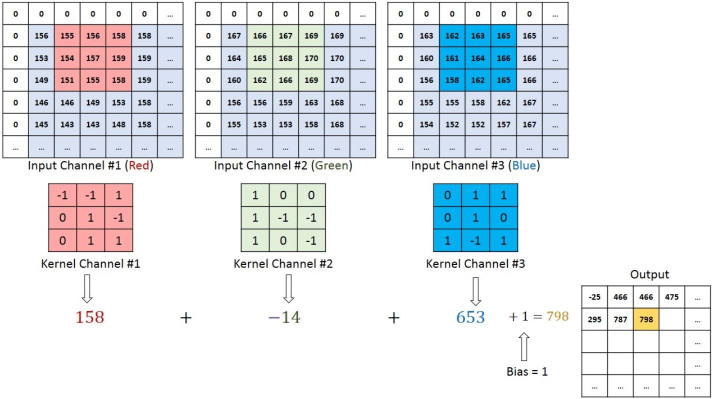
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Convolutional layer (here: three entry layers)



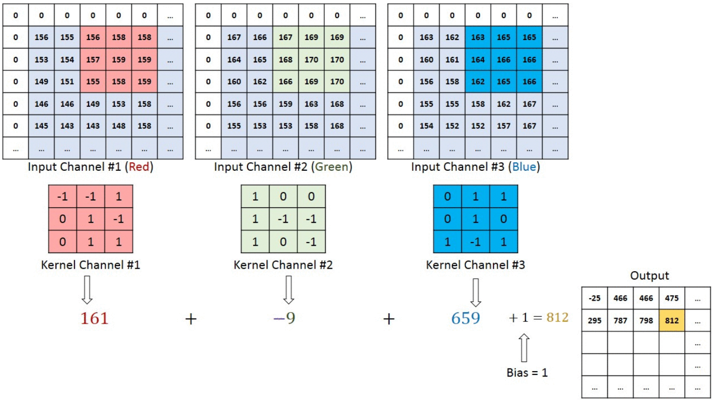
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Convolutional layer (here: three entry layers)



https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

Convolutional layer (here: three entry layers)

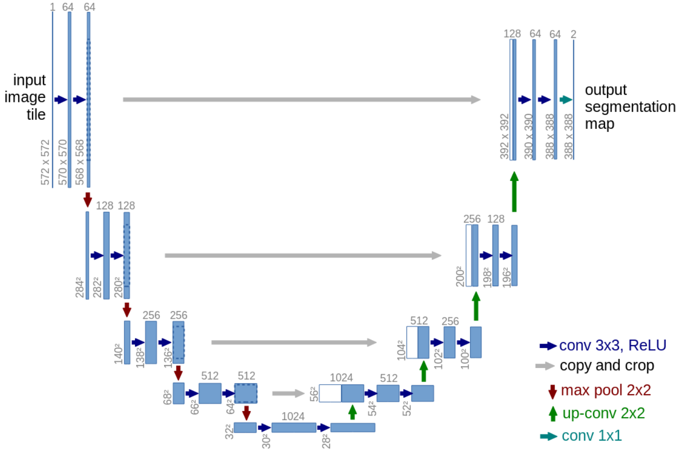


https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

convolution is a particular linear operation between layers of neurons (with **shared weights**)

→ convolution parameters are **learnt**
(contrary to filter design in traditional signal processing)

What is a U-Net?



$\rightarrow$ reference *image-to-image* model

$\rightarrow$ belongs to the family of *Convolutional Neural Network* (CNN)

O. Ronneberger, P. Fischer, T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, Proceedings MICCAI 2015. $>$ 110 000 citations!

The flying chairs dataset

22,872 frame pairs with ground truth (synthetic images)

Back to photomechanics

Observation: *applying deep-learning models dedicated to optical flow to speckle images gives nothing interesting...*

Optical flow vs. displacement in mechanics:

Images are different:

- natural images vs. speckle images
- image diversity vs. similar images

Optical flow and mechanical displacement are different:

- moderate accuracy (0.5-1px) vs. subpixel accuracy (0.1px)
- 3D motion of objects vs. mechanical deformation of a specimen (potentially out-of-plane displacement with stereo-DIC)
- occlusions but smooth motion vs. no occlusions but potentially large displacement gradient

Fine-tuning existing CNNs over a dedicated training set

36,663 pairs of 256×256 **synthetic** speckle images with different statistical properties

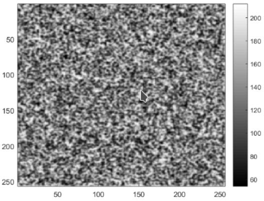
Random displacement between -1 and 1 pixel every 8 pixels, bilinear interpolation in-between.

Rendered with *BSpeckleRender*

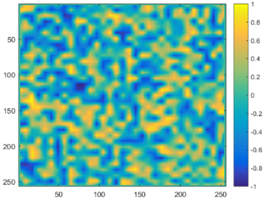
F. Sur, B. Blaysat, M. Grédiac. Rendering deformed speckle images with a Boolean model. JMIV 2018

`https://members.loria.fr/FSur/software/BSpeckleRender/`

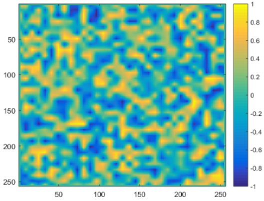
Typical example:



reference image



x-displacement

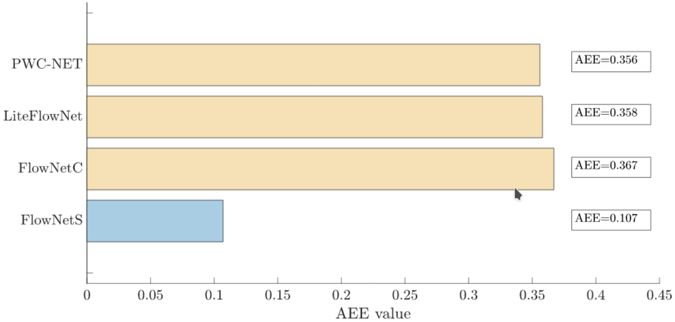


y-displacement

Holdout validation after fine-tuning

Average Endpoint Error: $AEE = \frac{1}{KL} \sum_{i=1}^{K} \sum_{j=1}^{L} \|\widehat{\mathbf{u}}(i,j) - \mathbf{u}(i,j)\|_2$

with $\widehat{\mathbf{u}}$ estimated 2D displacement and $\mathbf{u}$ ground truth



$\rightarrow$ PWC-NET: CVPR18 LiteFlowNet: CVPR18 FlowNet: ICCV15

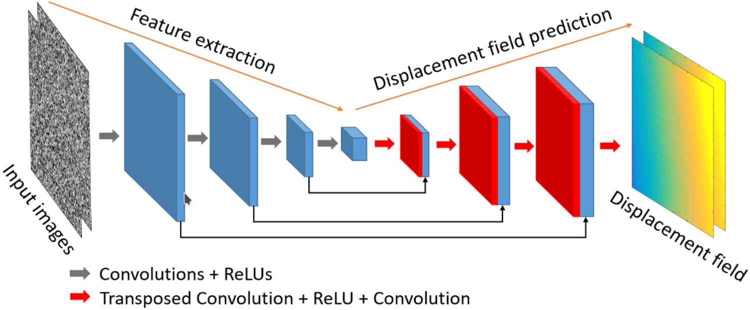
DIC or **LSA**: AEE $\simeq$ 0.01 - 0.05 pixel...

Adapting FlowNetS

FlowNetS: U-Net, **eight** /2 downsampling, **six** x2 upsampling
$\rightarrow$ dimension of optical flow map = dimension of images / 4,
then bilinear interpolation to upsample to the original resolution

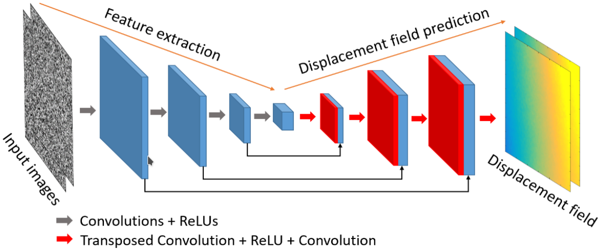
To make a long story short...

What we propose: **StrainNet-f**



S. Boukhtache, K. Abdelouahab, F. Berry, B. Blaysat, M. Grédiac, F. Sur. *When deep learning meets digital image correlation*. Optics and Lasers in Engineering, vol. 136, p. 106308, 2021

StrainNet-f



layer 1: 64 filters of size 7×7, stride=2 (downsampling /2)
layer 2: 128 filters of size 5×5, stride=2
layer 3: 256 filters of size 3×3, stride=2
layer 4: 512 filters of size 3×3, stride=2
etc. (see papers)

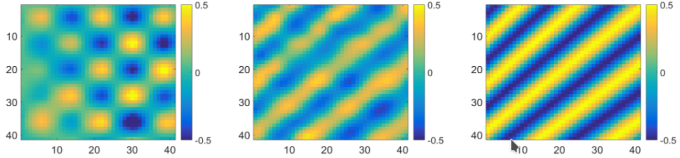
$\rightarrow \simeq$ 39 million parameters

$\rightarrow$ receptive field $\simeq 500 \times 500$ pixels

Training loss: $\sum_{i=1}^{4} \lambda_i e_i$,
with e_i = error of the output of each upsampling layer; λ_i= fixed weights
$\rightarrow$ deep supervision strategy

Adapting the training dataset for fine-tuning

The training dataset has to be carefully designed:



(a) StrainNet-h trained on Speckle dataset 1.0

(b) StrainNet-h trained on Speckle dataset 2.0

(c) Reference displacement

New dataset:
random displacement (between -1 and 1) at pixels 4, 8, 16, 32, 64, 128 pixels apart
bicubic interpolation in-between
Poisson-Gaussian noise added to speckle images

Fine-tuning: start from the pre-trained weights of FlowNetS

Lightweight models

$\rightarrow$ DIC from speckle “easier” than optical flow estimation from natural images. . .

$\rightarrow$ Several lightweight models proposed in OLEN23

Examples:

Model	StrainNet-F	StrainNet-F3	StrainNet-F2	StraiNet-F1
Param. ($\times 10^6$)	38.68	8.69	3.69	1.31
AAE (pixels)	0.0266	0.0230	0.0233	0.0391

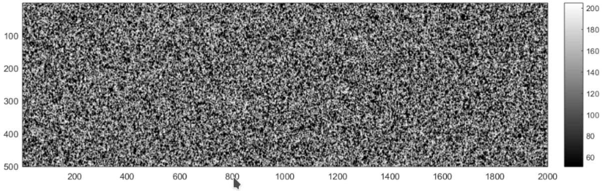
S. Boukhtache, K. Abdelouahab, A. Bahou, F. Berry, B. Blaysat, M. Grédiac, F. Sur. *A lightweight convolutional neural network as an alternative to DIC to measure in-plane displacement fields*. Optics and Lasers in Engineering, vol. 161, p. 107367, 2023

Assessment 1: synthetic image

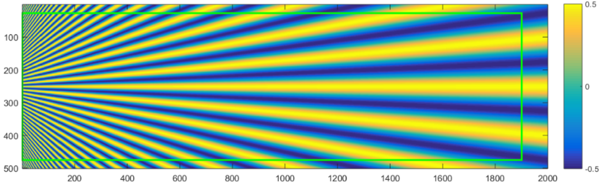
Test: pair of images from DIC challenge 2.0

`https://idics.org/challenge/`

generated by BSpeckleRender



(a) Reference image



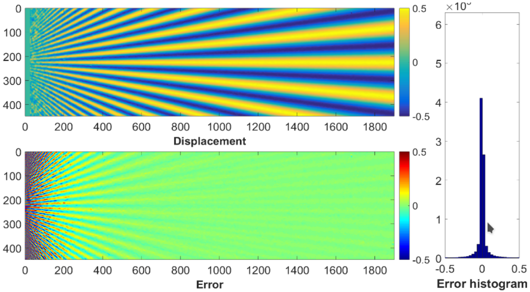
(b) Reference displacement

→ "frequency response" of the estimation method

Assessment 1: synthetic image

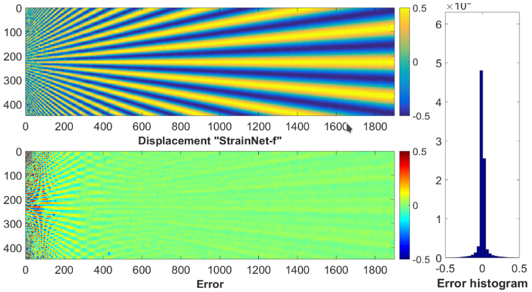
DIC at every pixel
11 $\times$ 11 subsets
2^{nd} order shape function

AEE: 0.0365[px]

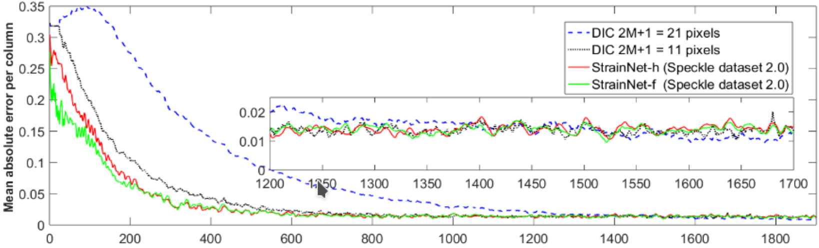


StrainNet-f

AEE: 0.0266[px]



Assessment 1: synthetic image



$\rightarrow$ high-frequency displacement:

$\text{error}_{\text{StrainNet-f}} < \text{error}_{\text{StrainNet-h}} < \text{error}_{\text{DIC 21}} < \text{error}_{\text{DIC 11}}$

$\rightarrow$ (very) low-frequency displacement:

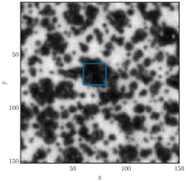
$\text{error}_{\text{DIC 21}} < \text{error}_{\text{DIC 11}} \simeq \text{error}_{\text{StrainNet-f}} \simeq \text{error}_{\text{StrainNet-h}} < 0.02[px]$

Assessment 2: another synthetic image

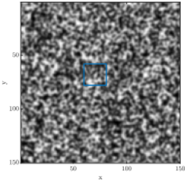
Speckle image pairs, **not** generated with BSpeckleRender

J. Orteu, D. Garcia, L. Robert, and F. Bugarin. *A speckle texture image generator*. Proceedings SPIE 2006

(from DIC challenge 1.0)



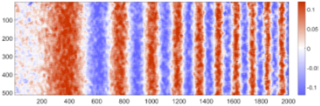
(a) Close-up view of the images of Sample 14 of the DIC challenge 1.0. A subset used in DIC is superimposed (size: $2M + 1 = 21$ pixels).



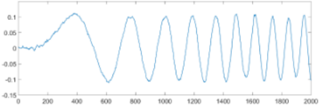
(b) Close-up view of one of the images of Speckle dataset 2.0 used to train the network.

Goal: test the generalization capability of StrainNet

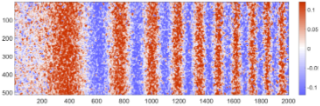
Assessment 2: synthetic image



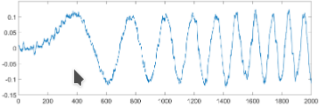
(c) u_x, DIC, $2M+1=21$ pixels



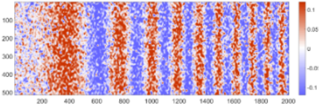
(d) $\overline{u}_x$, DIC, $2M+1=21$ pixels



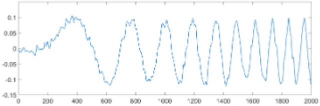
(e) u_x, DIC, $2M+1=11$ pixels



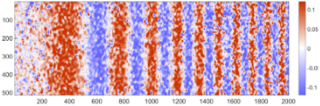
(f) $\overline{u}_x$, DIC, 11, $2M+1=11$ pixels



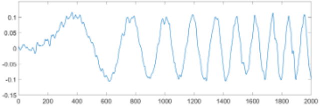
(g) u_x, StrainNet-h



(h) $\overline{u}_x$, StrainNet-h



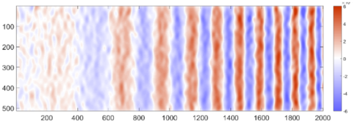
(i) u_x, StrainNet-f



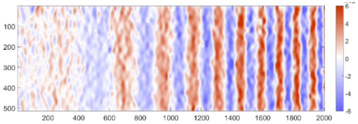
(j) $\overline{u}_x$, StrainNet-f

Assessment 2: synthetic image

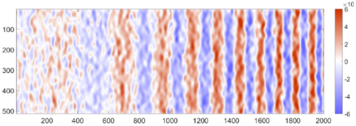
Strain component: $\varepsilon_{xx} = \dfrac{\partial u_x}{\partial x}$



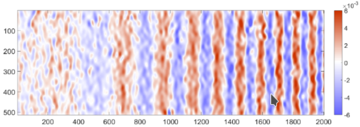
(a) DIC, $2M + 1 = 21$ pixels



(b) DIC, $2M + 1 = 11$ pixels



(c) StrainNet-h



(d) StrainNet-f

$\rightarrow$ StrainNet seems to generalize well

(i.e., StrainNet does not seem to depend on the training dataset)

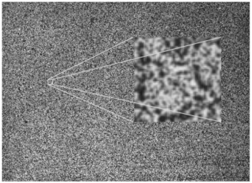
$\rightarrow$ strain estimation comparable to DIC

Assessment 3: real image (no ground truth)

Wood specimen: a stack of early and late wood (rings)

Consequence: stiffness spatially changes, and so does the strain distribution (rings are perpendicular to the loading force)

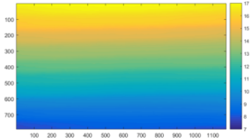
(a) Specimen before spray-painting,



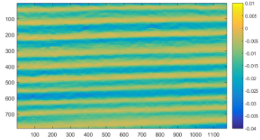
(b) Speckled surface of the specimen after spray painting,

M. Grédiac, B. Blaysat, F. Sur. *A robust-to-noise deconvolution algorithm to enhance displacement and strain maps obtained with local DIC and LSA*. Experimental Mechanics, 59(2):219–243, 2019

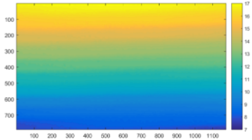
Assessment 3: real image (no ground truth)



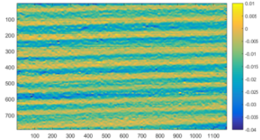
(c) DIC, $2M+1=21$ pixels



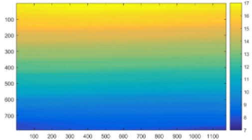
(d) DIC, $2M+1=21$ pixels



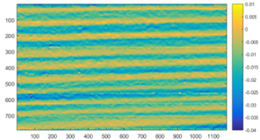
(e) StrainNet-h



(f) StrainNet-h



(g) StrainNet-f



(h) StrainNet-f

u_y ε_{yy}

$\rightarrow$ once again, results are visually similar

Typical calculation time

1) Training time (fine-tuning): a few hours to several days (depending on the model complexity) on NVIDIA Tesla V100

2) Inference: estimating a new displacement field once the network is trained...

StrainNet:
NVIDIA Tesla V100 (114 TFLOPs)

StrainNet-f: $1.3 \cdot 10^7$ pixels per second
StrainNet-h: $1.3 \cdot 10^8$ pixels per second

DIC implemented on GPU (code not available):
NVIDIA GTX 760 (2.3 TFLOPs)

$1.5 \cdot 10^5$ pixels per second $\rightarrow \simeq 7 \cdot 10^6$ pixels per second on V100?

L. Zhang, T. Wang, Z. Jiang, Q. Kemao, Y. Liu, Z. Liu, L. Tang, and S. Dong. *High accuracy digital image correlation powered by GPU-based parallel computing*. Optics and Lasers in Engineering, 2015

Conclusion

- a new full-field measurement method, after DIC and LSA / GPA / WGPA / SM (Fourier)
- accuracy similar to DIC with StrainNet, better in following papers
- much smaller measurement bias than DIC over high-frequency displacements
- calculation time compatible with real-time measurement

Pytorch implementation:
`https://github.com/DreamIP/StrainNet`

Perspectives

- further assessment of generalization capability: consider a wider panel of speckle patterns (synthetic or real)
- real experiments (but no ground truth. . .)
- redesign the network
  spatial pyramidal scheme for displacements > 1 pixel ?
- quantify metrological performance? (resolution, bias)
- what about Pattern Induced Bias ?
  F. Sur, B. Blaysat, M. Grédiac. *On biases in displacement estimation for image registration, with a focus on photomechanics*. Journal of Mathematical Imaging and Vision, vol. 63, no. 7, p. 777-806, 2021
- More recent models for optical flow estimation ?
  In particular:
  J.J. Yu, A.W. Harley, K.G. Derpanis. *Back to Basics: Unsupervised Learning of Optical Flow via Brightness Constacy and Motion Smoothness*. Proc. ECCV Workshop 2016
  $\rightarrow$ in DIC, amounts to modeling the value of the shape function by the output of a neural network