A lightweight convolutional neural network as an alternative to DIC to measure in-plane displacement fields

S. Boukhtache, K. Abdelouahab, A. Bahou, F. Berry, B. Blaysat, M. Grédiac, Frédéric Sur

A lightweight convolutional neural network as an alternative to DIC to measure in-plane displacement fields

S. Boukhtache[1,2], K. Abdelouahab[3], A. Bahou[1], F. Berry[1], B. Blaysat[1], M. Grédiac[1†], F. Sur[4]

[1] *Institut Pascal, UMR 6602, Université Clermont-Auvergne, CNRS, SIGMA Clermont, Clermont-Ferrand, France*

[2] *Present address: I-Virtual, Metz, France*

[3] *Sma-RTy SAS, Aubière, France*

[4] *LORIA, UMR 7503, Université de Lorraine, CNRS, INRIA, Nancy, France*

Abstract

Convolutional Neural Networks (CNNs) are now commonly used in the computer vision community, in particular for optical flow estimation. Some attempts to use such tools to measure displacement and strain fields from pairs of reference/deformed speckle images (like Digital Image Correlation) have been recently reported in the literature. The aim of this work is twofold. The first one is to customize a state-of-the-art CNN dedicated to optical flow estimation to reach better performance when processing speckle images. This is mainly obtained by removing the deepest levels. The second one is to further simplify the CNN by reducing as much as possible the number of filters in the remaining levels while keeping equivalent metrological performance to the original version, in order to accelerate image processing on a power-efficient compact Graphics Processing Unit (GPU).

Synthetic images deformed through a suitable displacement field are used to assess the metrological performance of the different versions of the CNN tested in this study. We focus on the sub-pixel part of the displacement is considered for this first attempt, this part being much more challenging to determine than integer displacements obtained at the pixel scale. The latter can be found by cross-correlation or with a rough version of DIC. Real images are tested with the simplest version of the CNN and obtained results are compared with those provided by classic subset-based Digital Image Correlation. The two main conclusions are *i*- that the customization procedure improves the metrological performance of the original version and *ii*- that the metrological performance of the ultimate simplified version of the CNN is globally equivalent to the one of the initial version despite the drastic simplification obtained at the end of the procedure. This performance lies between that of DIC used with first- and second-order subset shape functions.

Keywords: *Convolutional Neural Network, Deep learning, Digital Image Correlation, Error Quantification, Graphics Processing Unit, Photomechanics, Speckle*

1 Introduction

During the recent years, Digital Image Correlation (DIC) has widely spread in the experimental mechanics community because it offers a good tradeoff between ease of use and metrological performance. However, as any other measuring technique, DIC suffers from some limitations. The main one is certainly the fact that DIC relies on the iterative minimization of an optical residual estimated over small zones of speckled surface images, namely the subsets, and this induces a significant computational cost to extract displacement and strain maps. This is all the truer as the number of pixels of camera sensors increases with time, which automatically increases this computational cost. This limitation motivates the study of procedures speeding up DIC calculations, if not completely revisiting the principle of DIC itself by proposing alternative methods. The first approach is illustrated by the parallelization of the DIC code and its implementation in a Graphical Processing Unit (GPU), as proposed in [1, 2] for instance in the case of classic subset-based and integrated DIC, respectively. A significant reduction of the computing time is observed in both cases. Concerning the second approach, it has been proposed in [3, 4]to completely depart from the classic minimization of the optical residual over small subsets performed by classic DIC by employing Convolutional Neural Networks (CNNs). Indeed, a recent yet wide literature is available in the computer vision community to resolve optical flow problems with CNNs. These problems are similar to the one tackled by DIC in experimental mechanics, namely measuring displacement fields, apart from the amplitude of the displacement which is generally much lower in the latter case since sub-pixel resolution must be reached. The hindsight on employing CNNs instead of DIC to measure displacement and strain fields is by far insufficient to have a clear view on the numerous parameters which govern the quality of the solution. For instance, the CNNs developed and used in [3] were largely inspired from others already described in the literature for resolving the problem of optical flow estimation such as FlowNet [5, 6]. However, CNNs developed and discussed in [3] are deep, in the sense that the number of convolutional layers, and thus the

number of parameters governing them, is significant, since up to 38.60 millions coefficients are needed to define them. In this context, the aim of this paper is twofold. The first objective is to further adapt these CNNs to sub-pixel displacement field determination. The second one is to examine to what extent these CNNs can be simplified and lightened, so that they can be embedded in a power-efficient and compact GPU, which is the first step toward their integration in a smart camera providing displacement and strain fields in quasi-real time.

The paper is organized as follows: we first briefly recall how a CNN works, and give some details on the architecture of pre-existing CNNs aimed at retrieving displacement fields from pairs of reference/deformed speckle images. The adaptation and the simplification of these CNNs are then explained and their impact on the metrological performance is discussed by processing suitable synthetic images of speckle patterns. Results obtained after implementing the original CNNs and their simplified version on a "Jetson Xavier NX" low-cost embedded GPU are also discussed. The determination of the strain field obtained during a real compression test on a wood specimen is finally given.

2 StrainNet: how it works

2.1 Architecture

2.1.1 A reminder about StrainNet

A CNN dedicated to displacement field measurement is made of successive pyramidal levels, each of them containing layers. Two CNNs, namely StrainNet-f and StrainNet-h, were proposed in Ref. [3] to retrieve sub-pixel displacement and strain maps from pairs of reference/deformed speckle images. Their architecture was inspired from that of FlowNet-S [5, 6]. Suffix "-f", which is the initial of "full", is justified by the fact that the size of the maps is the same as the size of the speckle images. Suffix "-h", which is the initial of "half", is justified by the fact that the size of the maps is half the size of the speckle images. The architecture of StrainNet-f and StrainNet-h being similar, their general principle is illustrated

in Figure 1 with StrainNet-f as an example. This CNN is made of two main parts: feature extraction on the left and determination of the displacement field on the right. The feature extraction part consists of several convolution layers followed by an activation function. As in FlowNet-S, we chose here the leaky Rectified Linear Unit (leaky ReLU) [7], which is one of the most used activation functions in deep learning [8, 9]. This function is merely defined by $LeakyReLU(x) = max(0, x) + c \times min(0, x)$, with $c = 0.1$. This version keeps the negative values provided by the layer, contrary to other some usual versions of ReLU, for which only positive values are kept, such as $ReLU(x) = max(0, x)$. Some layers have a stride of two to perform down-sampling. Such a stack of several layers is called a level. The output of these levels can thus be represented by narrower and narrower blocks, in yellow in this figure. In the part dedicated to the determination of the displacement field, the levels are made of transposed convolution layers for up-sampling, and convolution layers followed again by a ReLU. The input of each level of this part is the output of the preceding level, concatenated with the output of the same level of the feature extraction part, as represented by the black arrows in Figure 1. The reader interested in more details on how CNNs work is referred to Section 2 "A short primer on deep learning" of Ref. [3].

Since the aim of this contribution is to reduce the number of levels and filters of two preexisting networks, among which StrainNet-f roughly illustrated in Figure 1, the following step is to describe in more details the architecture of this network as an example. StrainNet-h will be briefly described shortly after.

2.1.2 Architecture of StrainNet-f

StrainNet-f is made of 23 layers of four different types represented here by 23 pairs of vertical bars. The first layer (on the left) is the input layer. In the present case, this is a pair of images representing the reference and the deformed states. The last one (on the right) is the output layer. In the present case, this is a set of two displacement maps (along the $x-$ and $y-$ directions). The layers in between are called hidden layers. Each layer processes data

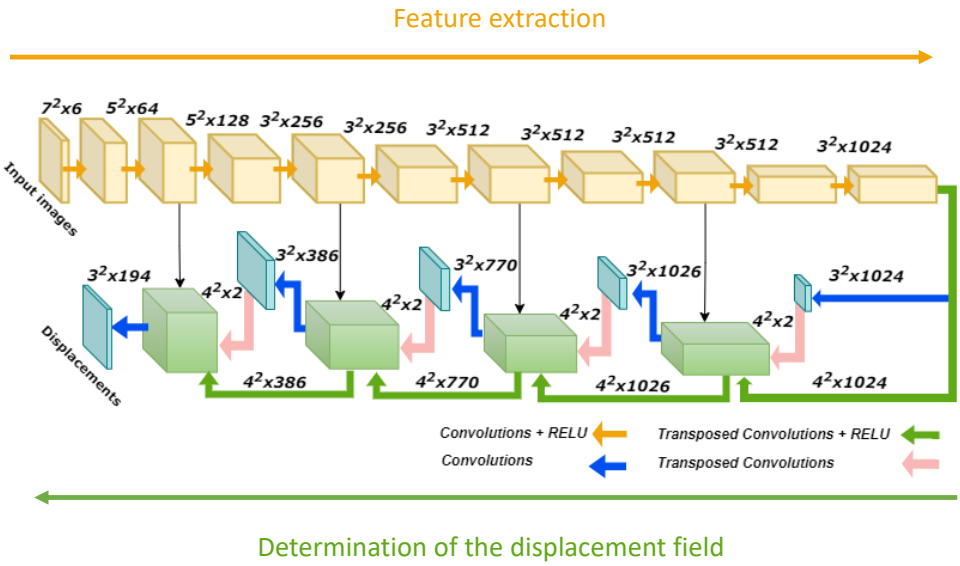


Figure 1: Schematic view of a convolutional neural network, with StrainNet-f introduced in [3] as a typical example. In this figure, the input image is 256×256 pixels in size.

files, the very first ones, which feed the input layer, correspond to the two speckle images from which displacement maps must be extracted. The output of each layer is a series of so-called feature maps apart from the output layer, which directly provides two displacement fields (one along each direction). In order to more easily visualize the successive simplifications of the architectures discussed in this paper, we propose to represent each CNN with a histogram, see a typical example in Figure 2 with the histogram corresponding to StrainNet-f. With this mode of representation, the height of each left-hand bar is proportional to the number of feature maps considered as input data. As an example, it is equal to 128 for the third layer named "l2-1" in Figure 2.

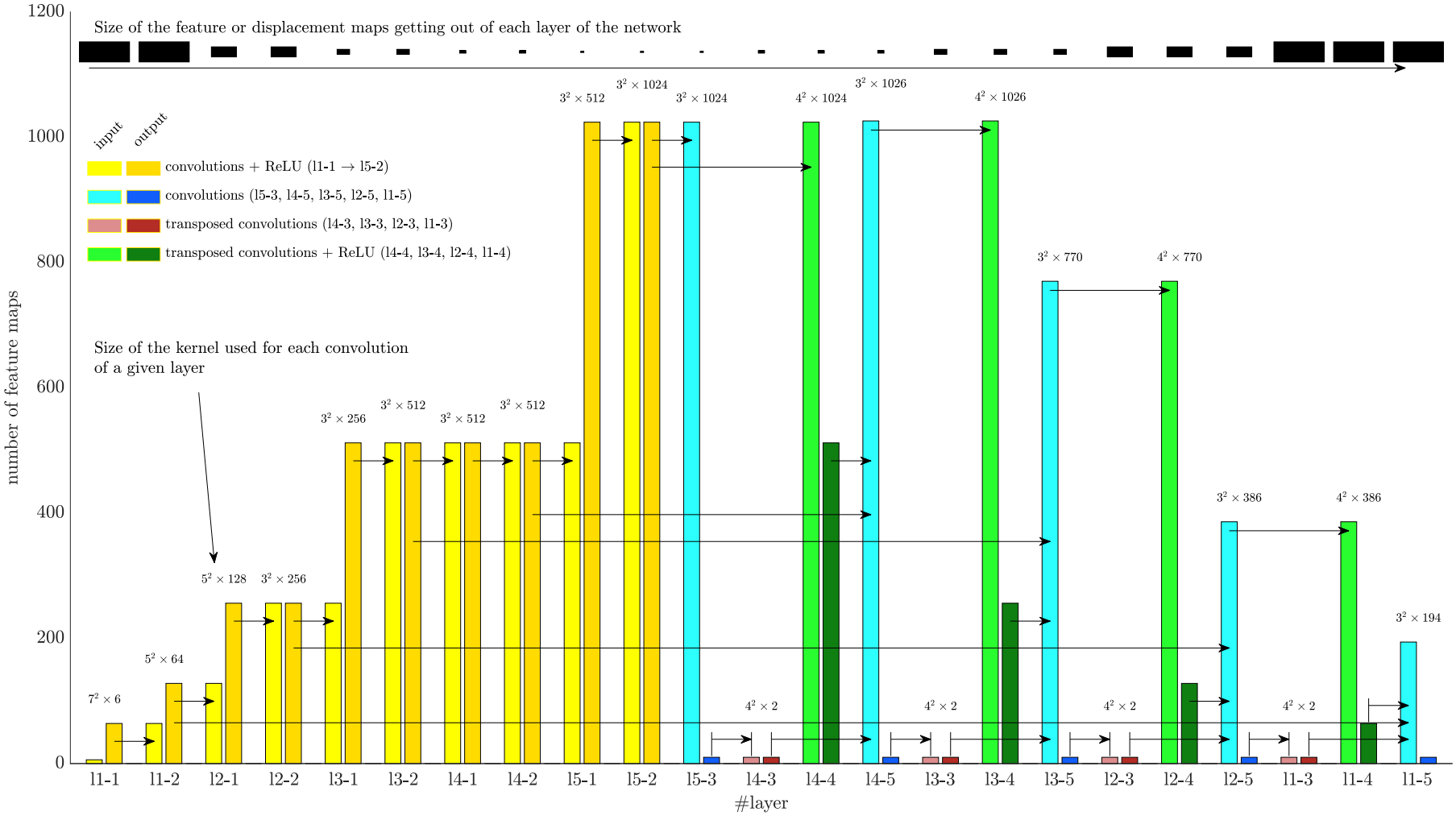


Figure 2: Schematic view of StrainNet-f discussed in [3]. Bright shade: number of input feature maps (or images for the first layer, equal here to 6 because there are two raw images encoded in RGB) entering each layer, dark shade: number of output feature maps (or displacement maps for the last layer) getting out of each layer. Layers of type 1 (in yellow): convolutions+leaky ReLU. Layers of type 2 (in blue): convolutions. Layers of type 3 (in red): transposed convolutions. Layers of type 4: (in green) transposed convolutions+leaky ReLU. The scale for the dark blue, light red and dark red bars is five times higher than the scale used for plotting the other bars. The link between the output and the input of the layers is illustrated by the horizontal black arrows plotted between the vertical bars.

Each layer is characterized by a certain number of convolutions applied to the input data. In Figure 2, the number of convolutions is proportional, for each layer represented by a set of two twin bars, to the height of the right-hand side bar. As an example, this number is equal to 64 for the first layer and to 256 for the third layer. The number of filters employed in the dark blue, light red and dark red bars being small (it is equal to 2), the scale for these three types of bars is five times higher than that used for plotting the other bars. The size of the kernels used for the convolutions is also reported in Figure 2. This size typically writes $a_1^2 \times a_2$, where a_1 is a small number defining the size of the kernel in the plane of each feature map, and a_2 the depth of this kernel. As an example, $a_1 = 7$ and $a_2 = 6$ for the first layer. $a_2 = 6$ is justified by the fact that the input of the CNN is a pair of two images encoded in Red, Green and Blue (RGB), thus leading to six input channels. It means that each filter processes both the reference and the deformed images at the same time. Since this layer has 64 filters, it provides 64 feature maps. For the third layer, $a_1 = 5$ and $a_2 = 128$ because 128 input feature maps form the input data in this case. Indeed, in common CNN architectures, the kernel depth is equal to the number of input feature maps. Since 256 different convolutions are performed in this third layer, 256 different kernels must be defined. The size of each kernel being equal to $5^2 \times 128$, $5^2 \times 128 \times 256 = 819,200$ coefficients must be set by learning for this third layer only. Furthermore, a bias term, also to be learnt, is deliberately added to the output of each convolution. A similar calculation can be made for the other 22 layers, which means that the total number of parameters to be learned for the whole CNN is huge: about 38 millions for the present CNN. Another point is the fact that the size of the feature maps progressively changes in the CNN. This is illustrated by the gray rectangles located at the top of Figure 2. It can be seen that this size first decreases from one layer to another. This size is indeed divided by 4 (2 along each direction) by applying a stride of two pixels along both directions. This downsampling of the data is followed by successive upsamplings on the right-hand side of the network, so the size of the feature maps progressively increases

to reach in this case the dimension of the two input speckle images (the reference image and the current one). Note this last size can also be smaller, as will be seen in the second example discussed below. Upsampling is obtained here by applying a transposed convolution to the data. This procedure can also be referred to as a deconvolution. Note that the number of feature maps also increases with downsampling, and then decreases with upsampling which occurs on the right-hand side of the network. Successive layers characterized by feature maps having the same size form a level. It can be checked in Figure 2 that this CNN has 5 different levels, which also govern the last two characters of the name of each layer. For instance, the name of the third layer is "l2-1", which means that this is the first layer of the second level. The size of the maps provided at the end of the procedure is the same as the one of the two speckle images used as input data. Typical feature maps are given in a particular exemple later on in this paper. It will be shown that these feature maps of various dimensions (the deeper the layer, the lower these dimensions) progressively look like the final displacement maps extracted from the initial RGB images.

These 23 layers are of four different types, characterized each by a different color. Two different shades are used for each color: a bright one for the left-hand side bars whose height is proportional to the number of input data files for each layer, and a dark one for the right-hand side bar whose height is proportional to the number of convolutions, thus to the number of feature maps provided by any layer (or the final displacement map for the last layer). The first ten layers (colored in yellow) belong to the first type, which consists in a convolution as explained above, followed each by a ReLU which provides a non-linear response.

Three different types of layers form the second part of the network. The layers of the second type (in blue) provide only two feature maps by convolution. These maps progressively become the two sought displacement maps at the very end of the network. The layers of the third type (in red) perform upscaling by transposed convolution, so it can be checked at the top of Figure 2 that the size after red layers is always greater than before (beware, the

corresponding vertical red bars are tiny because they contain a small number of layers). The last type of layer (in green) progressively upscales by transposed convolution + ReLU the feature maps given by the different layers of the left-hand side part of the network. It is worth noting that layers of the second type (in blue), apart from the first one, mix data coming from the three other types of layers.

2.1.3 Architecture of StrainNet-h

Finally, Figure 3 represents the architecture of the second network considered in this study, namely StrainNet-h described in [3]. The number of layers is the same as that of StrainNet-f but the size of the feature maps is smaller (because of the additional down-sampling). According to [3], this leads to a calculation time which is about ten times lower than that obtained with StrainNet-f. The price to pay is that the size of the final displacement maps rendered by the network is four times smaller than the size of the input images, which means that interpolation must be performed along both directions to reach the same size as that of the input images.

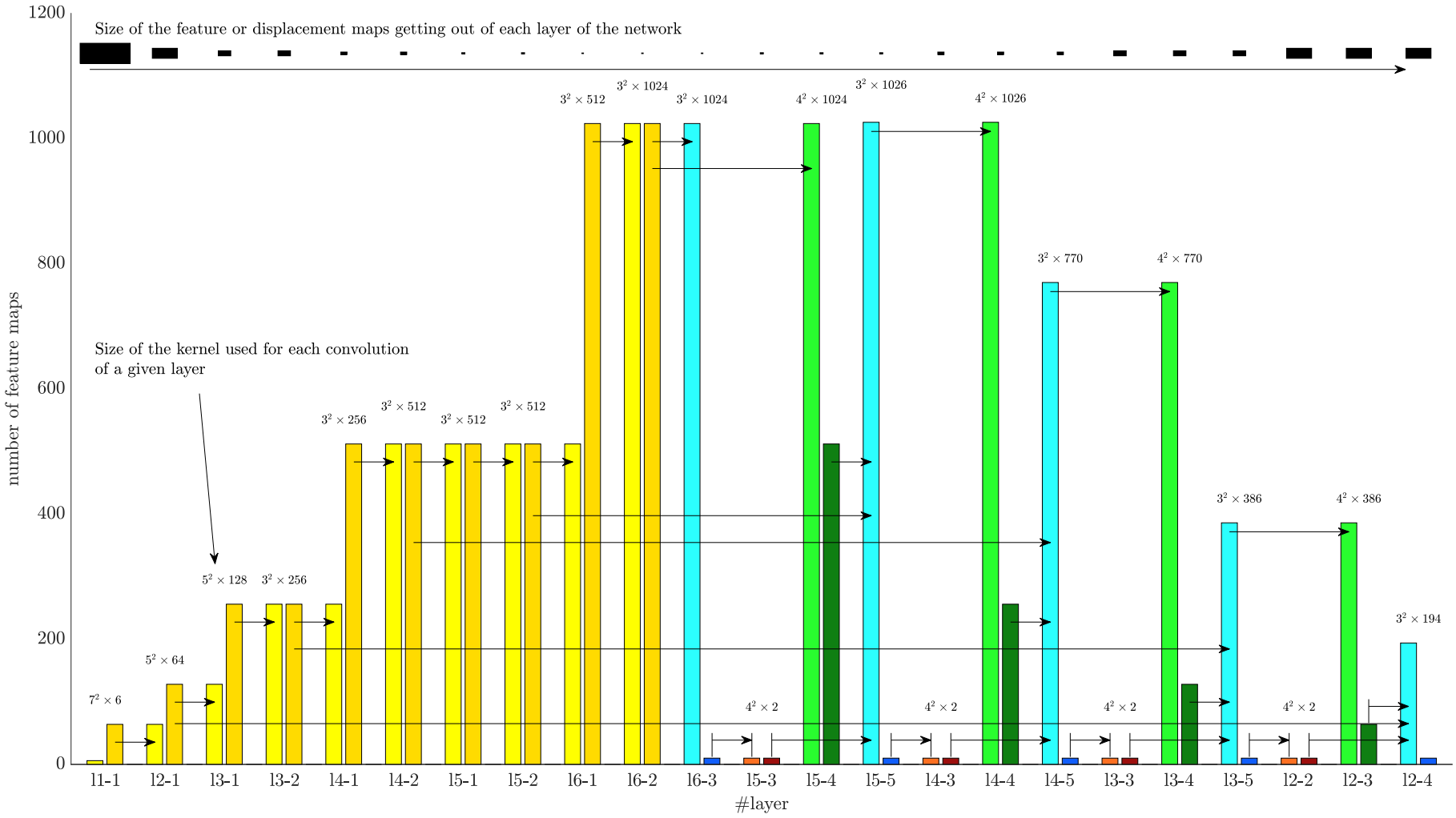


Figure 3: Schematic view of StrainNet-h discussed in [3]. Layers of type 1 (in yellow): convolutions+leaky ReLU, layers of type 2 (in blue): convolutions, layers of type 3 (in red): transposed convolutions, layers of type 4: (in green) transposed convolutions+leaky ReLU. Bright shade: number of input maps for each layer, dark shade: number of output maps. Suffix "-h" is justified by the fact that the size of the maps is half the size of the speckle images along their two directions. The scale for the dark blue, light red and dark red bars is five times higher than the scale used for plotting the other bars.

2.2 Dataset and training strategy

2.2.1 Datasets

Two datasets were used to train the CNNs presented above. The first one, called Speckle Dataset 1.0, was obtain first by rendering 363 synthetic reference speckle images with the speckle generator called BSpekleRender introduced in Ref. [10], in which the settings given in [3] to define for instance the number of speckles and their size were used. The images generated for this dataset were 256×256 pixels in size. A typical one is shown in Figure 4.

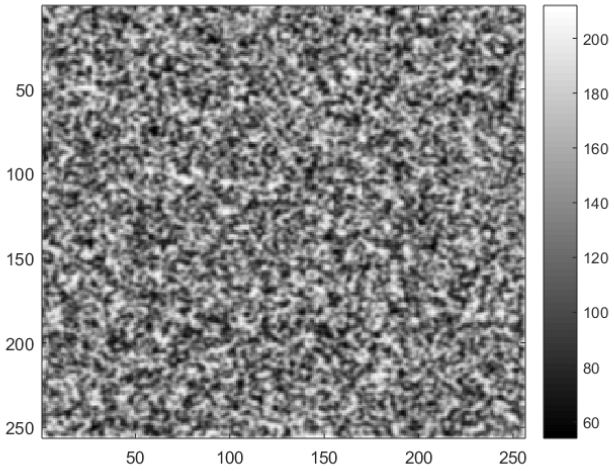


Figure 4: Typical synthetic speckle pattern rendered by BSpekleRender [10].

This speckle generator can also render deformed versions of these images through displacement fields defined by closed-form expressions like that used in Section 2.3 below. The advantage is that the deformed images are free from any interpolation bias. The drawback is that the present version of this generator is quite slow because of the Monte Carlo integration performed in the procedure. It means that with the present version developed with Matlab, only a limited number of pairs of reference/deformed images can reasonably be obtained. This is not suitable for the need to generate the several thousands of images which are necessary to train a CNN. The deformed images were therefore obtained by interpolation of the gray

levels of the reference image. With Speckle Dataset 1.0, each of the 363 reference images were deformed with 101 different displacement fields. These displacement field were defined by meshing the reference images with squares, which are all 8×8 pixels in size. The four corners of these element are nodes, which are subjected each to a horizontal and a vertical displacement. The displacement within each element is defined by bilinear interpolation of the nodal displacement. The amplitude of the displacement prescribed to the nodes was randomly drawn following a uniform distribution lying between -1 and 1 pixel. The amplitude of this random nodal displacement is 1 pixel since we focus here on the subpixel displacement, which is much more challenging to retrieve than integer displacements (in pixel). Displacements greater than one pixel can also be considered by the CNNs presented here, for instance by estimating the integer part of the displacement by cross-correlation, or by some rough DIC or optical flow program, and then determining the subpixel part with a CNN. This method is used in the last example given at the end of the present paper, where displacements greater than one pixel have to be measured. Another option, which will be considered in further studies, is to include reference displacement fields with an amplitude greater than 1 pixel in the dataset used to train the CNN.

The size of the elements constituting the mesh influencing the quality of the results, in particular in terms of spatial resolution, a second dataset named Dataset 2.0 was also generated, this time by considering the same 363 images meshed with elements of size $n \times n$ pixels, with $n \in 4, 8, 16, 32, 64, 128$ pixels. Bicubic interpolation was also used instead of a bilinear one to define the displacement within each element. Each reference image was deformed 10 times, which eventually gives $363 \times 6 \times 10 = 21,780$ different deformed images. Noise was finally added to the deformed images. This noise was heteroscedastic to faithfully mimic the noise affecting real camera sensors, with a variance v being an affine function of the brightness s in the image [11], thus $v = a \times s + b$, with $a = 0.0342$ and $b = 0.2679$, these values already used in previous studies ([3] for instance) being representative of a real sensor

noise.

2.2.2 Training strategy

Both the CNNs discussed in Ref.[3], namely StrainNet-f and -h, were first trained by using Speckle Dataset 1.0. The process was initialized with the parameters of FlowNet-S [5, 6]. The parameters obtained at the end of this first training were then considered as initial values for a second training performed with the images of Speckle Dataset 2.0. When simplifying the architecture of the CNNs by removing levels and filters, the obtained CNNs were trained with the images of Speckle Dataset 2.0. The coefficients of the layers which were kept were fine-tuned from one version to another. In other words, the coefficients of a given version were used as initial values of the coefficients of the following simplified version when launching the training of the latter.

2.3 STAR displacement field

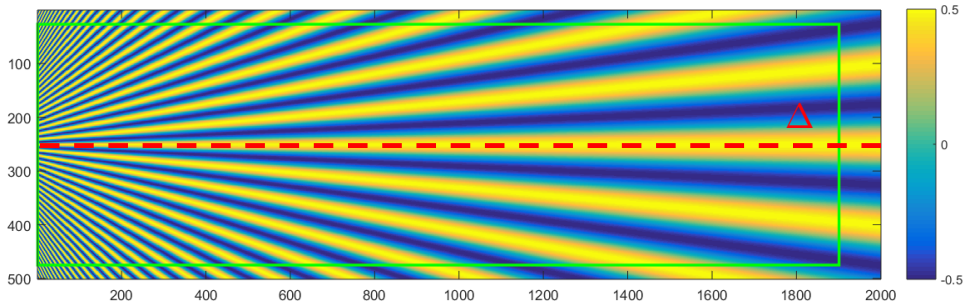
As in [3] and in other recent papers dealing with DIC [12, 13, 14] such as the recent one dedicated to the DIC Challenge 2.0 [15], we considered here as a reference the star-shaped vertical displacement field proposed in [16] to deform an artificial speckle image. It is such that the horizontal displacement is null whereas the vertical one is a mere sine function with a period linearly varying from the left to the right of the image. The mid-height also forms a horizontal axis of symmetry Δ along which the displacement is constant and equal to its maximum value u_{max}. Thus

$$u_2^{ref}(x,y) = 0.5\cos\left(\frac{2\pi}{p_{wave}(x)}(y - H/2)\right), \tag{1}$$

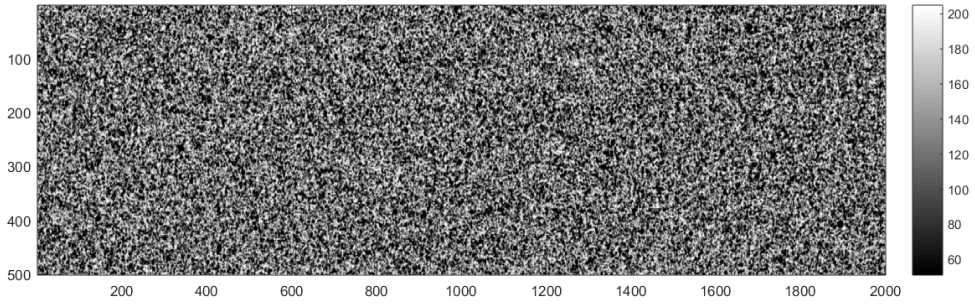
where H is the height of the image. The period p_{wave} is modeled by:

$$p_{wave} = p_{wave}^{mini} + \frac{p_{wave}^{maxi} - p_{wave}^{mini}}{L}(x-1)/(L-1), \tag{2}$$

where L is the length of the image. p_{wave}^{mini} and p_{wave}^{maxi} are the minimum and maximum values of the period of the sinusoidal displacement, respectively. x is the abscissa measured in pixel, thus x lies between 1 and L. The different parameters chosen here are $L = 2000$ pixels, H=501 pixels, p_{wave}^{mini}=10 pixels and p_{wave}^{maxi}=150 pixels. This reference vertical displacement field is depicted in Figure 5-a.



(a) STAR displacement field



(b) Typical artificial pattern

Figure 5: (a): STAR displacement field used as a reference. (b): Typical artificial speckle pattern obtained with BSpekleRender [10]

The benefit of using this type of reference displacement field is that one can assess the progressive attenuation of the displacement retrieved by the measuring technique under study. Indeed, it is for instance well known that DIC behaves like a low-pass filter [17], so the amplitude of high-frequency displacements returned by DIC is generally lower than the real one. A similar behavior is observed in the maps returned by the CNNs developed in [3], thus leading to also use this type of displacement field in the present study. We chose here the amplitude of the sine wave to be equal to $u_{max} = 0.5$ pixel. This setting is useful for DIC

since the interpolation bias is null for this displacement whatever the nature of the interpolant, which enables us to investigate other sources of errors. Finally, we will see below that plotting the cross-section along the midline Δ of the vertical displacement retrieved by the different techniques under study provides meaningful information on their metrological performance, in particular the attenuation of the displacement for the highest frequencies.

This STAR displacement field was used to generate a pair of reference and deformed artificial speckle images. An example of artificial speckle pattern is shown in Figure 5-b. Both this image and its deformed version were generated by the same speckle generator as the one used for rendering the reference images of the dataset discussed in Section 2.2.1 above. However, contrary to images of the dataset used to train the CNNs, the image pattern deformed through the STAR displacement field is free of any interpolation bias since no interpolation is used to obtain it [10]. Finally, noise was added to both the reference and the deformed STAR images when image noise propagation was studied. The same type of heteroscedastic noise as that descibed in Section 2.2.1 was used.

These reference and deformed images were then used to assess the metrological performance of the CNNs under study by comparing the reference STAR displacement with the displacement retrieved by the CNN from this pair of images. Image noise propagation was also observed and quantified with such a map by merely comparing displacement fields retrieved from noisy and noiseless images.

3 Toward lightweight CNNs

The architecture of StrainNet-f and -h described above was directly inspired from FlowNet-S developed for solving optical flow problems [5, 6]. In [3], moving on from the -f to the -h version of the CNN was a first attempt to adapt the architecture to the problem at hand. The aim here is to go much further by tackling in turn two issues:

- further adapting the architectures of StrainNet-h and -f to estimate displacements fea-

turing high spatial frequencies such as those of the STAR displacement field described above;

- simplifying as much as possible the network in order to reduce the calculation time and the required memory size while keeping as much as possible the metrological performance obtained with the original versions.

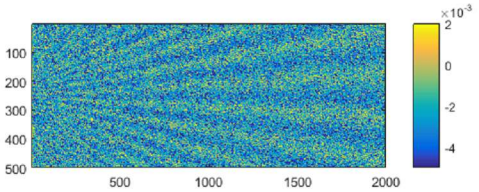
Tackling the first issue consists in studying the number of pyramidal levels and the impact of their number on the results. The second one, discussed later on in the paper, consists in reducing the number of filters in each layer of the network to be simplified. Note that the solution proposed for the first problem consists in reducing the number of layers, which will also positively contribute to resolving the second problem. Another remark is that developing CNNs widely relies on heuristics, by using a trial-and-error methodology, and this is precisely what we did here. We detail below the main stages of our approach, which has eventually led to a simplified CNN which responds to the two above requirements.

3.1 First step: removing the deepest levels

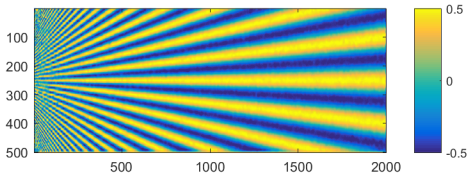
In this section we mainly examine the contribution of the deepest layers to the solution of the problem at hand. This is made by progressively removing the deepest levels and retraining the network. Both the StrainNet-h and -f versions are considered here.

3.1.1 Qualitative illustration of the contribution of the deepest levels

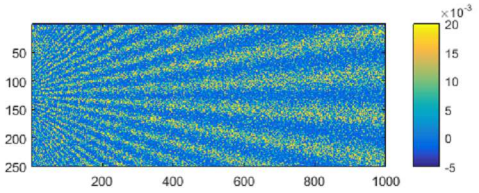
The first step is to examine, from a qualitative point of view, the contribution of the different levels on the displacement maps. Typical feature maps are shown in the first column of Figure 6 in order to help the reader figure out what these feature maps exactly are. They represent typical outputs of layers from different levels located on the left-hand side of StrainNet-f. The second column gathers typical displacement maps which are the outputs of different deconvolutions performed on the right-hand side of StrainNet-f. For both columns, increasingly deeper levels are considered when going from the top to the bottom of the figure.



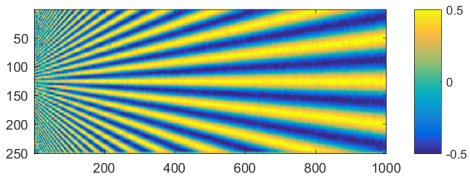
(a) typical feature map of level 1



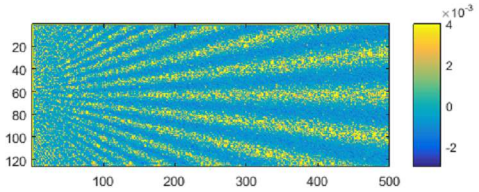
(b) typical feature map of level 1



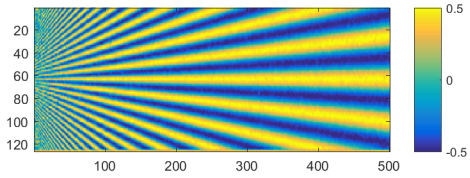
(c) typical feature map of level 2



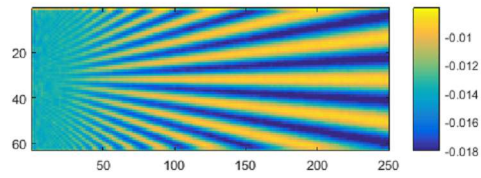
(d) typical feature map of level 2



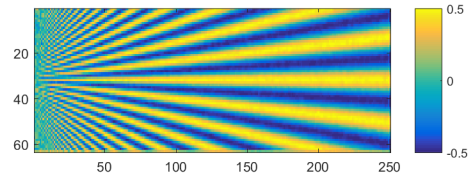
(e) typical feature map of level 3



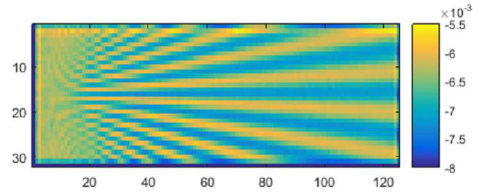
(f) typical feature map of level 3



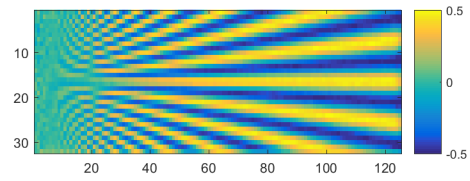
(g) typical feature map of level 4



(h) typical feature map of level 4



(i) typical feature map of level 5



(j) typical feature map of level 5

Column 1: left-hand side of StrainNet-f

Column 2: right-hand side of StrainNet-f

Figure 6: Results obtained by processing the STAR image. Left: typical feature maps obtained as outputs of filters of the left-hand side of StrainNet-f (see Figure 1). The filters belong to increasingly deeper levels. Right: typical displacement field obtained as outputs of filters of the right-hand side of StrainNet-f, in pixels. The filters belong to increasingly deeper levels. All dimensions along the two axes are in pixels.

Five remarks can be drawn from these maps:

1. The apparent size of the feature maps and the displacement maps is the same in all the sub-figures but the graduation along the axes shows that the real size of these maps diminishes when going to the deepest levels. This is due to the successive strides (all equal to 2 pixels) applied to the maps when going to the deepest levels.

2. In the left-hand side column, the feature maps have the same aspect as the final displacement field. It is however worth noting that their values have different amplitude and they do not all fluctuate around zero. Choosing other feature maps from the same levels would lead to other values. It can be observed that their amplitude tends to decrease when going to the deepest levels.

3. The deeper the levels, the lower the amplitude of the noise affecting the feature maps, This is due to the fact that the filters, which are successively applied when going to the deepest levels, progressively reduce the noise level.

4. The left-hand side of the displacement maps returned by the different deconvolutions shows that the highest frequencies are progressively lost when considering increasingly deeper levels. Indeed, since the actual size of the maps progressively diminishes, it becomes progressively impossible to correctly sample the vertical sine displacement featuring the lowest periods. More precisely, the height of the STAR image is 501 pixels. This height is divided by two each time a stride is applied. It means that after 4 strides, the height is equal to $501/2^4 = 31.3125$ rounded up to the next integer value, namely 32 pixels. Since the period p of the vertical sine wave goes from 10 to 150 pixels when x goes from 1 to 2000 pixels [12], the link between p and x reads as follows in the initial coordinate system: $p = 10 + (150 - 10)/(2000 - 1) \times (x - 1)$. After 4 strides, this equation becomes $p = 10/2^4 + (150 - 10)/(2000 - 1) \times 2^4 \times (x - 1)$. In this new coordinate system, the abscissa x for which the Nyquist frequency is reached satisfies the following

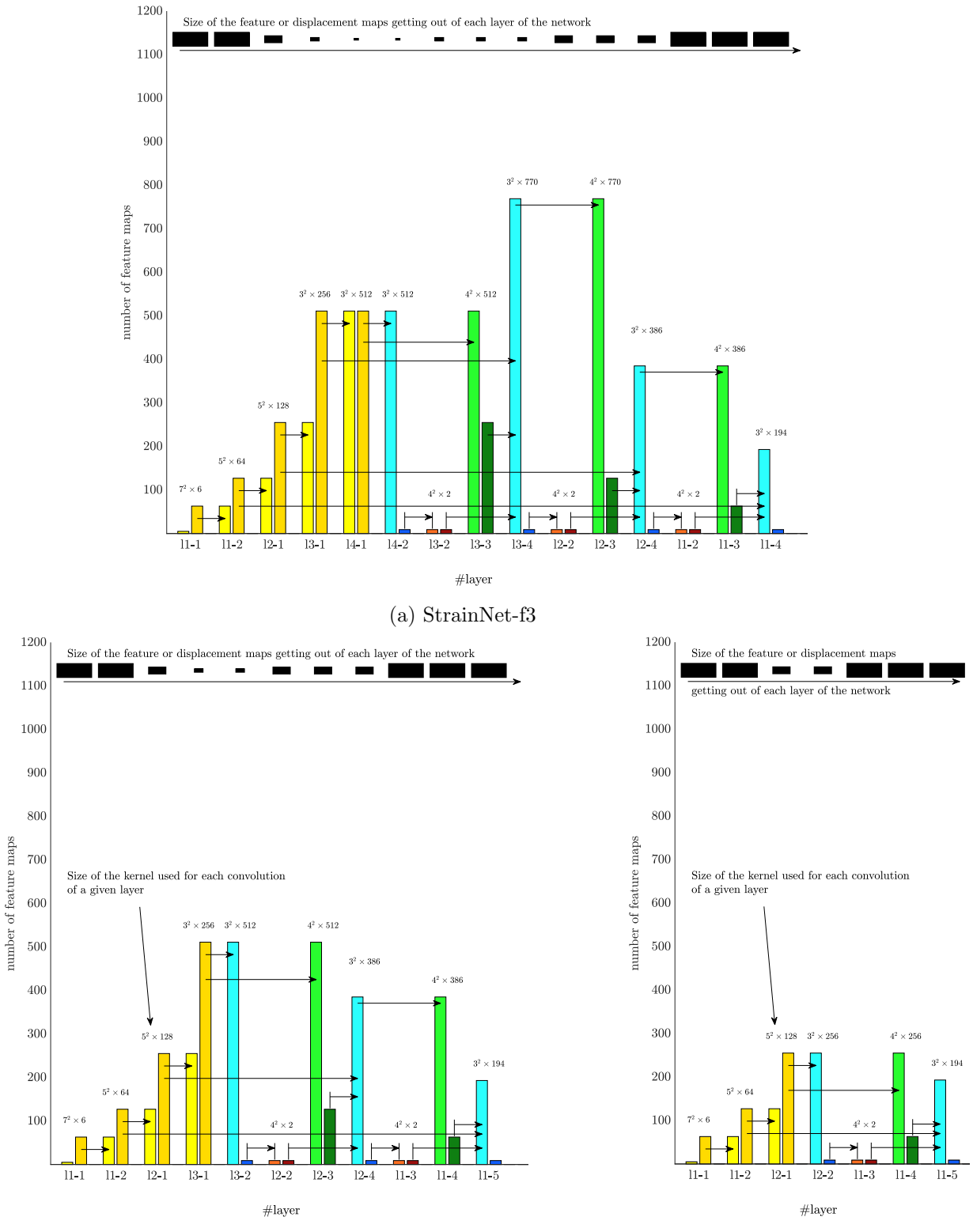
equation: $2 = 10/2^4 + (150-10)/(2000-1) \times (x-1)$, thus $x \simeq 20.6$ pixels. Indeed, this is approximately the coordinate for which the cutoff frequency is observed in Figure 6-j.

5. In the right-hand side column, all the feature maps have the same colorbar lying between -0.5 and 0.5 pixels, the last pair on the very right-hand side being the displacement maps provided by the CNN.

The main conclusion is that removing the deepest layers should mainly lead the CNN to focus more on high spatial frequencies, the lowest spatial frequencies being affected by this removal. This point is discussed further in the next section.

3.1.2 Effect of removing the deepest levels

Figures 7 and 8 show various architectures obtained by progressively reducing the number of levels in StrainNet-f and -h, respectively. These architectures are obtained by removing the deepest levels, which correspond to the central bars of the histograms in Figures 2 and 3.

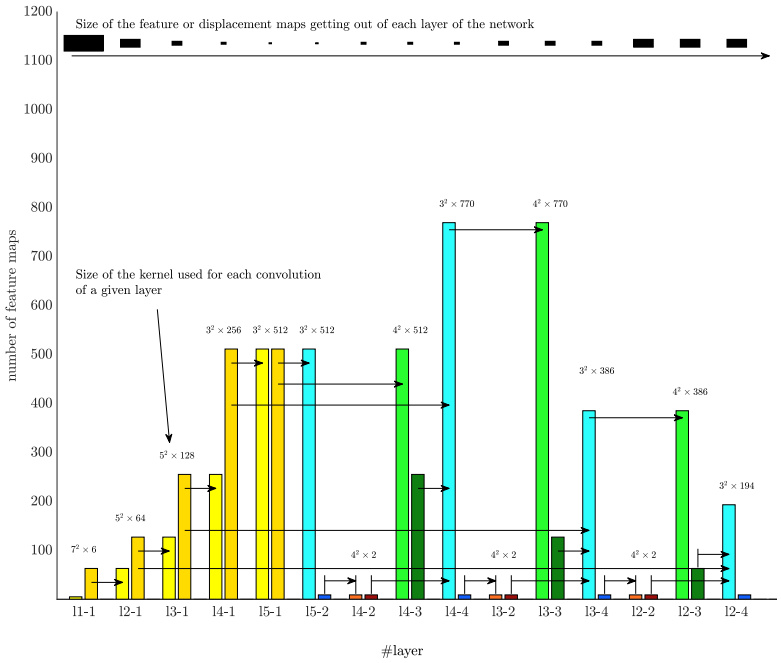


(a) StrainNet-f3

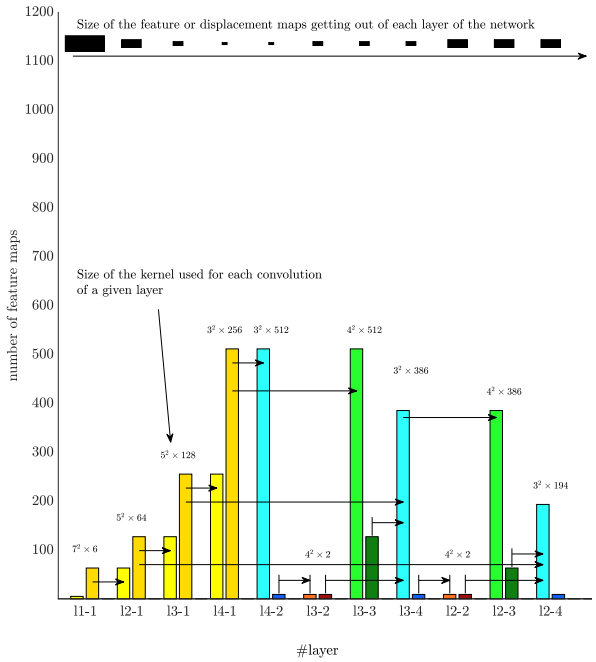
(b) StrainNet-f2

(c) StrainNet-f1

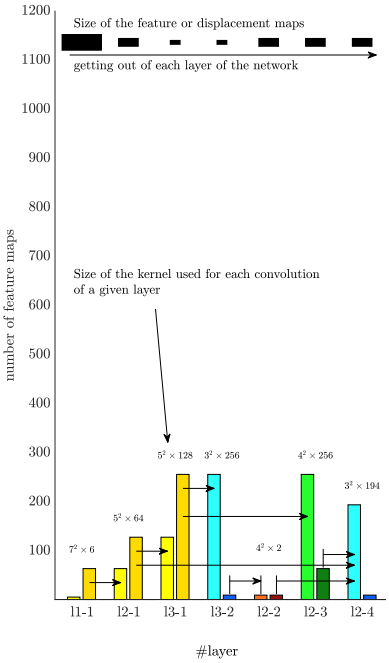
Figure 7: Various simplified versions of StrainNet-f4 obtained by progressively removing the central layers of the network. The scale for the dark blue, light red and dark red bars is five times higher than the scale used for plotting the other bars.



(a) StrainNet-h4



(b) StrainNet-h3



(c) StrainNet-h2

Figure 8: Various simplified versions of StrainNet-h5 obtained by progressively removing the central layers of the network. The scale for the dark blue, light red and dark red bars is five times higher than the scale used for plotting the other bars.

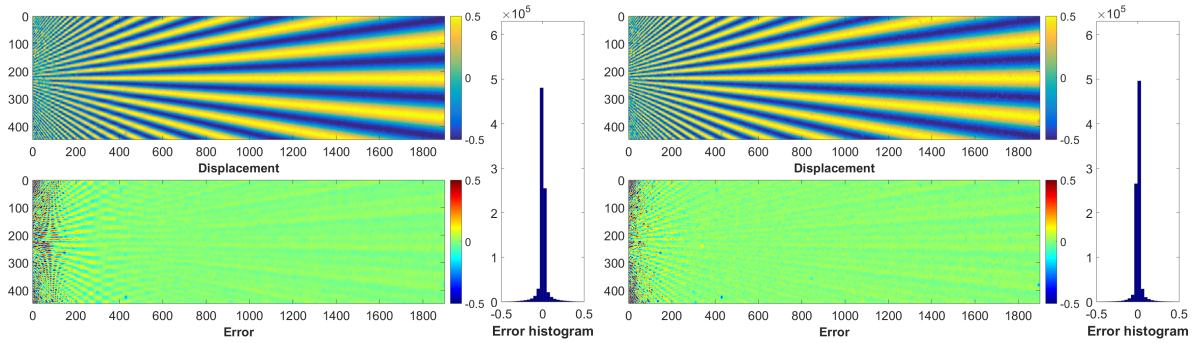
The suffix added to the name writes f-i or h-i, where i is the number of downsampling

stages in each CNN, thus the architecture becomes more and more simple as i decreases. With this rule, the original versions of StrainNet-f and StrainNet-h write StrainNet-f4 and StrainNet-h5 since they exhibit 4 and 5 downsampling stages, respectively. For the sake of consistency between the names of the initial and simplified versions, we adopt this notation in the remainder of the document. The same scale is kept for all the sub-figures in order to better visually observe the decreasing trend of the number of layers and feature maps when progressively modifying the architecture of the network.

A key point is to know how the reduction of the number of levels impacts the metrological performance of the CNN. As explained above, we rely for this on the STAR displacement field, which deforms the artificial speckle image shown in Figure 5-b. Noiseless pairs of reference and deformed images were processed in turn by the different versions of the CNNs. Figures 9 and 10 a to d show the displacement map retrieved by these different versions, as well as the error map obtained by subtracting the maps retrieved by each CNN and the reference displacement map. As expected, this error increases for all the versions when moving to the highest spatial frequencies, thus to the left-hand side of the maps.

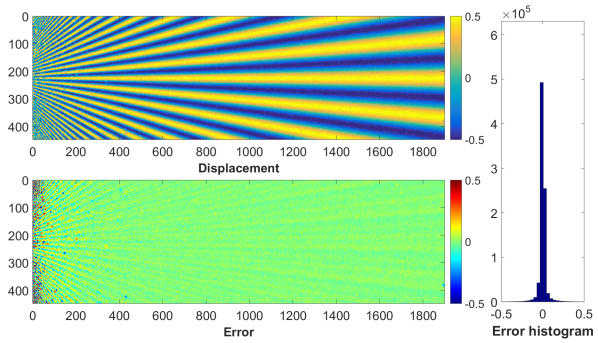
Figures 9-e and 10-e show cross-sections of the displacement and error maps obtained for all these versions of CNNs. The error maps represented at the bottom of each figure are calculated columnize (over the zone bordered by the green rectangle in Figure 5-a to avoid edge effects). This gives a more global estimation compared to the one given by the cross-section along Δ only. These curves clearly show the improvement brought about by the successive modifications on the possibility to reliably detect the details with the highest spatial frequencies. This is due to the fact that the deepest levels provide feature maps with the lowest size. Indeed, high numbers of downsampling stages lead to layers which provide information on the lowest spatial frequencies of the image, thus removing them leads the network to mainly focus on the determination of the highest spatial frequencies.

An interesting result is that this error tends first to decrease when reducing the number

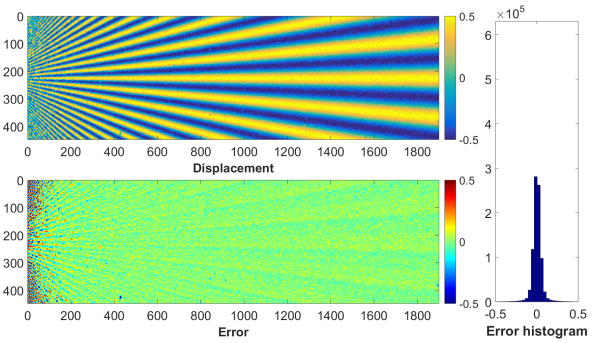


(a) StrainNet-f4, MAE = 0.0266

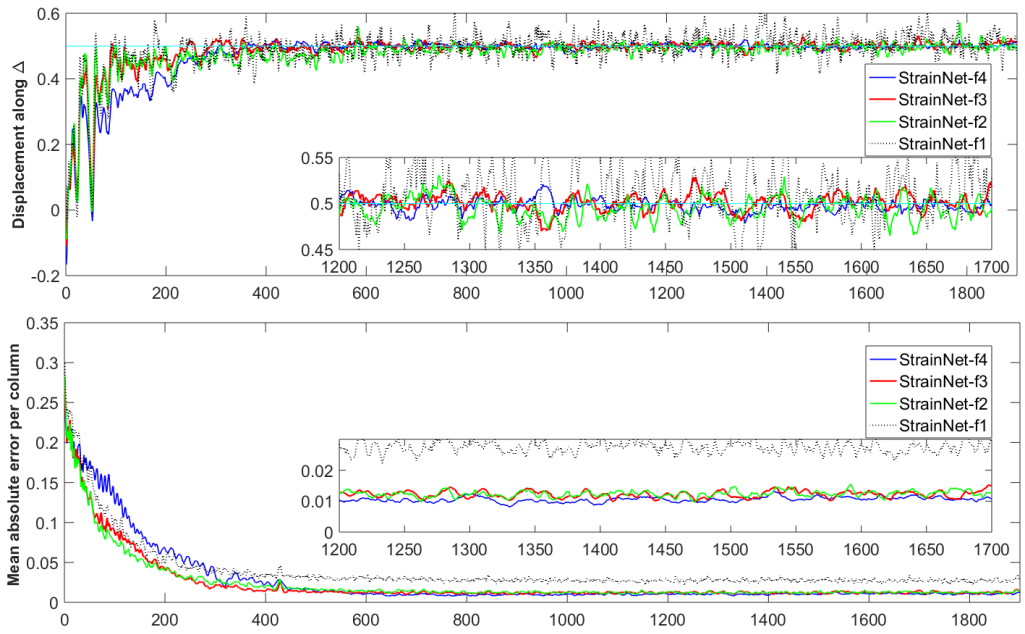
(b) StrainNet-f3, MAE = 0.0230



(c) StrainNet-f2, MAE = 0.0233

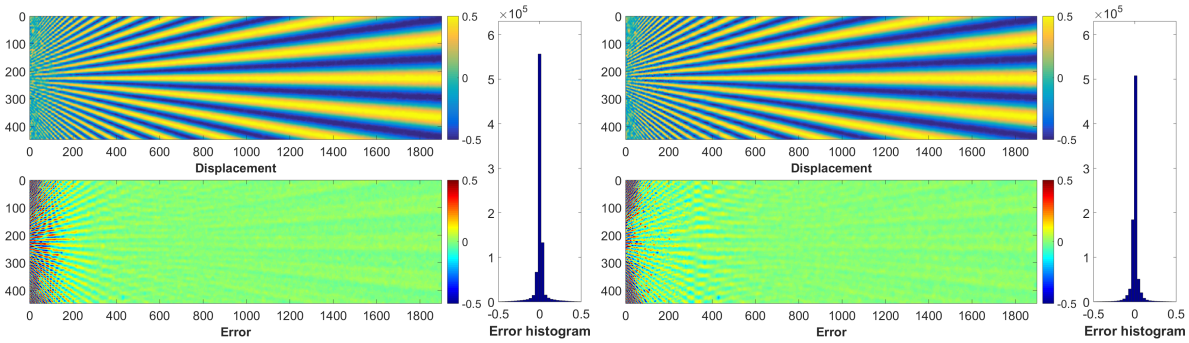


(d) StrainNet-f1, MAE = 0.0391



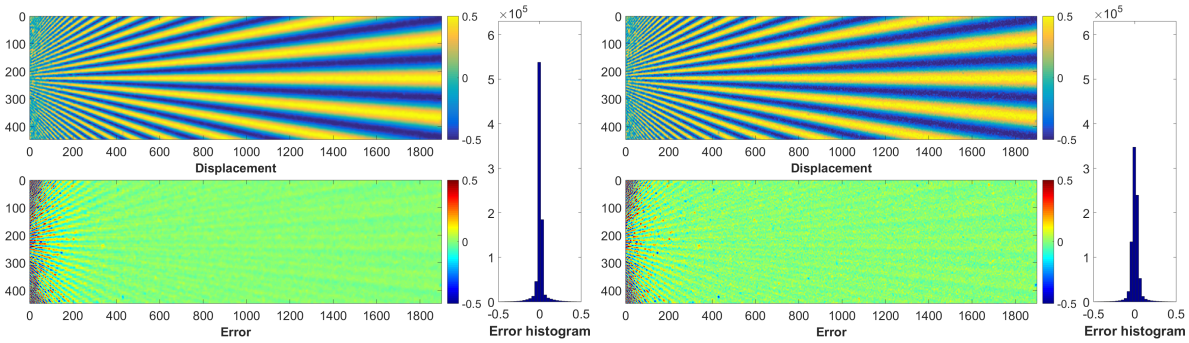
(e) Displacement along Δ and mean absolute error per column. All dimensions are in pixels.

Figure 9: Results obtained with noiseless STAR images StrainNet-f.



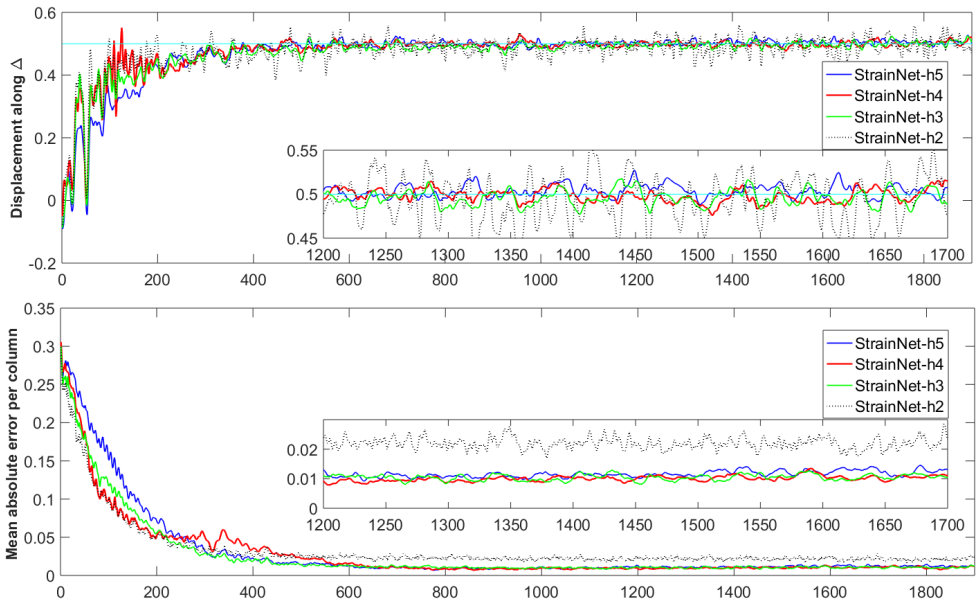
(a) StrainNet-h5, MAE = 0.0305

(b) StrainNet-h4, MAE = 0.0276



(c) StrainNet-h3, MAE = 0.0263

(d) StrainNet-h2, MAE = 0.0338



(e) Displacement along Δ and mean absolute error per column. All dimensions are in pixels.

Figure 10: Results obtained with noiseless STAR images StrainNet-h.

of levels of the architecture, except at the end of the simplification procedure, thus when going from the -f2 to -f1 and from the -h3 to -h2 versions. This may be explained by the fact that when using -f1 and -h2, higher spatial frequencies can be detected. Compared to the other simplified versions, this causes a lower smoothing of the displacement maps and a higher noise level to appear. The visual observation is confirmed by the global error in the displacement map, which is estimated by the mean absolute error (MAE) calculated over the field of interest. This quantity is defined by

$$\mathrm{MAE} = \frac{1}{NM} \sum_{i=1}^{N} \sum_{j=1}^{M} |u_e(i,j) - u_g(i,j)| \tag{3}$$

where u_e and u_g are the estimated displacement and ground truth, respectively, and $|\cdot|$ denotes the absolute value of "$\cdot$". M and N are here the dimensions of the green rectangle in Figure 5 over which this quantity is calculated. This global indicator is reported in the fourth line of Tables 1 and 2. This MAE is the smallest for StrainNet-f3 (the result for StrainNet-f2 is very close) and StrainNet-h3.

Tables 1 and 2 also give the number of layers and parameters to be learned for each modification of the -f and -h versions of the CNN. It can be seen that these quantities, which reflect the complexity of the CNN, dramatically decrease when removing levels from the architecture. This directly impacts the number of operations to be performed to obtain displacement at a given pixel. This quantity is given by the number of Multiplication-ACcumulation operations (MAC), which are necessary to obtain the value of the displacement along x and y at each pixel. This quantity, named MAC/pixel, is given in the third line of both tables. It logically decreases when going toward the right in these tables, thus to simpler networks. The MAC/pixel number directly governs the calculation times needed to get a displacement map from the speckle images while the number of coefficients governs the memory size.

Table 1: Performance of the proposed full-resolution networks

Network	StrainNet-f4 (=initial version)	StrainNet-f3	StrainNet-f2	StrainNet-f1
N. parameters $\times 10^6$	38.68	8.69	3.69	1.31
N. layers	23	15	11	7
MAC/pixel	12.3E5	7.76E5	6.73E5	4.99E5
MAE (pixel)	0.0266	0.0230	0.0233	0.0391

Table 2: Performance of the proposed half-resolution networks

Network	StrainNet-h5	StrainNet-h4 (=initial version)	StrainNet-h3	StrainNet-h2
N. parameters $\times 10^6$	38.68	8.69	3.69	1.31
N. layers	23	15	11	7
MAC/pixel	3.22E5	2.08E5	1.82E5	1.40E5
MAE (pixel)	0.0305	0.0276	0.0263	0.0338

It is worth noting that in these tables, the MAE characterizing StrainNet-f2 is slightly lower than the MAE characterizing StrainNet-f4 despite a significantly lower number of parameters. The same remark holds for StrainNet-h3 and StrainNet-h5, respectively. Furthermore, StrainNet-f2 and -h3 improve the estimation of the highest spatial frequencies. It can be con-

cluded that the quality of the displacement maps returned by StrainNet-f2 and StrainNet-h3 is better for the STAR displacement field than that obtained with the corresponding original versions of the CNNs, which are respectively StrainNet-f4 and StrainNet-h5. Since the goal of the second step of this study is to simplify as much as possible the architecture of the CNN, we finally selected StrainNet-h3 as the best candidate for the next step.

3.2 Second step: reducing the number of filters per layer

This second step consists in reducing the number of filters for each layer of the candidate selected at the end of the first step, namely StrainNet-h3. The simplification strategy depends on the nature of the layer:

- Input layer: as gray level images are used as inputs in our case, 2 input channels (corresponding to the reference and deformed states) are used instead of the 6 input channels required for RGB images;
- Hidden layers: StrainNet-h3 was simplified by using a filter pruning approach. This consists in progressively reducing the number of filters in each layer and evaluating the obtained results after the retraining step. This is repeated until reaching a reasonable trade-off between accuracy and complexity. The goal here is to keep a metrological performance close to that of the original versions of StrainNet-h and f;
- Output layer: no simplification is applied because two displacement maps are needed.

Figure 11 shows the architecture of StrainNet-l, the network obtained at the end of this procedure. Suffix "-l" is justified by the fact that a light CNN is obtained at the end of the two-step simplification procedure. The main striking point is the global decrease, compared to the histrogram representing StrainNet-h3 in Figure 8-b, of the height of the bars representing the number of filters in the different layers. This directly impacts the number of parameters and MAC/pixel reported in Table 3 without really impairing the MAE values which were initially

obtained with StrainNet-f4 and -h5. Figure 12 shows the displacement map retrieved by DIC with 2M+1 = 21 pixels and second-order subset shape functions, StrainNet-f4, StrainNet-h5, and StrainNet-l. Figures 13 shows the cross-sections of the displacement field along the midline Δ as well as the error distributions obtained for all these three versions of CNNs. These figures clearly show the improvement brought about by StrainNet-l to detect the details corresponding to the highest spatial frequencies on the left. This is clearer when considering the cross section of the displacement fields along y at $x' = 100$ pixels from the left border of the green zone in Figure 5, see Figure 14a. The absolute value of the difference between measured reference displacements is shown in Figure 14b. This difference is, on average, the lowest with the profile given by StrainNet-l. This is confirmed by calculating the MAE along this cross-section in each case, see Figure 14c. In conclusion, none of the four procedures enables us to reliably reconstruct the reference sine displacement, but these results show that SrainNet-l provides the profile which is the closest to this reference profile.

Table 3 compares the main characteristics of the architecture of the initial CNN versions of StrainNet-f (StrainNet-f4) and -h (StrainNet-h5) and the ultimate simplified version of StrainNet-h5, namely StrainNet-l. A Jetson Xavier NX compact power-efficient embedded GPU, which can potentially be connected to a camera to perform real-time measurements, was used to compare the performance of these CNNs. This is a compact (70 mm×45 mm), low-cost and low-power consumption (10 W) device.

As a general remark, it may be quite surprising and puzzling to see that a huge simplification of the network in terms of coefficients to be learned (from 38.68 millions to 0.67 million, thus about 58 times lower) leads to results, which are globally equivalent to those given by CNNs without simplification. This is due to the fact that StrainNet-h5, which was inspired from FlowNet-S [5, 6, 3], is much more specialized than the latter. Indeed, the former only considers speckle images, which is not the case of the latter. Several other simplifications of CNNs obtained in a similar way are available in the literature, see Refs.[18, 19, 20] for

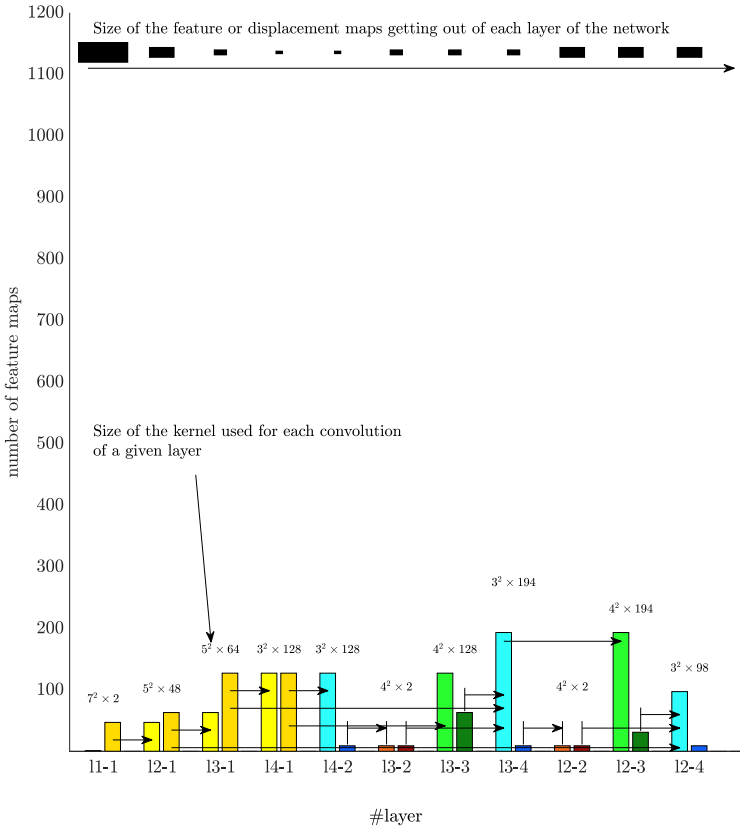


Figure 11: Final version of StrainNet-l after simplification of StrainNet-h3. The scale for the dark blue, light red and dark red bars is five times higher than the scale used for plotting the other bars.

instance. A possibility, not explored in the present paper, is to apply knowledge distillation, a procedure which consists in learning a small student model from a large teacher model [21]. This leads to a dramatic reduction of the number of parameters. CNN pruning can also be used to reduce the network complexity [19]. Pruning typically removes the CNN weights lower than a given threshold by forcing their value to zero. Pruned networks are known for their high sparsity (the portion of their weights having a null value), which reduces their model size as well as their execution time on dedicated hardware. Results shown in [19] feature a sparsity level of 75% can be reached while maintaining an acceptable loss of accuracy .

The impact of the low value of the MAC/pixel can be seen in the present case, with the greater value of the Points of Interest per second (PoI/s) of StrainNet-l compared to that of

StrainNet-h5. The PoI/s is a metric introduced in [1] to estimate the number of points per second at which a measurement is provided by the measuring system. This is a handy way to normalize the results obtained with different techniques and different frame sizes, and to fairly compare them. In the present case, the simplification of the original version of StrainNet-h5 to provide StrainNet-l leads to a factor of 3.3 between their PoI/s.

Finally, we estimated the metrological efficiency indicator of StrainNet-l and compared this quantity to its counterpart estimated with DIC and the original versions of the CNN. This indicator was introduced in [12] to compare different measurement techniques, and then used in other comparative studies, [13] for instance. The lower this indicator, the better the metrological performance. This quantity, denoted by α, is merely the product of the displacement resolution denoted by σ_u by the spatial resolution denoted by d. σ_u is equal to the standard deviation of the noise observed in the displacement maps obtained with noisy speckle images. This quantity was calculated by subtracting the displacement maps obtained with noisy and noiseless images. d is defined by the lowest period of a sinusoidal displacement for which the measuring system returns a displacement amplitude affected by a bias of 10% [22]. Is is obtained by plotting the cross-section along Δ of the STAR displacement field returned by the different techniques, and by searching the value of the period of the vertical sine displacement of the STAR image for which the amplitude is affected by a bias of 10%. Figures 15, 16 and 17 show a graphical construction leading to this quantity in each case. This value of α for the different techniques is reported in Table 4. It can be seen that StrainNet-l is characterized by a value of α, which is close to the one of the initial versions of StrainNet-f (StrainNet-f4) and StrainNet-h (StrainNet-h5). It is even slightly lower, thus better. This result means that the performance is globally equivalent between these different CNN versions. It is also worth noting that the value of α of StrainNet-l lies between that of DIC used with first-order subset shape functions and its counterpart obtained with second-order ones. This tends to prove that StrainNet-l can be considered to a viable alternative to

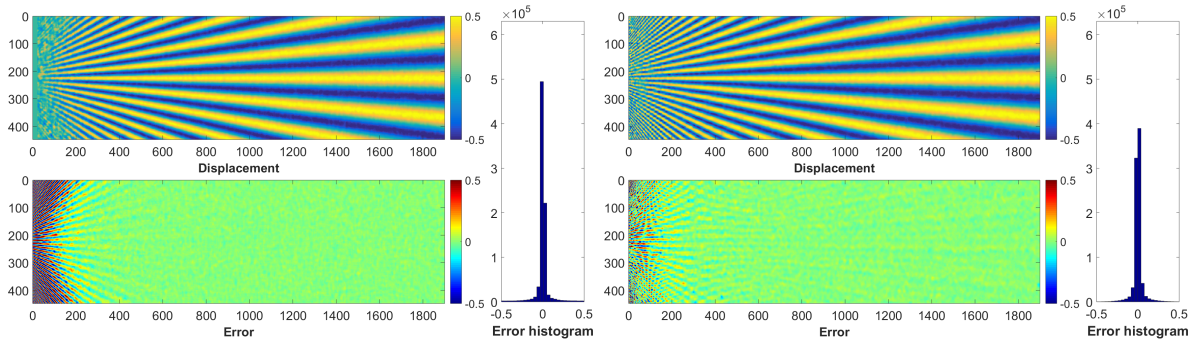
DIC.

3.3 Summary

In conclusion of this section, it can be said that simplifying the architecture of the CNN and dividing by 58 the number of coefficients governing the convolutive layers (thus the memory size) and accelerating the calculations by a factor of 3.3 does not negatively impact the metrological performance of the measuring system, the latter being estimated by the metrological efficiency indicator. The main consequence of removing the deepest levels of a CNN is that this CNN progressively correctly reconstructs ever smaller details since the spatial resolution decreases, but a negative impact is that the noise level progressively increases. The tradeoff between the two, reflected by the value of the metrological efficiency indicator, remains globally unchanged. Note also that it is easier to impair the spatial resolution (thus to increase this value) and consequently to improve the measurement resolution (thus to diminish this value) than the contrary. Indeed and for a given CNN architecture, merely spatially filtering the raw displacement map provided by this CNN by convolving this map with a Gaussian filter enables one to reach the first goal, while no simple operation is really available to reach the second one.

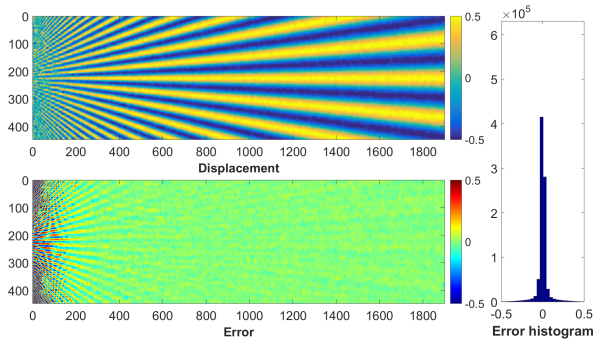
Table 3: Comparison between the characteristics of the initial CNNs and the final version after simplification

Network	StrainNet-f4 [3]	StrainNet-h5 [3]	StrainNet-l
MAC/pixel	12.3E+05	3.22E+05	0.48E+05
Numb. parameters $\times 10^6$	38.68	38.68	0.672
Numb. layers	23	23	11
MAE	0.0299	0.0333	0.0312
PoI/s (Jetson Xavier NX)	0.23E6	0.84E6	2.8E6

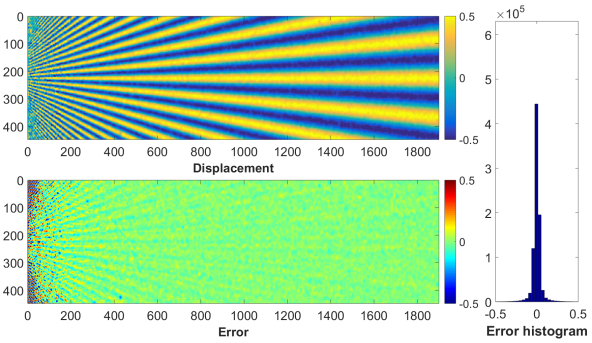


(a) DIC, 2M+1 = 21 pixels, MAE = 0.0394

(b) StrainNet-f, MAE = 0.0299



(c) StrainNet-h, MAE = 0.0333



(d) StrainNet-l, MAE = 0.0312

Figure 12: 2M+1 = 21 pixels, second-order subset shape functions (b) StrainNet-f4 (c) StrainNet-h5 and (d) StrainNet-l.

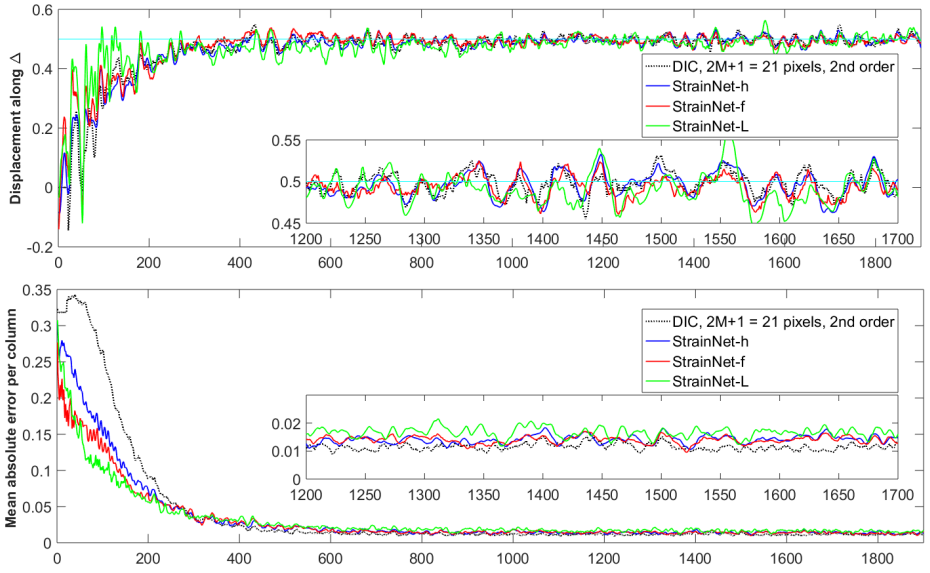
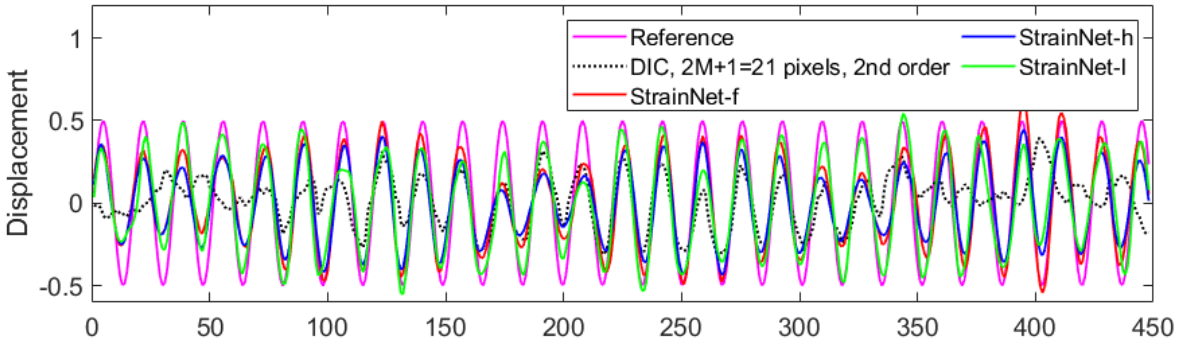
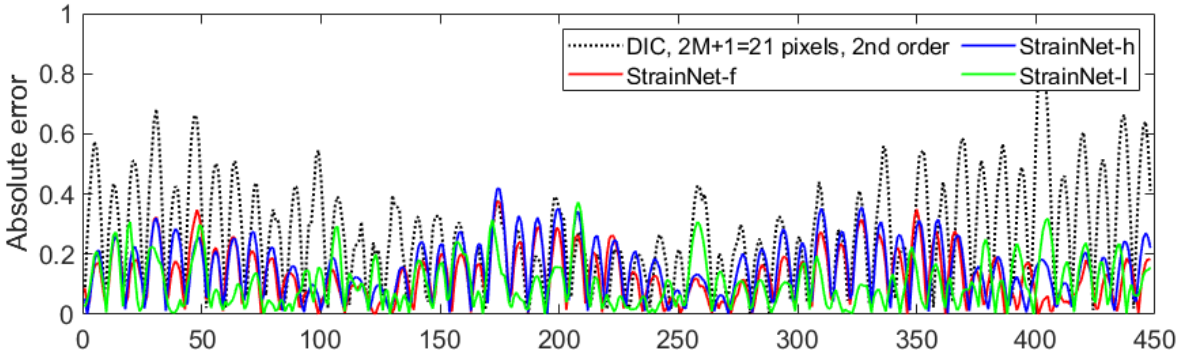


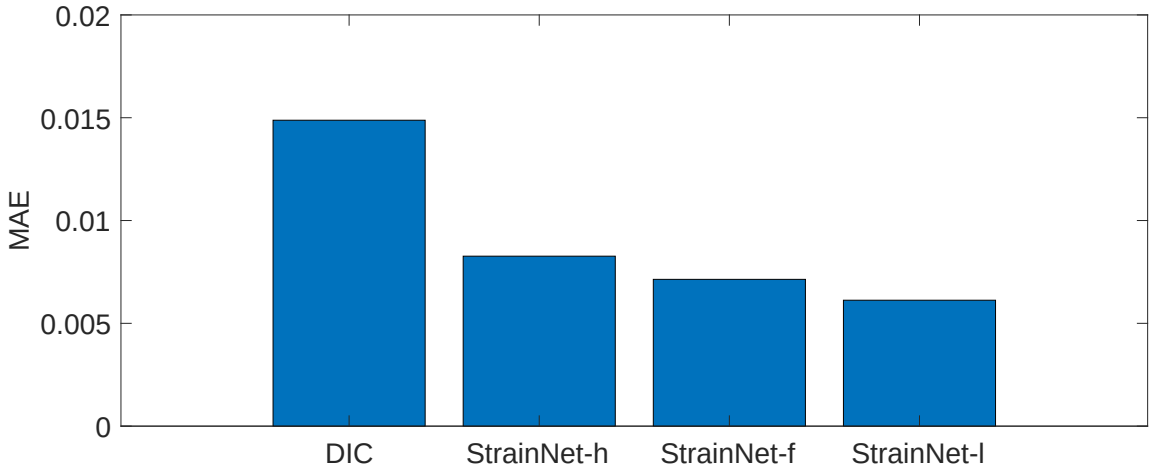
Figure 13: Results obtained with noisy STAR images: displacement along Δ and mean absolute error per column. All dimensions are in pixels.



(a) Cross-section

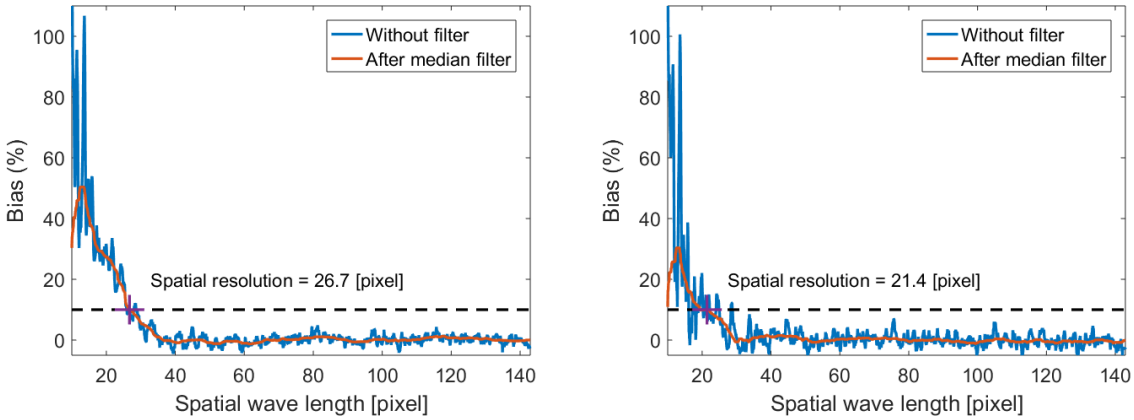


(b) Absolute error



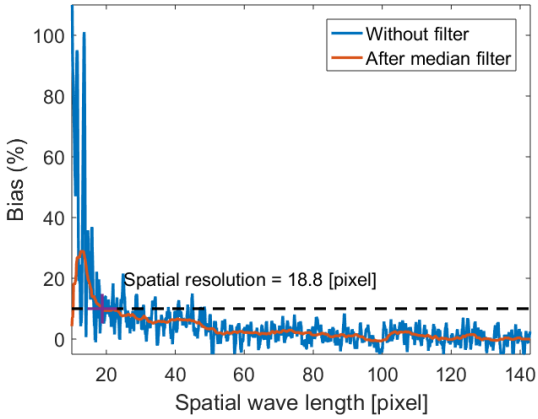
(c) Histogram of the MAE

Figure 14: Vertical cross-section of the displacement fields shown in Figure 12 , absolute error (absolute value of the difference between measurement and reference value), and histogram of the MAE estimated along this column for these four curves. The curves are plotted at $x' = 100$ pixels from the left border of the green box in Figure 5

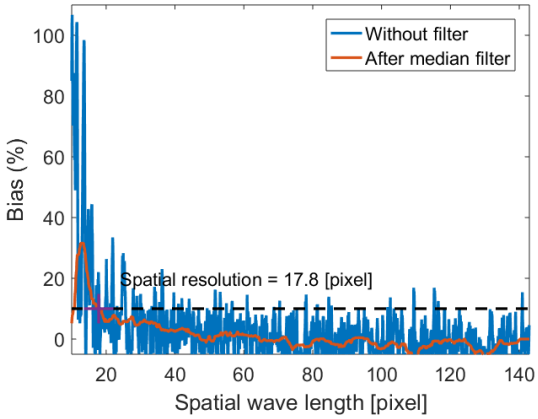


(a) StrainNet-f

(b) StrainNet-f3



(c) StrainNet-f2



(d) StrainNet-f1

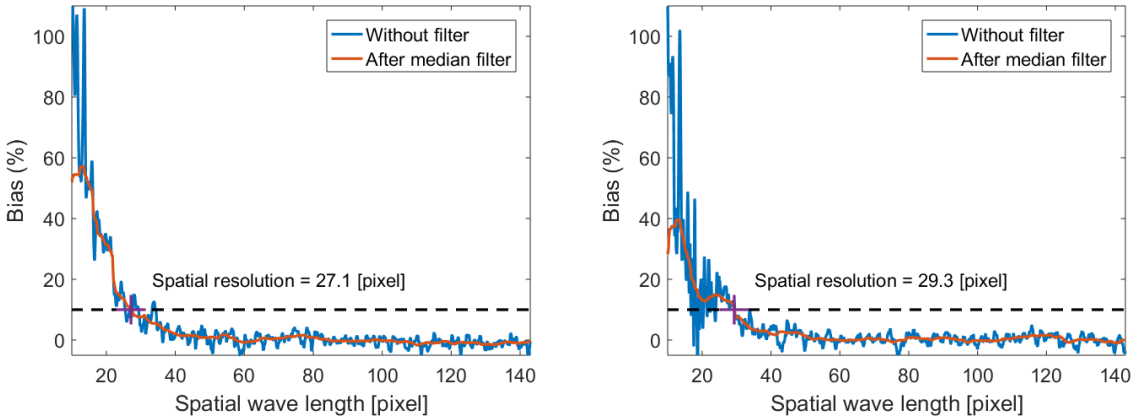
Figure 15: Seeking the spatial resolution of each version of StrainNet-f. The bias given here is a percentage of the displacement amplitude, which is equal to 0.5 pixel.

Table 4: Metrological efficiency indicator of DIC, initial CNNs and the final version after simplification

Algo.	DIC, 2M+1=11 (1st-order)	DIC, 2M+1=21 (1st-order)	DIC, 2M+1=21 (2nd-order)	StrainNet-f4 [3]	StrainNet-h5 [3]	StrainNet-l
α	0.62	0.59	0.39	0.49	0.46	0.45

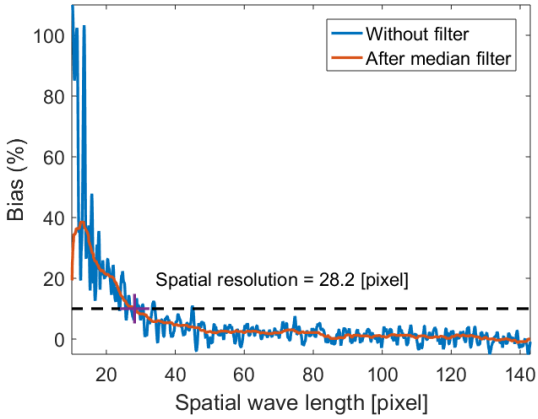
4 Application to real images

Before closing the paper, we consider real images obtained during a compression test performed along the vertical direction on the wood specimen shown in Figure 18-a. This test is presented in detail in Ref. [23]. The corresponding speckle images like those shown in

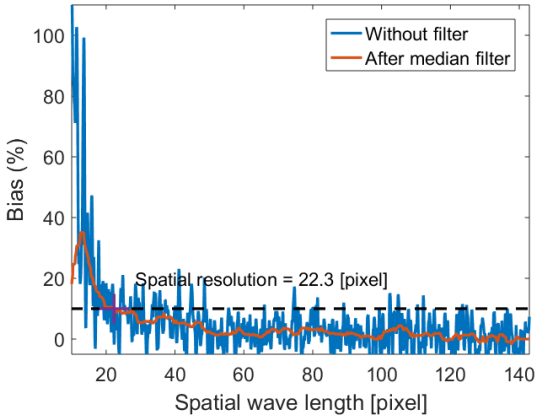


(a) StrainNet-h5

(b) StrainNet-h4



(c) StrainNet-h3



(d) StrainNet-h2

Figure 16: Seeking the spatial resolution of each version of StrainNet-h. The bias given here is a percentage of the displacement amplitude, which is equal to 0.5 pixel.

Figure 18-b were also processed by a CNN in Ref. [3]. Interestingly, the stiffness of the specimen changes along the vertical direction because of the annual rings which are clearly visible in Figure 18-a along the horizontal direction. Here, the vertical displacement is greater than 1 pixel while the CNN is trained on a dataset containing images deformed with a displacement lower or equal to one pixel only, as justified in Section 2.2.1. As in Ref. [3], the images are therefore processed sub-domain by sub-domain. These sub-domains are such that the round value of the displacement is the same within each sub-domain. This round value for the displacement can easily be found by cross-correlation or by applying a rough DIC for instance.

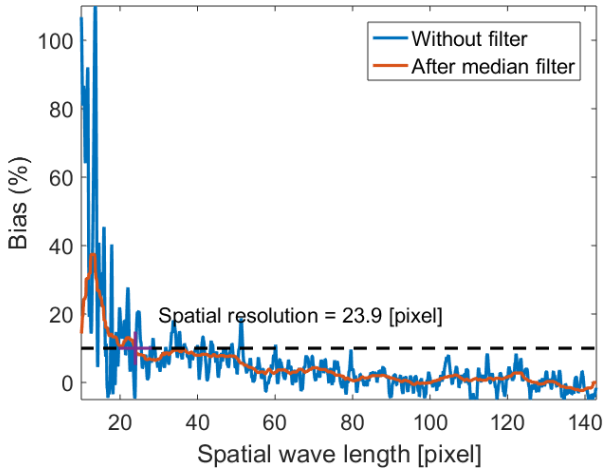


Figure 17: Seeking the spatial resolution of the simplified CNN. The bias given here is a percentage of the displacement amplitude, which is equal to 0.5 pixel.

The second option was adopted here. Figures 18-c and -d show first the results obtained by DIC performed with a shift equal to one pixel between subsets of size 21×21 pixels, the subset shape functions being here bilinear. The strain maps are obtained by convolving the displacement maps with a derivative kernel defined by the y-derivative of a Gaussian window of standard deviation equal to 6 pixels. The results obtained with StrainNet-h5 are shown in Figures 18-e and -f, and those obtained with StrainNet-l in Figures 18-g and -h. The main remark is that the global aspect of the maps is the same, in particular for the displacement maps where no difference can be detected to the naked eye. Derivation increasing the differences between the results obtained with different tools, they are more visible in the strain maps. It can be seen that the strain map is the smoothest with DIC. Details are more pronounced on the map obtained with StrainNet-h5, which is due to the fact that the highest frequencies are more easily detected with this tool. This trend is reinforced with StrainNet-l. This is logical since as mentioned above, the spatial resolution characterizing the maps obtained with StrainNet-l is smaller (thus better) than its counterpart characterizing the maps obtained with StrainNet-h5.

5 Conclusion

In this paper, it is shown that lightweight CNNs can be used to retrieve displacement maps from pairs of reference/deformed images."Lightweight" means here that a reasonable number of levels as well as a small number of filters per level can be potentially be used. We showed first that it was possible to simplify a CNN proposed in a recent study by using only 4 levels instead of 6, and that a better global metrological performance (estimated with the MAE) was obtained with these 4 levels instead of the 6 initial ones. Further simplification was then performed by reducing the number of filters in the 4 remaining levels, leading to a final simplified CNN version characterized by 0.67 millions coefficients instead of 38.68 millions for the initial version. An interesting conclusion is that the performance of this ultimate version of the CNN is not really affected by this second simplification. This performance globally lies between that of DIC used with first-order subset shape functions and DIC used with second-order subset shape functions, which tends to prove that the final lightweight CNN obtained in this study constitutes a viable alternative to DIC. The main visible effect of these two successive simplifications of the CNN is to improve the ability of the CNN to detect high spatial frequencies (thus to improve the spatial resolution) because the deepest layers which were removed mainly concern low spatial frequencies. The counterpart is that a higher noise level affects the displacement and strain maps. Another point is that the classic tradeoff between spatial resolution and measurement resolution can also be tailored by changing the version of the CNN, a general trend being that the lower the number of levels, the smaller (thus the better) the spatial resolution and the higher (thus the worse) the measurement resolution.

As a general remark, the experience on CNNs to measure displacement and strain fields is currently limited and developing specific CNNs to carry out this task is still a fledgling activity. Having a clear view on the causal relationship between the numerous factors that characterize CNNs and the quality of the maps they provide still necessitates additional

studies, for instance aimed to test other architectures, to change the content of the dataset used to train them for this specific task, to adjust the way this training is performed, or to enrich the pool of reference displacement fields used to assess the metrological performance. This will help bolster the use of CNNs to perform this task, better understand the pros and cons of CNNs for this type of application and provide important insights on optimizing their architecture. Another perspective is that considering lightweight CNNs for in-plane displacement measurement like the one studied in the present study paves the way for future applications, like the embedment of such CNNs in smart cameras. Such a camera would for instance be composed of a power-efficient compact GPU linked to a classic dedicated camera. Such a smart camera would directly provide nearly real-time displacement maps instead of images of speckle patterns.

Data Availability

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

Acknowledgment

This work has been sponsored by the French government research program "Investissements d'Avenir" through the IDEX-ISITE initiative 16-IDEX-0001 (CAP 20-25), the IMobS3 Laboratory of Excellence (ANR-10-LABX-16-01) and the AIM ANR project (ANR-20-THIA-0001).

References

[1] 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*, 69:7–12, 2015.

[2] V. Couty, J.-F. Witz, P. Lecomte-Grosbras, J. Berthe, E. Deletombe, and M. Brieu. GPUCorrel: A GPU accelerated digital image correlation software written in python. *SoftwareX*, 16:100815, 2021.

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

[4] R. Yang, Y. Li, D. Zeng, and P. Guo. Deep dic: Deep learning-based digital image correlation for end-to-end displacement and strain measurement. *Journal of Materials Processing Technology*, 302:117474, 2022.

[5] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. van der Smagt, D. Cremers, and T. Brox. FlowNet: Learning optical flow with convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015.

[6] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1647–1655, 2017.

[7] A. Maas, A. Hannun, and A. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.

[8] I.J. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, Cambridge, MA, USA, 2016.

[9] S. R. Dubey, S. K. Singh, and B. B. Chaudhuri. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomputing*, 503:92–108, 2022.

[10] F. Sur, B. Blaysat, and M. Grédiac. Rendering deformed speckle images with a Boolean model. *Journal of Mathematical Imaging and Vision*, 60(5):634–650, 2018.

[11] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10):1737–1754, 2008.

[12] M. Grédiac, B. Blaysat, and F. Sur. A critical comparison of some metrological parameters characterizing local digital image correlation and grid method. *Experimental Mechanics*, 57(6):871–903, 2017.

[13] B. Blaysat, J. Neggers, M. Grédiac, and F. Sur. Towards criteria characterizing the metrological performance of full-field measurement techniques. application to the comparison between local and global versions of DIC. *Experimental Mechanics*, 60(3):393–407, 2020.

[14] M. Grédiac, B. Blaysat, and F. Sur. On the optimal pattern for displacement field measurement: random speckle and DIC, or checkerboard and LSA? *Experimental Mechanics*, 60(4):509–534, 2020.

[15] P. L. Reu, B. Blaysat, E. Andó, K. Bhattacharya, C. Couture, V. Couty, D. Deb, S. S. Fayad, M. A. Iadicola, S. Jaminion, M. Klein, A. K. Landauer, P. Lava, M. Liu, L. K. Luan, S. N. Olufsen, J. Réthoré, E. Roubin, D. T. Seidl, T. Siebert, O. Stamati, E. Toussaint, D. Turner, C. S. R. Vemulapati, T. Weikert, J. F. Witz, O. Witzel, and J. Yang. Dic challenge 2.0: Developing images and guidelines for evaluating accuracy and resolution of 2d analyses. *Experimental Mechanics*, 62(4):639–654, 2022.

[16] M. Grédiac, F. Sur, C. Badulescu, and J.-D. Mathias. Using deconvolution to improve the metrological performance of the grid method. *Optics and Lasers in Engineering*, 51(6):716–734, 2013.

[17] M. Sutton, J.J. Orteu, and H. Schreier. *Image Correlation for Shape, Motion and Deformation Measurements. Basic Concepts,Theory and Applications*. Springer, 2009.

[18] Y. Lecun, B.E. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel. *Handwritten digit recognition with a back-propagation network*, pages 396–404. Chapman Hall/CRC Publishers, 1992.

[19] S. Han, J. Pool, J. Tran, and W. J. Dally. Learning both weights and connections for efficient neural networks. In *NIPS'15: Proceedings of the 28th International Conference on Neural Information Processing Systems*, volume 1, pages 1135–1143, Cambridge, MA, USA, 2015. MIT Press.

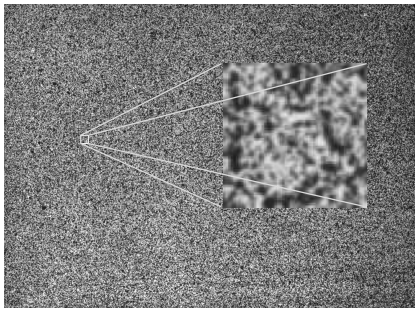
[20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[21] J. Gou, B. Yu, S.J. Maybank, and D. Tao. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819, 2021.

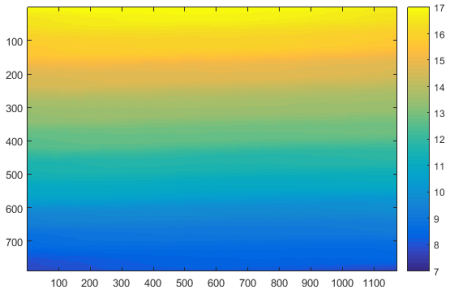
[22] L. Wittevrongel, P. Lava, S. V. Lomov, and D. Debruyne. A self adaptive global digital image correlation algorithm. *Experimental Mechanics*, 55(2):361–378, 2015.

[23] M. Grédiac, B. Blaysat, and 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.

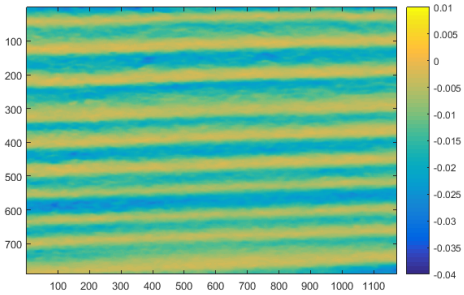
(a) Specimen before spray-painting, after [23]



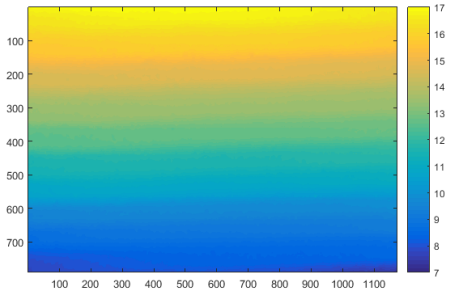
(b) Speckled surface of the specimen after spray painting, after [23]



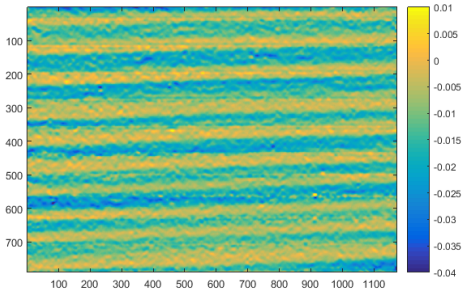
(c) DIC, $2M + 1 = 21$ pixels



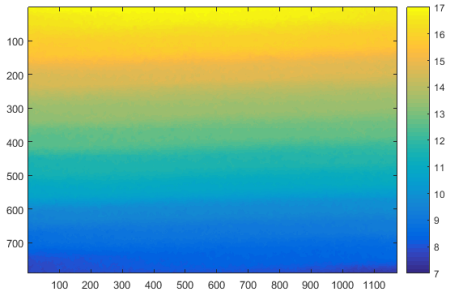
(d) DIC, $2M + 1 = 21$ pixels



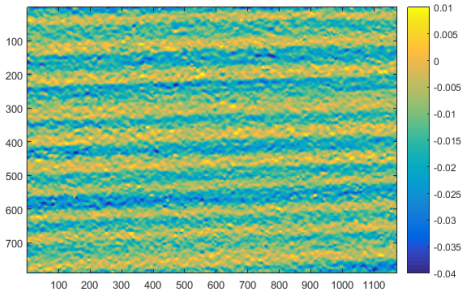
(e) StrainNet-h5



(f) StrainNet-h5



(g) StrainNet-l



(h) StrainNet-l

Figure 18: Results obtained by processing real images. Left: v displacement field in pixels. Right: ε_{yy} strain map. All dimensions are in pixels.