Obtaining subpixel level cutting tool displacements from video using a CNN architecture

Varun Raizada, Mohit Law*

Indian Institute of Technology Kanpur, Kanpur, India

Presented in International Conference on Precision, Micro, Meso and Nano Engineering (COPEN - 12: 2022)
December 8 - 10, 2022 IIT Kanpur, India

ABSTRACT

To register motion from video of vibrating tools, acquisition must ensure that motion is spatially and temporally resolved. However, since tools often vibrate with subpixel level motion, and since cameras often trade speed for resolution, if acquisition is to respect the Nyquist limit to avoid temporal aliasing, then the spatial resolution is often not sufficient to detect small cutting tool motion. To address this problem, this paper shows for the first time that subpixel level tool motion can be inferred instead by using convolution neural networks. We train our model on a database using the phase-based optical flow scheme that is a subpixel level motion registration algorithm. Our model is shown to be capable of detecting small motion correctly. Though the frequency of vibration estimated from the registered motion is correct, further work is necessary on fine tuning model architecture to fix the errors observed in the estimation of damping.

KEYWORDS

Convolution Neural Network,
Vibrations,
Phase-Based Optical Flow,
Visual Vibrometry.

1. Introduction

Vision-based modal analysis involves the extraction of modal parameters from motion registered from video of vibrating objects using image processing schemes. The method proffers advantages over the more traditional experimental modal analysis methods, especially for machine tool systems (Law et al., 2020; Gupta et al., 2021; Gupta & Law, 2021; Raizada et al., 2022). These include no mass loading effects, allowing for shape analysis with one measurement, and simple experimental setups. However, to truly leverage the potential of the methods, motion must be first resolved correctly. Since tools often vibrate at high frequencies with subpixel level motion (Gupta et al., 2021), and since cameras often trade speed for resolution, if acquisition is to respect the Nyquist limit to avoid temporal aliasing (Law et al, 2022; Rajput & Law, 2022; Lambora et al., 2022), then the spatial resolution is often not sufficient to detect small cutting tool motion (Raizada et al., 2022; Nuhman et al., 2022). Since recovery of motion from potentially temporally aliased signals has been addressed in prior work (Law et al, 2022; Rajput & Law, 2022; Lambora et al., 2022), this paper focuses its attention on addressing the problem of registering small tool motion from video.

Prior work on registering small motion has focused on the use of intensity- and phase-based optical flow algorithms (Nuhman et al., 2022; Javh et al., 2017; Luan et al., 2021), and/or on using hardware fixes such as extension tubes with lenses to magnify the field of view (Nuhman et al., 2022). Recent work has also reported the use of deep learning methods to infer small subpixel level motion of vibrating cantilevered structures and other civil infrastructure (Luan et al., 2021). In line with the work reported in (Luan et al., 2021), and because our prior work (Nuhman et al., 2022) has already addressed subpixel level motion registration using optical flow-based schemes and hardware fixes, this paper shows for the first time that subpixel level cutting tool motion can be inferred by using convolution neural networks.

Deep learning methods like convolutional neural networks (CNN) are seeing wide scale adoption across engineering domains for image/video analysis, and for motion estimation. The most striking feature of the CNN's architecture is its ability to successfully capture the spatial and temporal information in an image through the application of relevant filters which it can learn through training. This is an advantage that it has

*Corresponding author E-mail: mlaw@iitk.ac.in

https://doi.org/10.58368/MTT.22.3.2023.14-19

over other image processing techniques where filters are to be designed manually for every application of interest (Gupta et al., 2021; Raizada et al., 2022).

The oft-used CNN model for optical flow-based motion registration is the FlowNet model (Dosovitskiy et al., 2015). Since the FlowNet model has become the de facto CNN model, we too use the same architecture herein. This is supervised learning model whose model architecture comprises convolution and deconvolution layers with residual connections. The original model was trained on synthetic datasets, partly due to lack of sufficient 'ground truth' data to train the CNN. Since the original synthetic training dataset lacked subpixel level motion features, and since our aim is to extract subpixel level motion, we propose training the model with video that has subpixel level motion. We train the model using videos of a tool vibrating with potentially small and subpixel level motion, whose motion was recorded in our laboratory using a high-speed camera. Ground truths for training from these videos is generated using the phase-based optical flow algorithm which can provide an accurate displacement estimation even for subpixel motion (Nuhman et al., 2022; Luan et al., 2021; Fleet & Jepson, 1990).

The remainder of the paper is organized as follows. At first, in Section 2 we overview the CNN architecture. Section 3 discusses generation of training data sets. Section 4 discusses the experimental setups. These are used for training and for testing. Section 5 present results for two illustrative cases showing how the CNN model can estimate small subpixel level motion for a grooving blade and for an end mill. The main conclusions follow.

2. Overview of the Convolution Neural Network

The model architecture consists of several convolution layers followed by several deconvolution layers with residual connections between both layers. Every layer of a convolution neural network takes inputs, applies filters, and gives an output in the forward propagation. In the backward propagation of the network, filters are updated in such a way that they can extract useful information from the input. A schematic of such a network is shown in Fig. 1. The filters, otherwise called kernels with size $k \times k$ form the learnable unit of the neural network. k is a

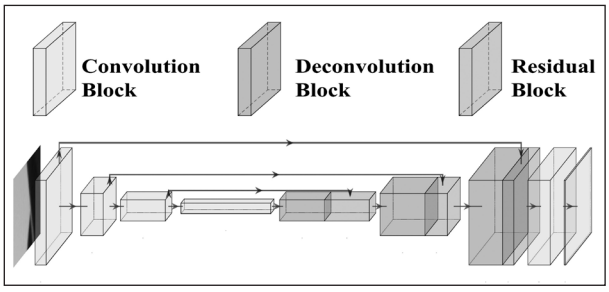

Fig. 1. CNN architecture for our model.

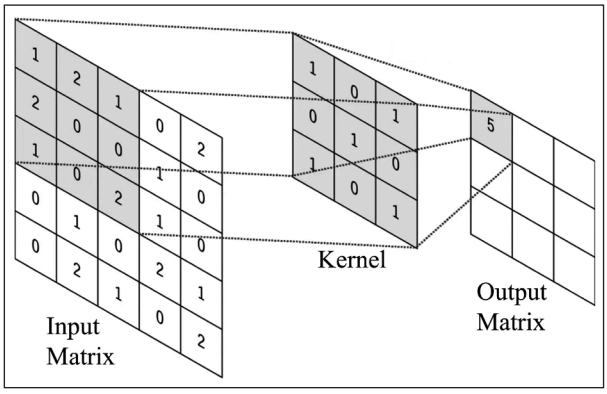

Fig. 2. Schematic of a convolution operation.

hyperparameter that must be given as an input by the user while designing the model architecture.

A convolution operation over a $n \times n$ sized input by a $k \times k$ sized kernel includes sliding the kernel over $k \times k$ subspace of input and performing a summation of the dot product of kernel and local subspace to give a single cell in the output as:

$$G(x,y) = \sum_{i=-k/2}^{k/2} \sum_{j=-k/2}^{k/2} W(i,j)F(x+i,y+j), \quad \text{....(1)}$$

wherein *G* is the output matrix, *F* is the input image matrix consisting of entries corresponding to pixel intensities, *W* is the kernel matrix of size $k \times k$ and x and y are coordinates of point output in matrix G as demonstrated in Fig. 2.

The total size of output after a convolution depends on two other hyperparameters namely, the stride which controls how much the kernel slides in every operation, and padding which controls the extra pixels added to an image border given by:

$$S_{\text{out}} = \frac{S_{\text{in}} + 2 \times p - k}{s} + 1, \quad \text{.....(2)}$$

wherein S_{out} is output size, S_{in} is input size, p is padding size, k is kernel size and s is the stride.

A deconvolution layer could be understood as backward convolution. It increases the spatial size of our input. It has hyperparameters like those of the convolution layer that also control its operations. The size of the kernels in the forward and backward propagation, the stride, and the number of layers are all tuned for our application of interest, as is discussed in Section 5 of this paper, i.e., after discussing how we generate datasets to train this model (Section 3) and what experiments we conduct for training and testing (Section 4).

3. Method to Generate Training Dataset

To train our CNN model for it to learn, we require ground truth datasets. For this, we use motion extracted from video of a tool vibrating with potentially small and subpixel level motion. To extract the ground truth data necessary for training, we use the phase-based optical flow scheme which is capable of subpixel level motion estimation (Nuhman et al., 2022; Luan et al., 2021) and has been shown to perform better than the intensity-based optical flow scheme (Nuhman et al., 2022). An overview of how we extract ground truth motion is summarized below. Details in (Nuhman et al., 2022; Fleet & Jepson, 1990).

In Phase-based optical flow, we process the image sequence with the help of an oriented Gabor filter to obtain information about the localized motion (Nuhman et al., 2022; Fleet & Jepson, 1990). Gabor filter is a complex filter that gives information about local phase ϕ and local amplitude A in orientation θ for an image of intensity $\mathrm{I}(x, y, t_0)$ at a time t_0:

$$A_\theta(x,y,t_0)e^{i\phi_\theta(x,y,t_0)} = (G_\theta + iH_\theta) * I(x,y,t_0), \quad ..(3)$$

wherein *G* and *H* are quadrature pair that differs in phase by 90° and represent a complex Gabor filter.

The phase-based scheme assumes a constant phase whose evolution with time gives the displacement signal (Fleet & Jepson, 1990):

$$\phi_\theta(x,y,t_0) = \mathrm{C}. \quad .....(4)$$

This constant phase assumption can be converted to a phase gradient equation expressed as:

$$\nabla\phi_\theta \cdot v + \psi = 0, \quad .....(5)$$

wherein $\nabla\phi = (\delta\phi/\delta x, \delta\phi/\delta y)$ is the spatial phase gradient, $\boldsymbol{v} = (\boldsymbol{v}_x, \boldsymbol{v}_y)$ is optical flow velocity in x and y direction and $\phi=(\delta\phi/\delta t)$ is temporal phase gradient.

For θ = 0 and θ = π/2, we get $\delta\phi_0/\delta y = 0$ and $\delta\phi_{\pi/2}/\delta x = 0$ respectively. Therefore, from Eq. (5) we obtain velocities in x and y directions as:

$$v_x = -\left[\frac{\delta\phi_0(x,y,t)}{\delta x}\right]^{-1}\left[\frac{\delta\phi_0(x,y,t)}{\delta t}\right], \quad .....(6)$$

and,

$$v_y = -\left[\frac{\delta\phi_{\frac{\pi}{2}}(x,y,t)}{\delta y}\right]^{-1}\left[\frac{\delta\phi_{\frac{\pi}{2}}(x,y,t)}{\delta t}\right]. \quad .....(7)$$

Displacement in x and y directions can be obtained by integrating velocities as:

$$d_x(t_0) = \left[\frac{\delta\phi_0(x,y,t_0)}{\delta x}\right]^{-1}[\phi_0(x,y,t_0) - \phi_0(x,y,0)], \quad ..(8)$$

and,

$$d_y(t_0) = \left[\frac{\delta\phi_{\frac{\pi}{2}}(x,y,t_0)}{\delta y}\right]^{-1}\left[\phi_{\frac{\pi}{2}}(x,y,t_0) - \phi_{\frac{\pi}{2}}(x,y,0)\right]. \quad ..(9)$$

Above procedures are used to generate datasets that serve as ground truths for training purposes. Experimental setups to acquire training datasets are discussed next.

4. Experimental Setups

The experimental setups to generate training data are shown in Fig. 3. These setups are also used to test our models. The setups include one to measure a flexible grooving blade mounted on a CNC turning machine – see Fig. 3(a), and another to measure a flexible end mill mounted on a CNC milling machine – see Fig. 3(b). We used a Chronos 2.1 camera to record videos of the tools when they were excited with an impulse excitation. The camera was focused on the free edge of the tool, since that is the location at which motion must be registered. We used a DC light and white background for all measurements such as to properly illuminate the tools being recorded and to minimize the influence of noise. For all measurements, we used a standard 18-55 mm lens with the aperture kept wide open and ISO settings as 2000. With the setups shown, we are only interested in estimating in plane motion, i.e., along the Z axis for the grooving blade, and along the Y axis for the end mill.

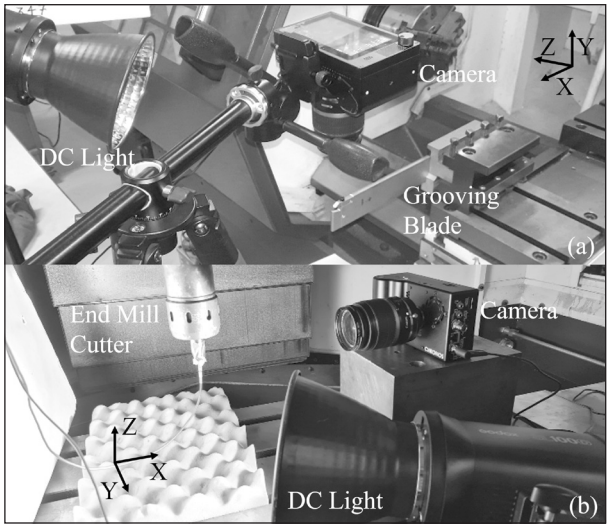

Fig. 3. Experimental setups for recording tool motion using a high-speed camera, (a) setup for grooving blade, and (b) setup for endmill cutter.

We recorded two videos of the vibrating grooving blade at frame rates of 5406 frames per second (fps) and at 10488 fps. The pixel resolution for the first recording was 800 x 480, and for the second it was 640 x 240. The tool had a width of 8 mm, making the per pixel resolution in the first case to be 87 µm, and it was 84 µm in the second case. Pixel sizes were different since the image resolutions in both cases were different. The end mill was recorded at 2142 fps and with an image resolution of 1280 x 720. In this case the camera was placed 115 mm away from the tool that had a diameter of 16 mm, making the per pixel resolution 83 µm. These pixel resolutions are larger than the expected motion amplitude, i.e., subpixel level motion is likely, and hence the phase-based optical flow scheme is expected to be able to resolve this motion. The three videos together make up a total of 90,000 images – a dataset thought to be large enough to properly train the CNN model.

5. Results with CNN

This section discusses results obtained from our supervised and trained CNN model. In our implementation, the input consists of a reference and a current image, i.e., an image pair that is stacked together making our input size in pixels to be 48×48×2. This input is passed through four convolution layers with kernel sizes of 7×7, 5×5, 3×3, and 3×3 with an output size of 8, 16, 32, and 64, respectively. The stride size is kept as 2 pixels. This gives us an output after convolution of dimensions 6×6×64. The output from the convolution is passed through three deconvolution

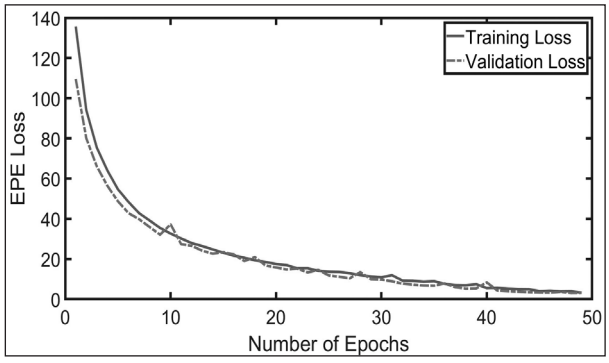

Fig. 4. Comparison of training and validation loss characterized by the end point error.

layers with kernel sizes 4×4, 4×4, and 3×3, respectively. After each deconvolution, we add input features from convolution layers residually making our output size to be 64, 48, and 32. Finally, we add two more convolution layers to get an output of dimension 48×48×1 which represents our full field horizontal displacements. The architecture is implemented using the Pytorch python library (Pinard, 2022).

Consecutive frames were stacked together with the first frame while creating the dataset. The generated dataset was further divided into two parts for training and validation during our training process. The learning rate for the model was kept constant at 0.001, the Adam optimizer was used for gradient descent, and the endpoint error (EPE) (Luan et al., 2021) was used as the loss function with a batch size of 500. The model was trained for fifty epochs with the loss for both training and validation set decreasing in sync with validation loss not exceeding training loss which ensured that there is no overfitting in our model as shown in Fig. 4.

To test our model, we took a different and smaller set of images from the recorded videos of the vibrating tools and processed them in a similar manner as that of the training dataset to create a testing database. Testing the model on previously unseen data would ensure the generalization of our model. Displacement thus obtained from our CNN model are shown in Fig. 5. Also shown in Fig. 5 are displacements obtained from the phase-based optical flow scheme – which is the ground truth for benchmarking.

It is evident from Fig. 5 that though the time period of the learnt response from the CNN model for both tools is like that of the ground truth estimated from the phase-based optical flow method, there is a marked difference in the magnitudes.

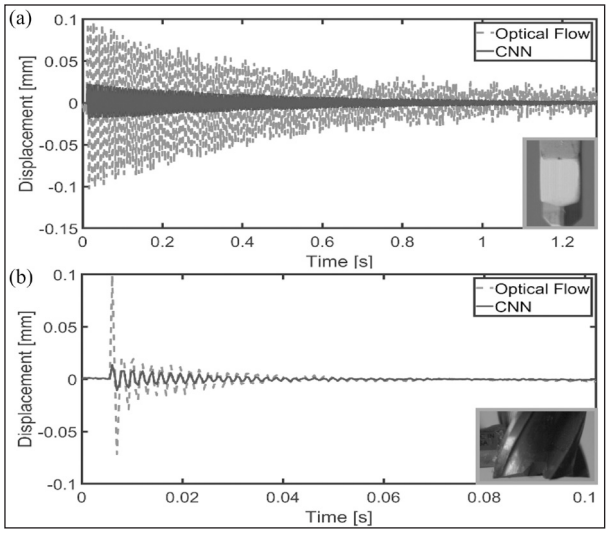

Fig. 5. Comparison between the displacements predicted by phase-based optical flow and those learnt using the CNN model, (a) response of the grooving blade, and (b) response for endmill cutter.

Table 1
Modal parameters comparison.

Method	Grooving Blade		Endmill Cutter	
	f [Hz]	ζ [%]	f [Hz]	ζ [%]
Optical Flow	136	0.24	525	2.15
CNN Model	136	0.22	524	0.82

This suggests that the CNN model may need further improvement by tuning.

Since the objective of registering motion from video is to extract modal parameters from that video, we extract modal parameters from response obtained for each tool using eigensystem realization algorithm (ERA) (Gupta et al., 2020) and list those in Table 1. We also list in Table 1, parameters extracted from phase-based optical flow that serve as the benchmark. And as is evident from Table 1, the CNN model can correctly estimate the natural frequencies (f), however, the damping ratio (ζ), especially for the end mill, is grossly underestimated. Since damping governs the magnitude and decay of the response, differences observed in the registered motion naturally transmit to errors in damping estimated.

6. Conclusion

This paper showed, for the first time, that a supervised convolution neural network model can be trained to learn and predict small subpixel level cutting tool motion. The model was trained on data generated from in-house recordings of cutting tools vibrating with small motion. For training, a phase-based optical flow algorithm was used to estimate small displacements that served as ground truths. Predictions from the CNN model shows that though frequencies of vibration are estimated correctly, there are, in some instances errors in estimating the vibration amplitudes. Further work is necessary to fine tune model architecture amongst other things to improve predictions.

Acknowledgments

This work was supported by the Government of India's Science and Engineering Research Board's Core Research Grant number CRG/2020/001010.

References

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Smagt, P. van der, Cremers, D., & Brox, T. (2015). Flownet: Learning optical flow with convolutional networks. *In Proceedings of the IEEE International Conference on Computer Vision,* 2758-2766. https://doi.org/10.1109/iccv.2015.316

Fleet, D. J., & Jepson, A. D. (1990). Computation of component image velocity from local phase information. *International Journal of Computer Vision, 5*(1), 77-104. https://doi.org/10.1007/BF00056772

Gupta, P., & Law, M. (2021). Evaluating tool point dynamics using smartphone-based visual vibrometry. *Procedia CIRP, 101,* 250-253. https://doi.org/10.1016/j.procir.2020.09.196

Gupta, P., Law, M., & Mukhopadhyay, S. (2020). Evaluating tool point dynamics using output-only modal analysis with mass-change methods. *CIRP Journal of Manufacturing Science and Technology, 31,* 251-264. https://doi.org/10.1016/j.cirpj.2020.06.001

Gupta, P., Rajput, H. S., & Law, M. (2021). Vision-based modal analysis of cutting tools. *CIRP Journal of Manufacturing Science and Technology, 32,* 91-107. https://doi.org/10.1016/j.cirpj.2020.11.012

Javh, J., Slavič, J., & Boltežar, M. (2017). The subpixel resolution of optical-flow-based modal analysis. *Mechanical Systems and Signal Processing, 88,* 89-99. https://doi.org/10.1016/j.ymssp.2016.11.009

Lambora, R., Law, M., & Mukhopadyay, S. (2022). Recovering cutting tool modal parameters from

fractionally uncorrelated and potentially aliased signals. *CIRP Journal of Manufacturing Science and Technology, 38,* 414-426. https://doi.org/10.1016/j.cirpj.2022.05.014

Law, M., Gupta, P., & Mukhopadhyay, S. (2020). Modal analysis of machine tools using visual vibrometry and output-only methods. *CIRP Annals, 69*(1), 357-360. https://doi.org/10.1016/j.cirp.2020.04.043

Law, M., Lambora, R., & Mukhopadhyay, S. (2022). Modal parameter recovery from temporally aliased video recordings of cutting tools. *CIRP Annals.* https://doi.org/10.1016/j.cirp.2022.03.023

Luan, L., Zheng, J., Wang, M. L., Yang, Y., Rizzo, P., & Sun, H. (2021). Extracting full-field subpixel structural displacements from videos via deep learning. *Journal of Sound and Vibration, 505,* 116142.https://doi.org/10.1016/j.jsv.2021.116142

Nuhman, P., Singh, A., Lambora, R., & Law, M. (2022). Methods to estimate subpixel level small motion from video of vibrating cutting tools. *CIRP Journal of Manufacturing Science and Technology, 39,* 175-184. https://doi.org/10.1016/j.cirpj.2022.08.005

Pinard, C. (2022). FlowNet Pytorch Implementation https://github.com/ClementPinard/FlowNetPytorch, last accessed 2022/08/28.

Raizada, V., Rajput, H. S., & Law, M. (2023). Comparative analysis of image processing schemes to register motion from video of vibrating cutting tools. Communicated for consideration of presentation in the 11th CIRP Global Web Conference (CIRPe 2023) and for appearing in the Procedia CIRP.

Rajput, H. S., & Law, M. (2023). Recovering cutting tool modal parameters from randomly sampled signals using compressed sensing. *Manufacturing Technology Today, 22*(2), 17-22.

Varun Raizada is a final year Master's Student in the Department of Mechanical Engineering at the Indian Institute of Technology Kanpur, India. He has received his Bachelor's Degree in Mechanical Engineering from Delhi Technological University, Delhi. His research interests centre on machine tools vibration, image processing and machine learning. (E-mail: varunr21@iitk.ac.in)

Mohit Law is an Associate Professor at the Department of Mechanical Engineering at the Indian Institute of Technology Kanpur, India. He received his Ph.D from the University of British Columbia, Canada. His research interests centre on understanding how and why machine tools vibrate, measuring those vibrations, and on how best to mitigate them.