Improving Space-Time Image Velocimetry Accuracy with Deep Learning: Integrating Synthetic and Real-World Data

Tenorio, A[a], Fernández, R[a], Engel, F[b], Liu, X[a]

[a] *Pennsylvania State University, Department of Civil and Environmental Engineering University Park, Pennsylvania, USA*
[b] *U.S. Geological Survey, Water Mission Area, Observing Systems Division, Hydrologic Remote Sensing Branch, San Antonio, Texas, USA*

Keywords: STIV, Deep Learning, Flow measurements

Abstract

Quantifying streamflow is essential for resource management, habitat monitoring, and emergency response, but extreme events pose challenges for direct measurements due to safety and accessibility concerns. Remote sensing techniques, such as Space-Time Image Velocimetry (STIV), allow for non-contact measurements of velocity and discharge. STIV uses video footage of the water surface to generate Space-Time Images (STI), assuming surface textures will act as passive tracers. However, environmental conditions such as sunlight conditions, surface reflections or rain can hinder the quality of the STIs. While Wavenumber–Frequency Spectra (WFS) filters can improve STI quality, they struggle in real-time applications. Incorporating deep learning can enhance real-time accuracy, as demonstrated in previous work with synthetic STIs. We aim to extend this by combining synthetic, computational, laboratory-controlled, and real-world data to improve the accuracy of flow measurements in natural settings.

1. Introduction

Different approaches have been explored to improve the accuracy and reliability of STIV. [1] proposed a method to estimate the velocity of surface waves using a combination of Wavenumber–Frequency Spectra (WFS)

filters. The method consist on generating a mask that ignores a portion of the WFS that does not seem to be related to the surface flow. [2] proposed a deep learning approach that trains a convolutional neural network (CNN) to estimate the predominant stripe pattern angle of the STI. The CNN was trained using synthetic STIs with results showing promising performance on STIs from real river footage.

Relying on CNNs to estimate the stripe pattern angle of the STI is a promising approach, but it is limited by the restraint of a

2. Background

Introduced by Fujita et al (2007), gradient tensor method [3] further improved in 2018 with a two dimensional autocorrelation function for the image intensity distribution of STI that is used for detecting the most probable texture gradient included in STI [4].

$$R(\tau_x, \tau_y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) f(x - \tau_x, y - \tau_y) dx dy, \tag{1}$$

where f(x,y) is the image pixel intensity distribution in the STI and (τ_x, τ_y) are shift space and time parameters.

The mean velocity component at the line segment can be estimated by

$$U = \frac{S_x}{S_t} \tan(\theta_{max}) \tag{2}$$

where S_x is the size of an image pixel and S_t is the time interval between frames.

Most recently, STIV started adopting a machine learning approach to improve the accuracy of the velocity estimation. Watanabe et al. proposed a deep learning approach that trains a convolutional neural network (CNN) to estimate the predominant stripe pattern angle of the STI [2]. The CNN was trained using synthetic STIs with results showing promising performance on STIs from real river footage.

Although, STIV remains as a 1D technique for estimating surface velocities. This assumes the search line has to be placed in the streamwise direction [3, 4, 2, 1]. For an automated technique, this is not ideal since it requires more user intervention, limiting its applicability in real-time or large-scale scenarios.

3. Methodology

3.1. Training set

3.2. Deep Learning Architecture

The angle prediction model employs a Convolutional Neural Network (CNN) architecture. The input to the network is a single-channel 128x128 grayscale STI. The architecture consists of four convolutional blocks and one fully connected (FC) layers, as detailed in Table 1. This design allows the network to learn increasingly complex features as it progresses through the layers.

In between convolutions, the specified activation (GELU) and normalization (Batch Normalization) layers are applied in sequence. The GELU activation function is chosen for its smooth approximation of the ReLU function, which helps mitigate the "dying ReLU" problem [5, 6]. Batch Normalization (BN) is applied to stabilize and accelerate training by normalizing the outputs of the convolutional layers [7].

Following the convolutional blocks, a Global Average Pooling (GAP) layer is applied to the feature maps to avoid overfitting [2, 8]. The output of the GAP layer is then fed into a fully connected (FC) head consisting of two linear layers. Finally, the output is a single value representing the angle of the predominant stripe pattern in the STI. This output is subsequently scaled to the normalized target range of [-0.5, 0.5].

The model is trained using Mean Squared Error (MSE) loss, and the Adam optimizer is employed for optimization.

3.3. Detecting flow direction (ϕ)

3.4. Predicting flow velocity (θ_{max})

References

[1] I. Fujita, T. Shibano, K. Tani, Application of masked two-dimensional fourier spectra for improving the accuracy of stiv-based river surface flow velocity measurements, Measurement Science and Technology 31 (9) (2020) 094015.

[2] K. Watanabe, I. Fujita, M. Iguchi, M. Hasegawa, Improving accuracy and robustness of space-time image velocimetry (stiv) with deep learning, Water 13 (15) (2021) 2079.

Table 1: CNN Architecture for Angle Prediction from STI Images.

Layer Type	Input Shape	Output Shape	Kernel size	Filters/Units	Activation	Norm.
Convolutional Block 1						
Conv2D	1x128x128	32x128x128	3x3	32	GELU	BN2D
MaxPool2D	32x128x128	32x64x64	2x2	—	—	—
Convolutional Block 2						
Conv2D	32x64x64	64x64x64	3x3	64	GELU	BN2D
MaxPool2D	64x64x64	64x32x32	2x2	—	—	—
Convolutional Block 3						
Conv2D	64x32x32	128x32x32	3x3	128	GELU	BN2D
MaxPool2D	128x32x32	128x16x16	2x2	—	—	—
Convolutional Block 4						
Conv2D	128x16x16	128x16x16	3x3	128	GELU	BN2D
MaxPool2D	128x16x16	128x8x8	2x2	—	—	—
GAP	128x8x8	128x1x1	8x8	128	—	—
Flatten	128x1x1	128	—	—	—	—
Fully Connected (FC) Block						
Linear	128	128	—	128	GELU	BN1D
Linear	128	1	—	1	Tanh	—

Notes: BN2D: Batch Normalization 2D. BN1D: Batch Normalization 1D. For Conv2D layers,

[3] I. Fujita, H. Watanabe, R. Tsubaki, Development of a non-intrusive and efficient flow monitoring technique: The space-time image velocimetry (stiv), International Journal of River Basin Management 5 (2) (2007) 105–114.

[4] I. Fujita, Y. Notoya, K. Tani, S. Tateguchi, Efficient and accurate estimation of water surface velocity in stiv, Environmental Fluid Mechanics 19 (5) (2019) 1363–1378.

[5] D. Hendrycks, K. Gimpel, Gaussian error linear units (gelus), arXiv preprint arXiv:1606.08415 (2016).

[6] L. Lu, Y. Shin, Y. Su, G. E. Karniadakis, Dying relu and initialization: Theory and numerical examples, arXiv preprint arXiv:1903.06733 (2019).

[7] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International conference on machine learning, pmlr, 2015, pp. 448–456.

[8] M. Lin, Q. Chen, S. Yan, Network in network, arXiv preprint arXiv:1312.4400 (2013).