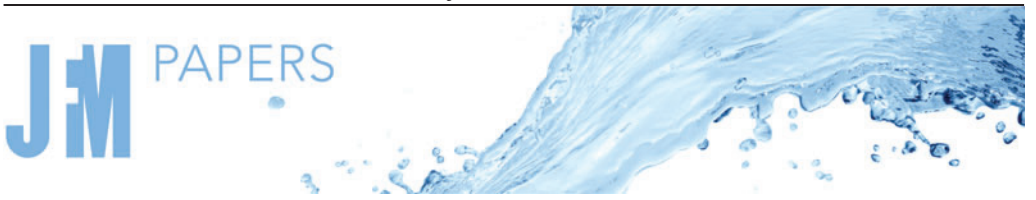
J. Fluid Mech. (2022), *vol*. 939, A2, doi:10.1017/jfm.2022.135



Accurate prediction of the particle image velocimetry flow field and rotor thrust using deep learning

Sehyeok Oh[1,2], Seungcheol Lee[1], Myeonggyun Son[1], Jooha Kim[1,†] and Hyungson Ki[1,†]

[1]Department of Mechanical Engineering, Ulsan National Institute of Science and Technology (UNIST), 50 UNIST-gil, Ulsan 44919, South Korea

[2]Department of Materials AI & Big-Data, Korea Institute of Materials Science (KIMS), 797 Changwon-daero, Changwon 51508, South Korea

(Received 22 July 2021; revised 11 January 2022; accepted 7 February 2022)

With particle image velocimetry (PIV), cross-correlation and optical flow methods have been mainly adopted to obtain the velocity field from particle images. In this study, a novel artificial intelligence (AI) architecture is proposed to predict an accurate flow field and drone rotor thrust from high-resolution particle images. As the ground truth, the flow fields past a high-speed drone rotor obtained from a fast Fourier transform-based cross-correlation algorithm were used along with the thrusts measured by a load cell. Two deep-learning models were developed, and for instantaneous flow-field prediction, a generative adversarial network (GAN) was employed for the first time. It is a spectral-norm-based residual conditional GAN translator that provides a stable adversarial training and high-quality flow generation. Its prediction accuracy is 97.21 % (coefficient of determination, or R^2). Subsequently, a deep convolutional neural network was trained to predict the instantaneous rotor thrust from the flow field, and the model is the first AI architecture to predict the thrust. Based on an input of the generated flow field, the network had an R^2 accuracy of 94.57 %. To understand the prediction pathways, the internal part of the model was investigated using a class activation map. The results showed that the model recognized the area receiving kinetic energy from the rotor and successfully made a prediction. The proposed architecture is accurate and nearly 600 times faster than the cross-correlation PIV method for real-world complex turbulent flows. In this study, the rotor thrust was calculated directly from the flow field using deep learning for the first time.

Key words: machine learning

† Email addresses for correspondence: hski@unist.ac.kr, kimjooha@unist.ac.kr

© The Author(s), 2022. Published by Cambridge University Press

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

1. Introduction

Particle image velocimetry (PIV) is a quantitative and non-intrusive flow analysis method that has been widely used to visualize complex flows and measure instantaneous velocities (Adrian, Adrian & Westerweel 2011). Thus far, cross-correlation (CC) and optical flow (OF) methods have been actively employed to extract the velocity using displacement vectors from two consecutive images. With the recent advancements in artificial intelligence (AI) in computer science field, a deep-learning-based method has emerged, which provides a new way of obtaining the velocity by conducting PIV.

Recently, a few PIV studies that translated particle images into a flow field using AI have been reported. Rabault, Kolaas & Jensen (2017) introduced a PIV-performing AI using a convolutional neural network (CNN) (Lecun *et al.* 1998) and artificial neural network. They evaluated the AI using realistic synthetic images and real-world data. Lee, Yang & Yin (2017) developed a cascaded CNN model and tested it using a synthetic uniform flow and experimental PIV images of a vortex pair and turbulent jet flow. To predict the velocity field at high resolution, Cai *et al.* (2019*b*) proposed an OF-based CNN model that exhibited good performance in terms of accuracy and computation time. Cai *et al.* (2019*a*) reported a CNN model for PIV that enhanced the accuracy of small-scale vortex structures and robustness to particle image noise. The model was investigated using a vortical flow field, two-dimensional turbulent flow and laminar and turbulent boundary-layer flows. Zhang & Piggott (2020) used an unsupervised CNN model for PIV and generated, with comparable quality, velocity fields from various types of artificial particle flows.

In the field of computational fluid dynamics (CFD), several AI models have been developed to predict flow fields or forces using CFD datasets. Miyanawala & Jaiman (2017) presented a CNN model to predict unsteady fluid forces (namely, drag and lift coefficients) past bluff-body geometries at low Reynolds numbers. They also reported a CNN model combined with a proper orthogonal decomposition (POD) technique, which was applied to predict the time series of turbulent wake flows and unsteady pressure fields of a fluid–structure system for a square cylinder (Miyanawala & Jaiman 2019). Bukka *et al.* (2021) proposed two deep-learning models, namely, POD-based recurrent neural networks and convolutional recurrent autoencoder networks, to predict a wake pattern for a side-by-side cylinder with a time history of the drag and lift forces.

In this study, two deep-learning models were developed to predict a highly accurate flow field and drone rotor thrust from high-resolution particle images (2048×2048 pixels). The novel contributions of this study are as follows. First, it is the first application of a generative adversarial network (GAN) (Goodfellow *et al.* 2014) (more specifically, spectral-norm-based residual conditional GAN, or SNRGAN, as it was called by the authors) to translate a pair of PIV particle images into a flow field. Our GAN translator provides stable adversarial training and generates much better quality flow compared with those of common convolutional encoder–decoder models (Hinton & Salakhutdinov 2006) with a regressive objective (mean absolute error (MAE) or mean squared error (MSE)), which have commonly been adopted in other AI + PIV studies. These existing practices are known to produce blurry predictions with missing details (Pathak *et al.* 2016; Zhang, Isola & Efros 2016; Isola *et al.* 2017).

Second, this study is the first attempt to predict the instantaneous thrust of a single rotor directly from an instantaneous flow field using deep learning. It is well known that accurately predicting rotor aerodynamics remains a challenge in the CFD community (Yoon, Lee & Pulliam 2016). The prediction result was analysed using a class activation map, which demonstrated that the model identified the area that accepted the kinetic energy from the rotor and made predictions. In our opinion, this research offers an

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

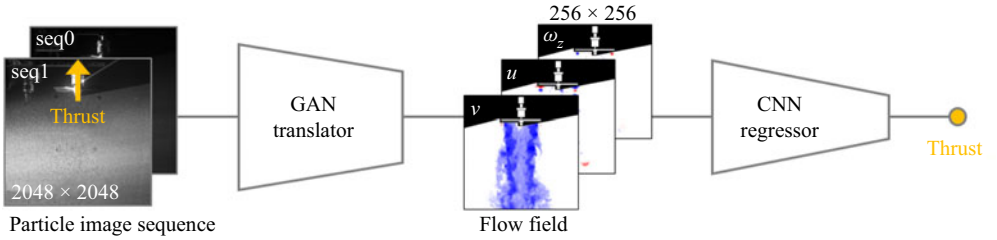


Figure 1. Schematic flow diagram of the proposed deep-learning architecture.

important insight into how deep learning can be applied to numerous other complex flow problems to obtain important flow characteristics (such as the lift and drag forces) directly from either PIV- or CFD-obtained flow fields.

Third, in this study, a real-world complex turbulent flow was investigated instead of relatively simple flows, such as synthetic, uniform laminar, vortex pair, turbulent jet and turbulent boundary-layer flows, which have been mainly tested in other AI + PIV studies. The flow across a drone rotor (a high-speed rotor operating at up to 9000 revolutions per minute (RPM)) examined in this study is extremely complicated because it is three-dimensional, highly turbulent and unsteady in nature. The most dominant vortical structure of the flow across a drone rotor is the rotor tip vortex, which is convected downstream in the form of a contracting helix, producing a significant three-dimensionality in the rotor wake. Another prominent vortical structure is the turbulent vortex sheet, which extends over the entire span of the rotor blade and is convected faster near the rotor tip than at the hub, which results in a turbulent conical inner wake region. The rotor tip vortex and turbulent vortex sheet have opposite vorticity signs and are located very close to each other, which leads to a highly unsteady interaction between the two vortical structures. The particle image obtained from this complex turbulent flow was used as a deep-learning model input, and to preserve the maximum particle information and flow characteristics, the high-resolution (2048×2048 pixels) image size was retained without any preprocessing.

Finally, very fast predictions are possible with the developed deep-learning models, even when using high-resolution input particle images. The computation took only 12.5 s to predict 270 flow fields (approximately 600 times faster than the conventional CC-based PIV technique) and 2.1 s to predict the same amount of instantaneous rotor thrusts.

2. Data preparation

Figure 1 presents a schematic flow diagram of the AI architecture developed in this study. It consists of two parts: (i) particle-image-to-instantaneous-flow-field translation using a GAN model (Goodfellow *et al.* 2014; Mirza & Osindero 2014) and (ii) flow-field-to-thrust regression using a CNN model (He *et al.* 2016). The detailed structures of these models are presented in figures 3(*a*) (GAN) and 3(*b*) (CNN), respectively.

For the data set-up, 1170 pairs of particle image sequences, flow fields calculated from them using a conventional PIV algorithm (CC) and thrust forces corresponding to each flow field were prepared experimentally using a drone rotor. A detailed description of the experimental set-up is provided in Lee *et al.* (2021). A particle image sequence consisted of two-channel images of 2048×2048 pixels, covering an area of 207×207 mm^2, captured by a charge-coupled device camera with a time interval of 20 μs. An example of a pair of particle image sequences (seq0 and seq1) obtained at 8000 RPM is shown on the left

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

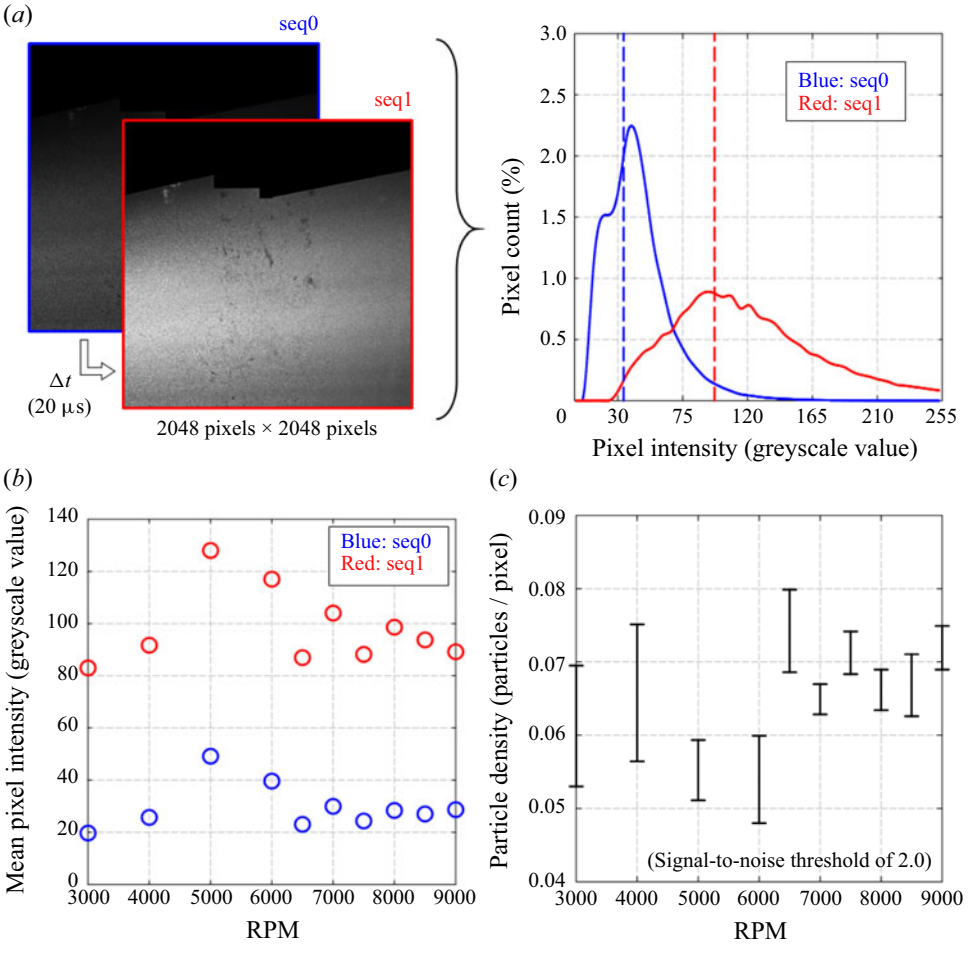


Figure 2. (*a*) A particle image pair example (left) and its pixel intensity (greyscale value) distribution (right). Vertical dashes represent mean of the distributions (blue, seq0; red, seq1), (*b*) mean pixel intensity and (*c*) particle density (seeding density, or particles per pixel), according to the rotor RPM.

side of figure 2(*a*). The image for seq1 is brighter (i.e. has higher pixel intensities) than that for seq0 because the camera exposure time for seq1 was set to be longer than that for seq0. Note that the rotor model and shadow cast by the model are masked in black at the top of the image; this area was excluded from the calculations when constructing the three graphs presented in figure 2.

The right side of figure 2(*a*) shows the pixel intensity distribution for the image sequences shown on the left side of the figure. The image pixel intensity for seq0 was concentrated from a greyscale value of 0 to approximately 110, with a mean value of 34; whereas that for seq1 was widely distributed, with a higher mean greyscale value of 97. Figure 2(*b*) shows the variation in the pixel intensity averaged over images taken at a given rotor speed. The mean pixel intensity of the particle image sequences used in this study varied with rotor speed in the range of 19–50 for seq0 and 83–128 for seq1, as shown in the figure. The particle images used differ not only in the pixel intensity distributions but also in the particle densities, depending on the rotor speed. The vertical bar in figure 2(*c*) indicates the particle density range with a signal-to-noise threshold of

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

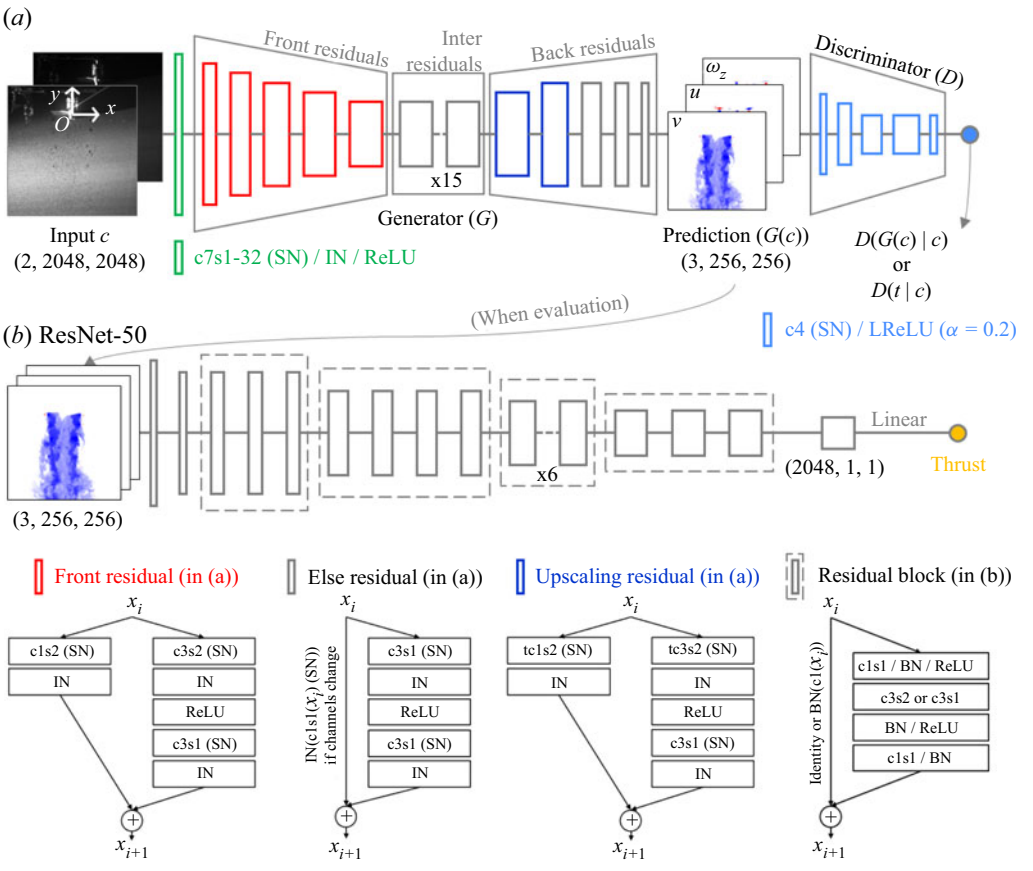


Figure 3. (*a*) Spectral-norm-based residual cGAN model (SNRGAN) for translating the input particle image sequence to the flow field. (*b*) CNN model for predicting the thrust from the flow field. At the bottom, legends for the residual blocks in (*a*,*b*) are presented. (c3s2, 3×3 convolution with stride 2; tc3s2, 3×3 transposed convolution (Zeiler *et al.* 2010) with stride 2; IN, instance normalization of the layer (Ulyanov, Vedaldi & Lempitsky 2016); SN, spectral normalization of the weight; BN, batch normalization of the layer (Ioffe & Szegedy 2015).)

2 in the images captured at a given rotor speed. As shown, the particle density in the images ranges from 0.048 particles/pixel to 0.08 particles/pixel, which corresponds to 12–20 particles in an interrogation window. The rotor speed was varied from 3000 to 9000 RPM (3000, 4000, 5000, 6000, 6500, 7000, 7500, 8000, 8500 and 9000; a total of ten RPM setpoints). The corresponding Reynolds numbers based on the rotor tip speed were 12 633, 16 844, 21 055, 25 266, 27 371, 29 477, 31 582, 33 688, 35 793 and 37 899, respectively. Consequently, particle images with diverse pixel intensity distributions, particle densities and RPM setpoints were included in the training data, allowing the development of a more robust deep-learning model. Note that consideration could also be given to introducing more variations in the data, such as employing noise in the input particle images, changing the particle size (currently 1 μm) or varying the laser light intensity (currently, 3.40 W cm^{-2}).

Using the particle images described above, the flow field was calculated as a three-channel image consisting of instantaneous y-directional velocity v, x-directional velocity u and z-directional vorticity ω_z (refer to figure 3*a* for the adopted coordinate

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

	Particle image sequence	Flow field	Thrust
Unit tensor size	1 seq = 2 × 2048 × 2048 (seq0, seq1)	1 set = 3 × 256 × 256 (v, u, ω_z)	1 thrust = 1 × 1 (a scalar)
Training	700 seqs (21.8 GB)	700 sets (3.26 GB)	700 thrusts (17.0 KB)
Validation	200 seqs (6.25 GB)	200 sets (955 MB)	200 thrusts (4.88 KB)
Test	270 seqs (8.43 GB)	270 sets (1.25 GB)	270 thrusts (6.60 KB)

Table 1. Data configurations for training, validation and test datasets, including unit tensor size, quantity and disk space occupied.

system). A fast Fourier transform (FFT)-based CC algorithm was employed for the calculation using an in-house PIV software under the following conditions. An iterative CC analysis was conducted recursively from an initial interrogation size of 64 × 64 pixels to a final size of 16 × 16 pixels with a 50 % overlap, resulting in a corresponding spatial resolution of 1.08 % of the rotor diameter. The spurious vectors were detected with a local median filter that rejected vectors greater than three times the root mean square (r.m.s.) value of a 3 × 3 window. The outliers were replaced with linearly interpolated values from the neighbouring vectors. Image sharpening for the pixel-wise change in intensity was applied for image preprocessing, and polynomial subpixel interpolation was used to reduce errors and improve spatial accuracy. This method was applied in our previous studies (Chae *et al.* 2019; Lee *et al.* 2021) and has been proven to be accurate. The flow fields obtained using the above method were compared with those calculated using a commercial program (Dantec DynamicStudio) utilizing the same particle images for further validation. This comparison showed an MAE value of 0.0916 m s^{-1} (0.00350 when non-dimensionalized), root mean square error (RMSE) of 0.160 m s^{-1} (0.00597 when non-dimensionalized) and R^2 value of 96.185 %, confirming the reliability of the flow fields used in this study. The rotor thrust was measured simultaneously with the flow field by setting the force sampling rate equal to the PIV repetition rate. In this study, the particle image sequences, flow fields calculated from the images using CC and thrust forces corresponding to each flow field were treated as the 'ground truth' for the deep-learning models. Note that these ground-truth data are inherently imperfect and may contain errors.

Before training the models, the data were divided into training, validation and test datasets. Table 1 presents the detailed data configurations for the training, validation and test datasets, including the unit tensor size, quantity and occupied disk space. Each dataset contained the same amount of data for each RPM setpoint (e.g. the test data for the flow field included twenty-seven flow fields per RPM).

3. Deep-learning architecture

As depicted in figure 3(*a*), for instantaneous flow-field predictions (i.e. y-directional velocity v, x-directional velocity u and z-directional vorticity ω_z), a GAN translator is adopted instead of a CNN-based encoder–decoder model with a regressive objective, which has been used in other AI studies. For CNN models using the mean absolute error or mean squared error as a cost function, the generated results tend to be blurry (Pathak *et al.* 2016; Zhang *et al.* 2016; Isola *et al.* 2017) because training is simply directed toward reducing the mean of the image difference. However, when a GAN is introduced, images with much better quality (i.e. higher accuracy) can be generated because the discriminator

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

imposes a penalty on patches assessed to be different from their real representations, such as blurred areas. (Detailed comparison results can be found in Appendix C.) Moreover, in our GAN model, the computational cost of the discriminator is much lower than that of the generator (∼1 %), resulting in an efficient accuracy improvement. High-quality flow generation also helps improve the performance of the thrust prediction model because it takes the predicted flow field as an input during evaluation.

As presented in figure 3(*a*), the developed GAN model is composed of two trainable functions, namely, the flow-field generator and discriminator (denoted by G and D, respectively). First, the generator accepts an experimentally observed particle image sequence (denoted by c) and translates it into an instantaneous flow field (a three-channel flow-field image; denoted by $G(c)$) through deep convolution layers. To avoid damaging the particle information, a 2048 × 2048-pixel high-resolution input particle image was used without any preprocessing (e.g. interpolation to low resolutions). Moreover, to stack many layers in a stable manner (i.e. for a secure gradient propagation), a residual connection was utilized (Gross & Wilber 2016; He *et al.* 2016; Wang *et al.* 2018) throughout the generator. Furthermore, to stabilize adversarial training, which is known to be difficult because the objectives of the two components head in opposite directions, spectral normalization (Horn & Johnson 2012) was applied to every convolutional weight in the generator (Zhang *et al.* 2019) and discriminator (Miyato *et al.* 2018). The detailed structures of the residual blocks are presented at the bottom of figure 3.

Second, the discriminator is trained to classify half the data (ground truth, i.e. FFT-based cross-correlated flow field; denoted by t) as real data and the other half ($G(c)$) as fake data on patches (Isola *et al.* 2017) through five convolution layers. This is achieved by minimizing the given loss function $\mathcal{L}_D$ (Mao *et al.* 2017) ($\mathbb{E}$, expectation; p, probability distribution):

$$\mathcal{L}_D = \tfrac{1}{2}\mathbb{E}_{t,c\sim p_{data}(t,c)}[(D(t|c)-1)^2] + \tfrac{1}{2}\mathbb{E}_{c\sim p_{data}(c)}[(D(G(c)|c))^2]. \tag{3.1}$$

Thus, the discriminator strives to make $D(t|c)$ near 1 and $D(G(c)|c)$ near 0. In contrast, the generator loss function $\mathcal{L}_G$ is given by

$$\mathcal{L}_G = \frac{1}{2}\mathbb{E}_{c\sim p_{data}(c)}[(D(G(c)|c)-1)^2] + \lambda\sum_{l=1}^{L}\frac{1}{N_l}[\|D^{(l)}(G(c)|c) - D^{(l)}(t|c)\|_1], \tag{3.2}$$

where the feature-matching loss (Johnson, Alahi & Fei-Fei 2016) is additionally introduced to match the intermediate representations recognized by D ($\lambda = 100$; the number of D inner layers $L = 4$; N_l is the number of elements in the lth layer). The feature-matching loss stabilizes GAN learning and has a good effect on style transfer (Johnson *et al.* 2016). Furthermore, without spectral normalization or the feature-matching term, or a small λ, an equilibrium between G and D could not be reached. By minimizing (3.2), the generator tries to make $D(G(c)|c)$ near 1 and $D^{(l)}(G(c)|c)$ approach $D^{(l)}(t|c)$, creating more realistic sharp flow fields as the training proceeds. Therefore, when adversarial learning is stably and sufficiently progressed, and if G and D have sufficient capacity, the probability distribution of the prediction approaches that of the ground truth ($p_{G(c)} \to p_t$). Finally, an equilibrium is reached, as shown at the top of figure 4(*a*) (at near 500 epochs; at this point, the discriminator cannot differentiate between $G(c)$ and t, that is, $D(G(c)|c) = D(t|c) = 0.5$. Both values cannot improve as $p_{G(c)} = p_t$ (Goodfellow *et al.* 2014)). Thus, using our spectral-norm-based residual conditional GAN architecture, a very stable high-resolution image-to-image translation was possible with a superior generation quality. The developed GAN translator can also be applied to other domain sizes by simply adding or removing

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

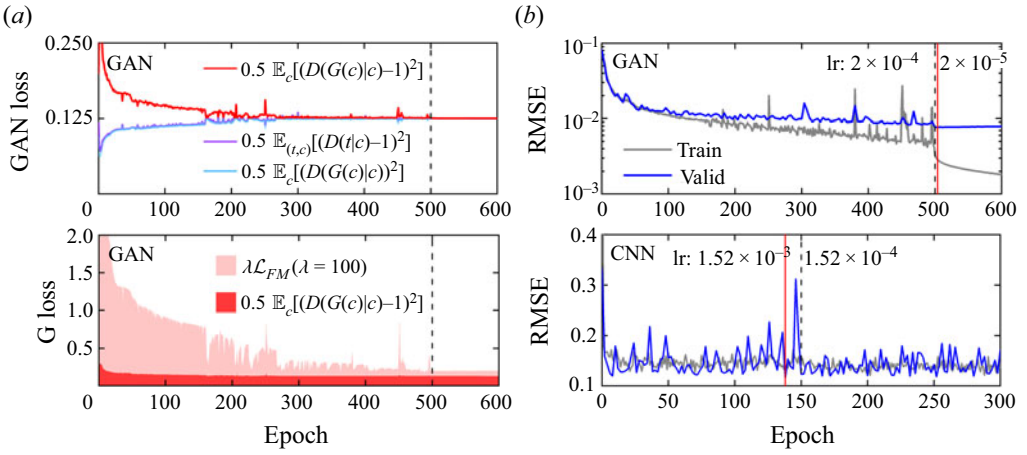


Figure 4. (*a*) Profile of the GAN train loss (top) and generator loss (bottom). (*b*) RMSE profile of the GAN translator (top) and CNN regressor (bottom) for the training (grey) and validation (blue) datasets.

the front or upscaling residuals and adjusting the number of intermediate residual blocks, as shown in figure 3(*a*). More details about the model and loss function can be found in Appendix A (generator), Appendix B (discriminator) or the referenced articles. They are omitted here for conciseness.

Third, the thrust-predicting CNN model was trained as shown in figure 3(*b*), and its unit structure is presented at the bottom right corner of the figure. It is a well-known ResNet-50 feature recognizer (Gross & Wilber 2016; He *et al.* 2016). The extracted features are regressed into a scalar value, the instantaneous thrust, by minimizing the following MSE $\mathcal{L}_F$ (F, ResNet-50, T, experimentally measured thrust using a load cell):

$$\mathcal{L}_F = (F(t) - T)^2. \tag{3.3}$$

Note that, although ResNet is not the most recent model in image recognition, it still attains very good performance (Kolesnikov *et al.* 2019) and is based on a simple and intuitive architecture.

In figure 4(*a*), the profiles of the GAN train and generator losses are shown in the top and bottom parts, respectively. The figure indicates that the model was successfully trained and reached an equilibrium with a near-zero feature-matching loss (0.068×10^{-2} at epoch 504). Figure 4(*b*) presents the RMSE profiles of the two AI models (top: GAN translator, bottom: CNN regressor) for the training (grey) and validation (blue) datasets. The RMSE and R^2 accuracies for the flow-field prediction were calculated after masking (excluding) the rotor area in this study (see the black masked area presented in figure 1, including the rotor). The red vertical line in each graph indicates the validation epoch number, at which the final performance of the model was evaluated using the test dataset.

Both models were trained from scratch with PyTorch using dual TITAN RTX GPUs, and the details of the training hyperparameters are summarized in table 2. The training data were pre-augmented by applying a horizontal flip (thus, 700×2 pairs) to both models. For the GAN translator, the number of D training steps was four, whereas that of G training steps was one for each batch iteration (better results were obtained when the D train step was larger than the G train step, as reported in Arjovsky, Chintala & Bottou 2017). Moreover, the AdamW optimizer (Loshchilov & Hutter 2017) showed a better performance

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

	GAN translator	CNN regressor
Total epochs	600	300
Batch size	2	20
Learning rate	2×10^{-4} (decayed by 1/10 at 500 epochs)	1.52×10^{-3} (Snoek, Larochelle & Adams 2012) (decayed by 1/10 at 150 epochs)
Weight decay coefficient	0.05	5.92×10^{-8} (Snoek *et al.* 2012)
Optimizer	AdamW (Loshchilov & Hutter 2017) ($\beta_1 = 0.5$)	Adam (Kingma & Ba 2014) ($\beta_1 = 0.9$)
Validated point (minimum valid RMSE)	Epoch, 504; RMSE, 0.0077; R^2, 97.03 % (validation dataset)	Epoch, 138; RMSE, 0.1156; R^2, 92.81 % (validation dataset)
Training time	121 h	2 h 30 min

Table 2. Information regarding the data and training of the two developed AI models.

with our GAN model than that of Adam optimizer (Kingma & Ba 2014). (See Appendix D for more details on the computational set-ups.)

4. Results and discussion

In this section, the performance of the developed AI model is demonstrated using the test dataset (namely, the data not used in any training or validation process). In figure 5(*a*), the particle image sequence inputted to the GAN translator (top left two images) and predicted instantaneous flow fields are shown (top right, v; left bottom, u; and right bottom, ω_z). In each field, the ground truth and prediction are displayed on the left and right sides, respectively, with the R^2 accuracy written at the top-right corner. One randomly selected datum out of a total of 270 test data is shown in the figure after the non-dimensionalization of the values and axes (V_{tip}, rotor tip velocity; R_r, rotor radius (38 mm)). In the first particle image, the employed Cartesian coordinates are marked with the tip path plane (TPP).

As shown in the figure, the prediction is so accurate that the ground truth (t) and AI generation ($G(c)$) are almost indistinguishable, as was the goal of the GAN. In figure 6(*a*), the R^2 accuracy of the instantaneous flow-field prediction with respect to RPM (bottom x-axis) and Reynolds number based on the rotor tip speed (Re_{tip}; top x-axis) is plotted (each scatter point denotes the mean of the accuracy for the 27 test samples). ($Re_{tip} = V_{tip}c/\nu$, where c and ν denote the chord length of the rotor blade (0.016 m) and kinematic viscosity of air, respectively.) The RPM-wise (or Re_{tip}-wise) mean accuracy was 99.67 % for v, 97.07 % for u and 94.90 % for ω_z (their average is 97.21 %). This result demonstrated the capability of the developed GAN translator. Note that the accuracies and errors in this study were calculated after excluding the rotor area, as mentioned earlier.

In figure 5(*b*), using the predicted instantaneous flow fields, a statistical turbulent flow field is constructed (top left, $\langle v \rangle / V_{tip}$; top right, v'_{rms}/V_{tip}; middle left, $\langle v \rangle / V_{tip}$; middle right, u'_{rms}/V_{tip}; bottom left, $\langle \omega_z \rangle / (V_{tip}/R_r)$; and bottom right, $\omega'_{z,rms}/(V_{tip}/R_r)$; $\langle\,\rangle$: time average; apostrophe r.m.s., r.m.s., velocity fluctuation; number of samples, 27; RPM, 8500). Figure 6(*b*) depicts the R^2 accuracy of the mean and r.m.s. flow fields for all RPMs (for all Re_{tip}s). Because the instantaneous flow-field prediction was accurate, the mean and r.m.s. flow fields also showed a high accuracy of approximately 99 %, even in a region with high turbulence (i.e. a large r.m.s. region). Note that the r.m.s. velocity fluctuations were more prominent in the x-direction than they were in the y-direction (compare the contours of u'_{rms}/V_{tip} with v'_{rms}/V_{tip}) along the trajectory of the rotor tip

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

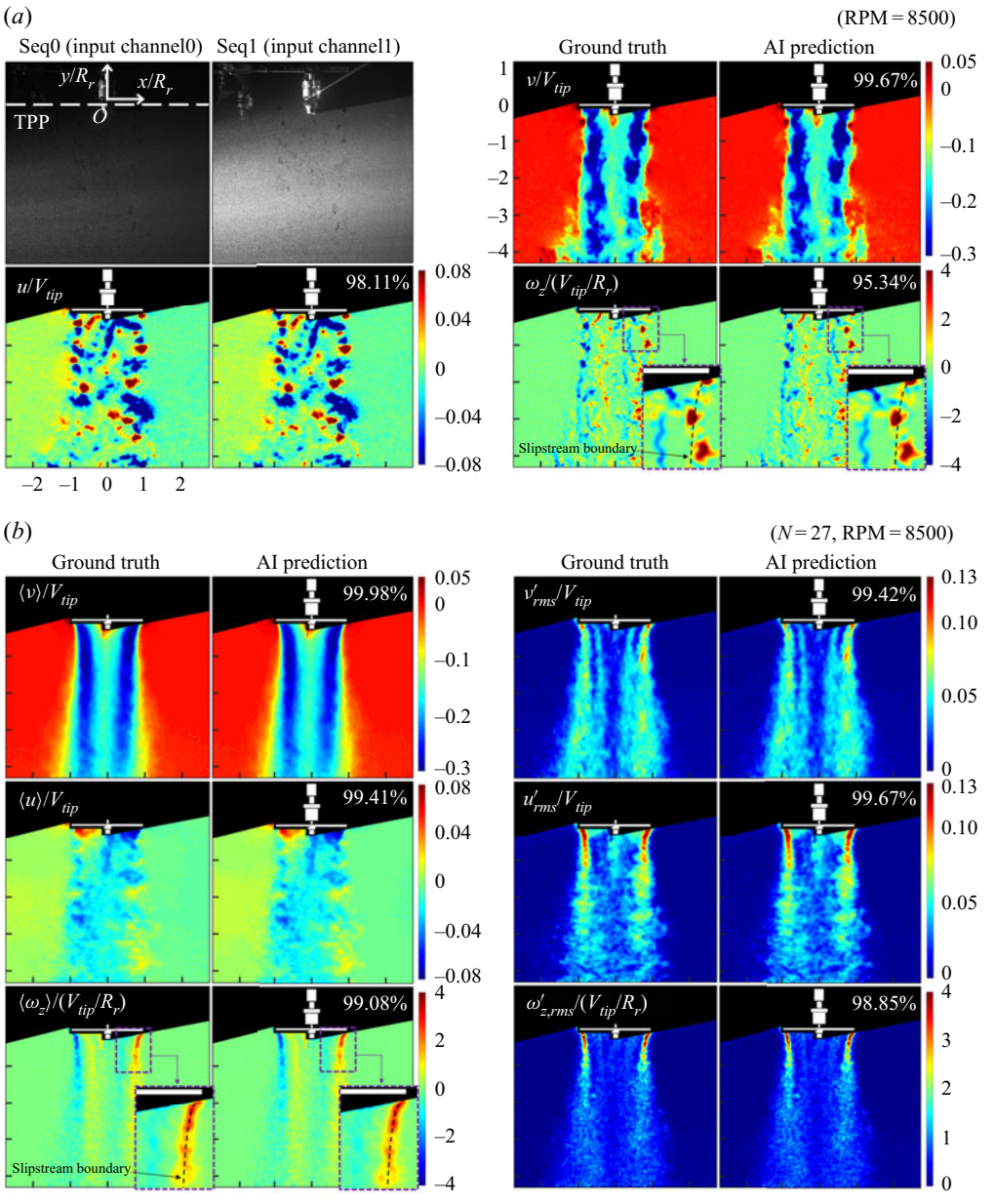


Figure 5. (*a*) Predicted instantaneous flow field using the developed GAN translator. (*b*) Statistical turbulent flow field obtained from the 27 predicted instantaneous flow fields at 8500 RPM.

vortices close to the TPP. This indicates that the tip vortices 'wander' mainly in the direction normal to their trajectory. The mean trajectory of the tip vortices is called the slipstream boundary and is shown as a black dashed curve in the enlarged inset of the $\langle\omega_z\rangle/(V_{tip}/R_r)$ contour. To demonstrate the wandering motion of the tip vortices, the slipstream boundary was overlapped on the normalized instantaneous vorticity contour (see the enlarged inset in the lower right panel of figure 5*a*). As shown in the figure, the rotor tip vortices convected downstream, deviating from side to side with respect

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

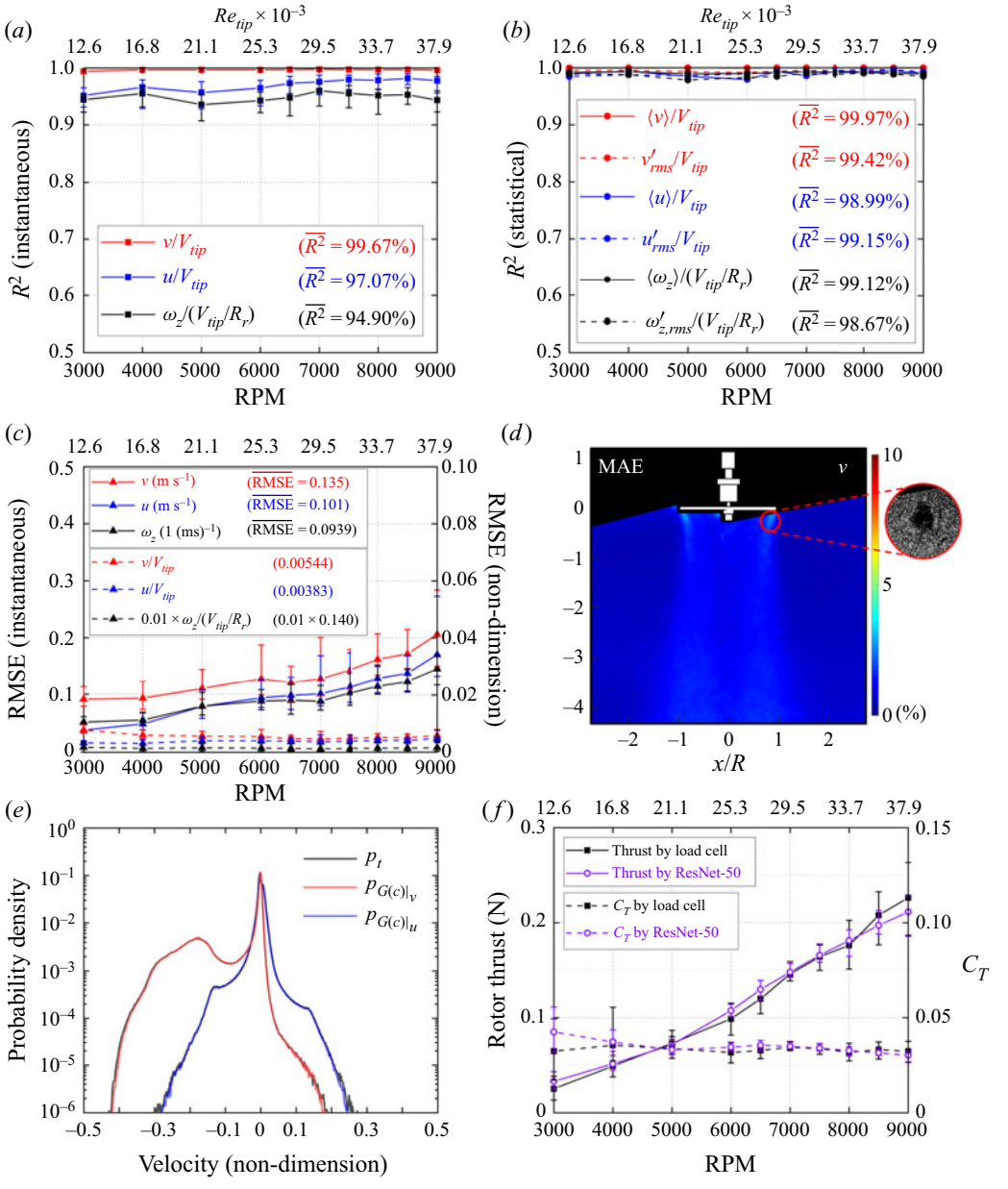


Figure 6. Prediction accuracies for (*a*) the instantaneous flow field and (*b*) statistical turbulent flow field, versus the rotor RPM (bottom *x*-axis) and Reynolds number based on the rotor tip speed (Re_{tip}; top *x*-axis). Note that non-dimensionalization does not affect the accuracy. (*c*) RPM-wise (or Re_{tip}-wise) RMSE (solid lines; left *y*-axis) and non-dimensionalized RMSE (dashed lines; right *y*-axis) for the instantaneous flow field. (*d*) MAE map for the streamwise velocity v on a y/R–x/R plane. It was normalized with respect to the v range. (*e*) Probability density of the instantaneous velocity (black, ground truth; red, predicted v; and blue, predicted u; c and t, entire test data). (*f*) Results of the instantaneous rotor thrust prediction using the CNN regressor (black, ground truth measured by the load cell; violet, AI prediction). Solid and dashed lines denote rotor thrust (left *y*-axis) and thrust coefficient C_T (right *y*-axis), respectively, according to the rotor RPM (bottom *x*-axis) and Re_{tip} (top *x*-axis).

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

to the slipstream boundary. It has been suggested that vortex wandering is induced by inflow unsteadiness, shear layer turbulence and cooperative instabilities (Mula *et al.* 2013). Considering that wake geometry plays an essential role when assessing the aerodynamic performance, such as rotor efficiency and blade loads (Landgrebe 1972; Leishman, Baker & Coyne 1996; Leishman 1998; Lee *et al.* 2021), vortex wandering hinders the accurate prediction of drone performance because it makes it difficult to predict the rotor wake boundary. Thus, vortex wandering requires further investigation. The accurate prediction method presented in this study will pave the way for further detailed studies on the underlying mechanism of vortex wandering.

To assess the quality of the flow-field prediction, the RMSE of the instantaneous flow-field prediction and MAE map for the y-directional (streamwise) velocity v are presented in figures 6(c) and 6(d), respectively (the error map was normalized with respect to the v range). In figure 6(c), the RMSE and its non-dimensionalized value are shown as solid and dashed lines on the left and right y-axes, respectively, according to the RPM (bottom x-axis) and Re_{tip} (top x-axis). As shown in the figure, the RPM-wise (or Re_{tip}-wise) mean RMSE is 0.135 m s^{-1} for v (0.00544 for v/V_{tip}), 0.101 m s^{-1} for u (0.00383 for u/V_{tip}) and 0.0939 ms for ω_z (0.140 for $\omega_z/(V_{tip}/R)$). As the RPM increased, the error also increased (solid lines), which is a mathematically natural behaviour because the magnitudes of v, u and ω_z increase with the RPM. As shown in the error map presented in figure 6(d), the error was nearly zero over the entire domain, except for the area along the trajectory of the tip vortices near the rotor tip (red-circled area; the maximum error was 2.91 %). The relatively large MAE near the rotor tip is due to a locally low particle density. As shown in the enlarged inset of figure 6(d), there is a region where few particles are observed. This region is known to be formed as the centrifugal force induced by the rotor tip vortex causes the seeding particles to move radially outward from the vortex core (Leishman 1998). Because there are few particles in this black void area, CC-based PIV predicts the flow using a series of image processing techniques, such as subpixel interpolation. Because the AI learned by looking at the results obtained in this manner, the accuracy in that area was inevitably lower than that in other areas that directly matched the flow field from the particle information. In addition, because the ground truth itself included errors due to interpolation, it was natural to observe larger errors in AI prediction in that area. However, even if the error was larger, it was less than 3 %, which is acceptable. Finally, in addition to the RMSE and error map, the MAE, peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) (Wang *et al.* 2004) were calculated. The RPM-wise mean MAE values were 0.0693 m s^{-1} for v, 0.0499 m s^{-1} for u and 0.0444 (ms)$^{-1}$ for ω_z (non-dimensionalized MAE values were 0.00279 for v, 0.00194 for u and 0.0683 for ω_z). The PSNR values were 46.27 dB for v, 50.02 dB for u and 51.41 dB for ω_z (their average was 49.24 dB). The values of SSIM were 98.86 % for v, 99.47 % for u and 98.92 % for ω_z (an average of 99.08 %). (See table 3 in Appendix C for the tabulated data.)

In figure 6(e), the probability distributions of the instantaneous velocity for prediction ($p_{G(c)}$) and ground-truth (p_t) values are plotted on a probability density–velocity plane (black, ground truth; red, predicted v; and blue, predicted u; c and t, entire test data; rotor area excluded). As shown in the figure, the distributions of the predictions faithfully follow those of the ground truths, even for the test data, underlining the good generalization ability of the model with minimal overfitting.

From the predicted instantaneous flow field ($G(c)$), instantaneous rotor thrust was predicted using the proposed CNN regressor, and the results are presented in figure 6(f). The black and violet lines denote the measured and predicted thrusts, respectively, with

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

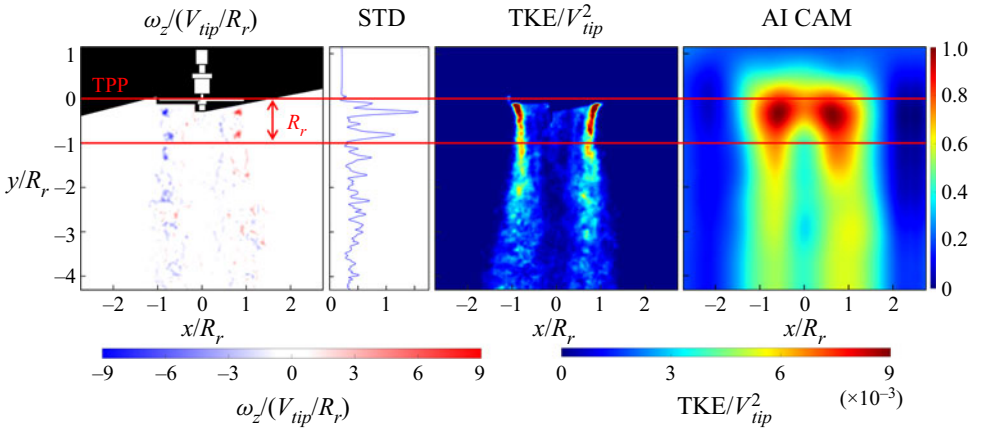


Figure 7. From left to right: input instantaneous vorticity, y-directional standard deviation of absolute vorticity, turbulent kinetic energy (TKE/V_{tip}^2) and CAM at 7500 RPM.

respect to the rotor RPM (bottom x-axis) and Re_{tip} (top x-axis). The solid and dashed lines represent the rotor thrust (left y-axis) and thrust coefficient C_T (right y-axis), respectively. The value of C_T was calculated as $T/(\rho A V_{tip}^2)$, where T, ρ and A denote the rotor thrust, density of air and rotor disk area (πR_r^2), respectively. The R^2 accuracy is 94.57 %, suggesting that instantaneous rotor thrust can be accurately predicted using deep learning from only the flow-field information without any additional measuring equipment. Note that when the rotor thrust is predicted with additional experimental data for RPM setpoints of 3500, 4500 and 5500, accuracy is 92.71 %, 96.01 % and 97.29 %, respectively.

In figure 7, from left to right, the input vorticity field, its y-directional standard deviation ($\sigma_y(|\omega_z/(V_{tip}/R_r)|)$), which reveals the relative strength of vorticity according to y, a contour of turbulent kinetic energy (TKE; normalized with respect to V_{tip}^2) and a class activation map (CAM) (Zhou *et al.* 2016) are presented. A TKE contour demonstrates a long growth of TKE along the trajectory of the rotor tip vortices, called the slipstream boundary, where it appears high in the area from the TPP down to one rotor radius (R_r) and gradually weakens downstream. This result indicates that when the tip vortices formed by the rotor rotation are periodically convected downstream, they maintain a distinct shape downstream to $y/R_r = -1$ but are mixed and dissipated later. This result can also be confirmed with the instantaneous vorticity contours. In the region where the tip vortices remain distinct (from TPP down to $1R_r$), the streamwise velocity is accelerated because the wake-induced velocity exists within the slipstream boundary. Subsequently, the wake region begins to decelerate as the wake spreads widely and decays owing to the viscous effect.

As shown in figure 7, CAM is highly activated within the $1R_r$ section under TPP (inside the area bounded by two horizontal red lines). CAM was used to investigate where the CNN model was looking when making predictions. Thus, the AI model examined this area and predicted the thrust. In this $1R_r$ region, as shown in the figure, energy is continuously provided by the rotor rotation and dominant wake structures are maintained. After this region (toward $-y$), the energy supplied by the rotor is consumed and the momentum is dissipated (wake structures are demolished). The thrust is proportional to the energy supplied by the rotor (according to momentum theory). The AI model examined the area where the energy was provided and made predictions, thus the model succeeded in

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

capturing the underlying physics by itself and was trained properly. The only information provided to the model was the input instantaneous flow field and corresponding thrust value.

The computational factors for time and memory consumption are presented in Appendix D along with a comparison against the conventional CC-based PIV method. Specifically, the prediction took only 12.5 s for the GAN translator (nearly 600 times faster than the CC PIV) and 2.1 s for the CNN regressor when using 270 test datasets. The two AI models presented in this study (GAN translator and CNN regressor) can be utilized separately; therefore, the developed CNN regressor can be applied to a flow field obtained from an existing PIV technique, such as CC and OF, to predict the instantaneous rotor thrust.

5. Conclusions

In this study, a novel deep-learning architecture was proposed to predict the instantaneous flow field and thrust under a single drone rotor from PIV-observed particle images. A real-world complex turbulent flow induced by the high-speed rotation of a rotor (up to 9000 RPM) was examined. Two different deep-learning models were presented, and the main results of this study can be summarized as follows:

(i) For a highly accurate prediction of the instantaneous flow field from high-resolution particle images (2048×2048 pixels), a GAN framework was introduced for the first time instead of a conventional CNN-based encoder–decoder model with a regressive objective. The developed model was a spectral-norm-based residual conditional GAN (SNRGAN), which can conduct highly stable adversarial training and yield high-accuracy predictions. Training hyperparameters optimized for the model were also presented.

(ii) The mean accuracy for the instantaneous flow-field prediction was 97.21 % (R^2), 49.24 dB (PSNR) and 99.08 % (SSIM). The prediction was accurate for all RPM ranges (i.e. 3000–9000). Moreover, the probability distributions of the prediction and ground-truth values were well matched, demonstrating the good generalization ability of the developed GAN model with minimal overfitting.

(iii) For the first time, instantaneous rotor thrust was predicted using a deep CNN model (ResNet-50) from an instantaneous flow field. For the regression task, the last classifier was replaced with a linear regressor, and the MSE cost function was employed. The prediction accuracy was 94.57 %, which highlights the potential of AI methods in drone thrust prediction. Moreover, where the model examined and made predictions, as well as whether the training proceeded properly from a physical standpoint, were discussed. The results confirmed that the trained AI model predicted the thrust by spontaneously recognizing the part that received kinetic energy from the rotor.

(iv) The developed deep-learning models significantly improved the computational speed. The flow-field calculation took only 12.5 s for 270 test datasets, whereas it took 120–125 min using the conventional CC-based PIV technique (nearly 600 times slower compared with the proposed method). For thrust prediction, it only took 2.1 s for the same amount of data. Consequently, with our deep-learning architecture, a high-accuracy and high-speed solution can be achieved even with real-world complex turbulent flow data.

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

Funding. This research was supported by the Basic Science Research Program of the National Research Foundation (NRF) of Korea, funded by the Ministry of Science and ICT (grant no. NRF-2019R1A2C2089114).

Declaration of interests. The authors report no conflict of interest.

Author ORCIDs.
Sehyeok Oh https://orcid.org/0000-0002-3721-3953;
Seungcheol Lee https://orcid.org/0000-0001-6403-3424;
Myeonggyun Son https://orcid.org/0000-0002-7288-4832;
Jooha Kim https://orcid.org/0000-0001-6476-514X;
Hyungson Ki https://orcid.org/0000-0003-3602-2036.

Appendix A. Architecture of the GAN generator

Our GAN generator has a spectral-norm-based residual convolutional encoder–decoder structure, which consists of four stages: a stem stage, front residuals, intermediate residuals and back residuals, as presented in figure 3. Here, c7s1-k denotes a 7×7 convolution with k filters and a stride of 1. Rd-k denotes a downscaling residual block consisting of (c3s2-k, IN, ReLU, c3s1-k, IN) and (c1s2-k, IN), where IN and ReLU represent instance normalization and rectified linear units (Nair & Hinton 2010), respectively. Note that, in this paragraph, the former and latter parentheses indicate the main operations and residual shortcut operations, respectively. Ru-k denotes an upscaling residual block composed of (tc3s2-k, IN, ReLU, c3s1-k, IN) and (tc1s2-k, IN), where tc represents the transposed convolution (output padding = 1). R-k denotes a normal residual block consisting of (c3s1-k, IN, ReLU, c3s1-k, IN) with an identity residual summation ((c1s1-k, IN) if the number of channels changes). In every residual block, the padding was 1 for the main convolutions and 0 for the shortcut convolution. Spectral normalization was applied to all convolutional weights. All biases were off to reduce overfitting. The architecture of the generator is as follows.

(i) Stem stage: c7s1-32 (padding = 3), IN, ReLU.
(ii) Front residuals: Rd-64, Rd-128, Rd-256, Rd-512, Rd-1024.
(iii) Intermediate residuals: R-1024 $\times$ 15.
(iv) Back residuals: Ru-512, Ru-256, R-128, R-64, R-3*.

(*The last R-3 block is (c3s1-3, IN, ReLU, c3s1-3) and (c1s1-3), with the last tanh activation.)

Appendix B. Architecture of the GAN discriminator

Our GAN discriminator is a patchGAN-style (Isola *et al.* 2017) spectral-norm-based encoder. Here, c4s2-k-L denotes a 4×4 convolution with k filters and a stride of 2 (padding = 1), followed by a leaky ReLU (Maas, Hannun & Ng 2013) (negative slope = 0.2 (Radford, Metz & Chintala 2015)). Spectral normalization was applied to all convolutional weights, and all layer normalizations were cancelled accordingly (Miyato *et al.* 2018). All biases were off. The architecture of the discriminator is as follows.

(i) Channel concatenation between $G(c)$ (tensor size, (3, 256, 256)) and c after average pooling (tensor size, (2, 256, 256)): $G(c)|c$ (tensor size, (5, 256, 256)).
(ii) c4s2-64-L, c4s2-128-L, c4s2-256-L, c4s1-512-L, c4s1-1.

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

Objective function	Flow field	MAE	RMSE	R^2 (%)	PSNR	SSIM (%)
MAE	v	0.0753 m s^{-1} (0.00309)	0.148 m s^{-1} (0.00604)	99.59	45.42 dB	98.63
	u	0.0589 m s^{-1} (0.00232)	0.114 m s^{-1} (0.00435)	96.11	48.87 dB	99.31
	ω_z	0.0530 (ms)$^{-1}$ (0.0818)	0.105 (ms)$^{-1}$ (0.158)	93.53	50.37 dB	98.57
	(Mean)	—	—	96.41	48.22 dB	98.84
MSE	v	0.0866 m s^{-1} (0.00359)	0.163 m s^{-1} (0.00662)	99.52	44.58 dB	98.32
	u	0.0773 m s^{-1} (0.00313)	0.137 m s^{-1} (0.00536)	93.52	47.00 dB	98.96
	ω_z	0.0705 (ms)$^{-1}$ (0.110)	0.126 (ms)$^{-1}$ (0.190)	90.63	48.75 dB	97.87
	(Mean)	—	—	94.56	46.78 dB	98.38
Learnable function (Discriminator; Ours)	v	0.0693 m s^{-1} (0.00279)	0.135 m s^{-1} (0.00544)	99.67	46.27 dB	98.86
	u	0.0499 m s^{-1} (0.00194)	0.101 m s^{-1} (0.00383)	97.07	50.02 dB	99.47
	ω_z	0.0444 (ms)$^{-1}$ (0.0683)	0.0939 (ms)$^{-1}$ (0.140)	94.90	51.41 dB	98.92
	(Mean)	—	—	97.21	49.24 dB	99.08

Table 3. Performance comparison based on objective function of flow-field generator using 270 test datasets. The structure of the generator was fixed to that described in Appendix A. Numbers in parentheses in the third and fourth columns were non-dimensionalized errors.

Appendix C. Performance comparison based on objective function of flow-field generator

Tests were conducted by changing the objective function of the flow-field generator to determine the effectiveness of the employed GAN architecture compared with a normal CNN-based encoder–decoder structure with a regressive objective (that is, training without the discriminator having the structure of the flow-field generator presented in Appendix A fixed). Two conventional regressive objective functions, namely, MAE and MSE, were tested, whose results are presented in table 3. A total of 270 test datasets were used as input, and the MAE, RMSE, R^2, PSNR and SSIM metrics were assessed. As previously mentioned, our SNRGAN model attained the best performance in terms of these five metrics. Because all three models shared the same generator structure, the computational costs incurred for the prediction were the same (the discriminator of the GAN did not participate in the prediction).

For a visual comparison of the generation quality, figure 8 presents, from left to right, the generation results obtained from the objective functions of MAE, MSE, discriminator (SNRGAN) and ground truth, respectively. Non-dimensionalized v, u and ω_z maps are shown from top to bottom, respectively. The RPM is 3000, and the domain of each figure corresponds to the square area from -1 to 1 on the x/R_r-axis and -2 to 0 on the y/R_r-axis in figure 5(a). The R^2, PSNR and SSIM quality assessment metrics measured in the given square area are provided sequentially below each flow-field prediction map. Because of the deep residual layers, the MAE and MSE objectives demonstrate competitive results in terms of R^2, PSNR and SSIM. However, areas with blurring or missing details are often observed (circled areas), which reveals the innate limitation of the regressive cost functions. On the other hand, our GAN model obtains high-quality predictions that are

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

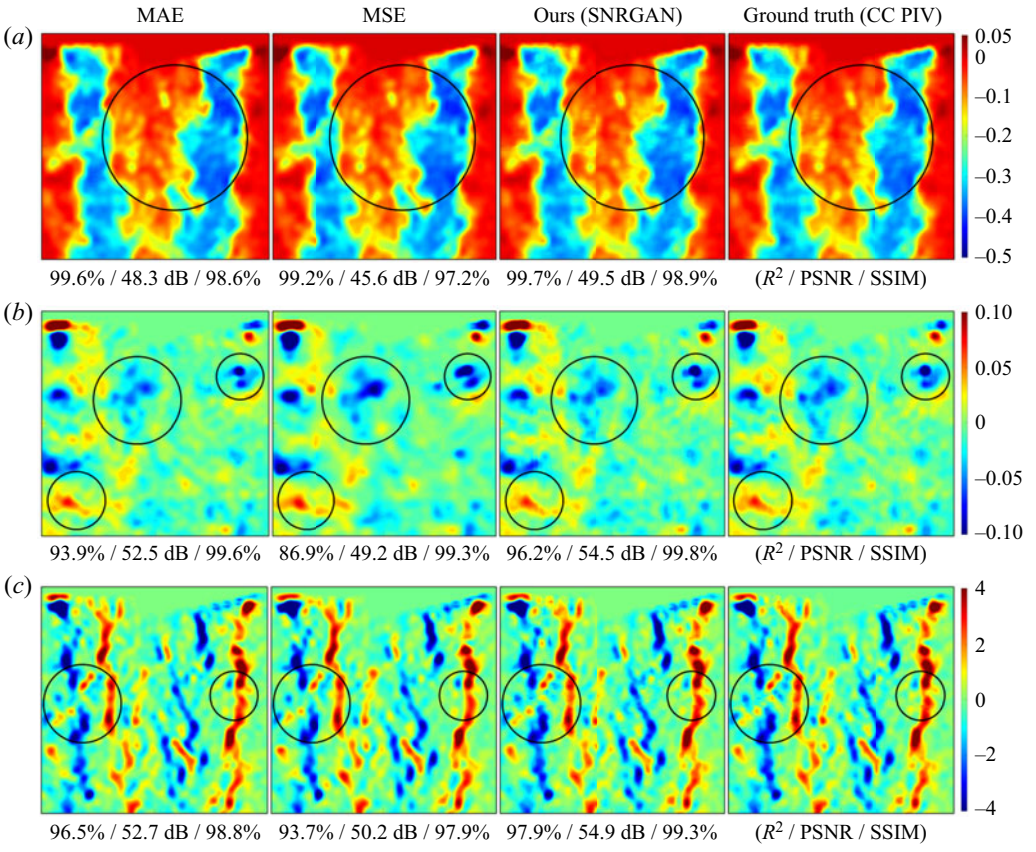


Figure 8. Visual comparison of flow-field prediction results based on objective functions (MAE, MSE, discriminator and ground truth from left to right, respectively), with a fixed generator structure. The non-dimensionalized v, u and ω_z maps are shown from (*a*) to (*c*), respectively, at 3000 RPM.

clear and sharp, with abundant details for the flow field. The metrics for SNRGAN were the best as well. Note that, as presented in the results for the MAE and MSE objectives in figure 8, the R^2, PSNR and SSIM assessment metrics do not seem to fully reflect the true perceptual similarity (the calculated metrics are still good despite the missing sharp details), which has been actively investigated in the field of computer vision (Heusel *et al.* 2017; Ledig *et al.* 2017; Blau & Michaeli 2018; Zhang *et al.* 2018; Gu *et al.* 2020).

Appendix D. Computational cost and comparison with conventional CC-based PIV method

The computational cost was measured using conventional CC-based PIV techniques (in-house software and commercial software (Dantec DynamicStudio)) and deep-learning models (GAN translator and CNN regressor); the results are presented in table 4. The tests were conducted using 270 test datasets (see table 1 for the detailed data configuration). CC-based PIV techniques were performed on an Intel® i5-7500 CPU (3.4 GHz) desktop computer, including the extraction of velocities, data smoothing and vorticity calculations. The two deep-learning models were tested on an AMD EPYC™ 7282 CPU (2.8 GHz) computer aided by four Nvidia RTX A5000 GPUs using PyTorch. Each of these two sets of computational specifications represents a computer environment commonly used

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

Methods			Prediction time	CPU memory (RAM)	GPU memory
Flow-field prediction	CC PIV	In-house software	124.5 min	15 MB	—
		Commercial software (Dantec Dynamic Studio)	120.9 min	1.3 GB	—
	AI (SNRGAN)		12.5 s (node batch size: 12)	15.9 GB	12.7 GB/GPU
Thrust prediction	AI (ResNet-50)		2.1 s (node batch size: 40)	9.8 GB	2.3 GB/GPU

Table 4. Comparison of computation factors using 270 test datasets.

in conventional PIV and deep learning. In the case of AI GPU computing, time and memory consumption were measured using GPU distributed data parallelism (DDP). In addition, to accelerate data loading, two subprocesses were utilized per worker process, with compute unified device architecture (CUDA)-pinned memory prefetching and GPU asynchronous copy. Note that this set-up was used for training as well. As shown in table 4, the AI computation was much faster (approximately 600 times) than the CC-based PIV techniques because of the DDP-based general-purpose computing on graphics processing units (GPGPU) computing, which is well optimized for a neural network. Although AIs require a larger memory, it is reasonable from the point of view of an actual user. In the case of thrust prediction, the prediction time was only 2.1 s, as presented in the table, underlining the high potential of deep learning in real-time thrust prediction applications. One could also consider employing a quantization technique for AI models to decrease the computational cost. Memory usage is known to decrease by a factor of four (float32-to-int8), with a 2–4 times faster computational speed, when typical float32 tensors are replaced by 8-bit integer-type tensors (lower bitwidths) (PyTorch 2021).

Appendix E. Test on a different rotor model and toward generalizations

To demonstrate the generalization ability of the AI model for different rotor geometries, the proposed architecture was additionally tested using a different configuration of the drone rotor operating at five Reynolds numbers. Figure 9(*a*) presents two rotor models – the left one is the rotor used routinely in this study (having a flat-bottomed airfoil; denoted by rotor A) and the right one is the new rotor introduced to test the AI models (having a symmetric airfoil; denoted by rotor B). Both rotor models have untwisted rectangular blades, and their rotor radii, blade pitch angles and chord lengths are reported in the figures. RPM setpoints for the new rotor were 2500, 4000, 5000, 7000 and 12 500 (corresponding Re_{tip}s were 10 389, 16 622, 20 778, 29 089 and 51 944, respectively).

Using one hundred rotor B data (i.e. particle image sequences taken from rotor B, flow fields calculated from them using CC and thrust forces corresponding to each flow field), the developed GAN translator and CNN regressor were tested. Subsequently, R^2 accuracies of 96.49% and 96.79% were obtained for the flow-field prediction and thrust prediction, respectively. In figure 9(*c*), the particle image sequence inputted to the GAN translator (top left two images) and predicted instantaneous flow fields with their ground truths are shown (top right: v, bottom left: u and bottom right: ω_z) after non-dimensionalization. As expected, the differences between the prediction results and ground truths were

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

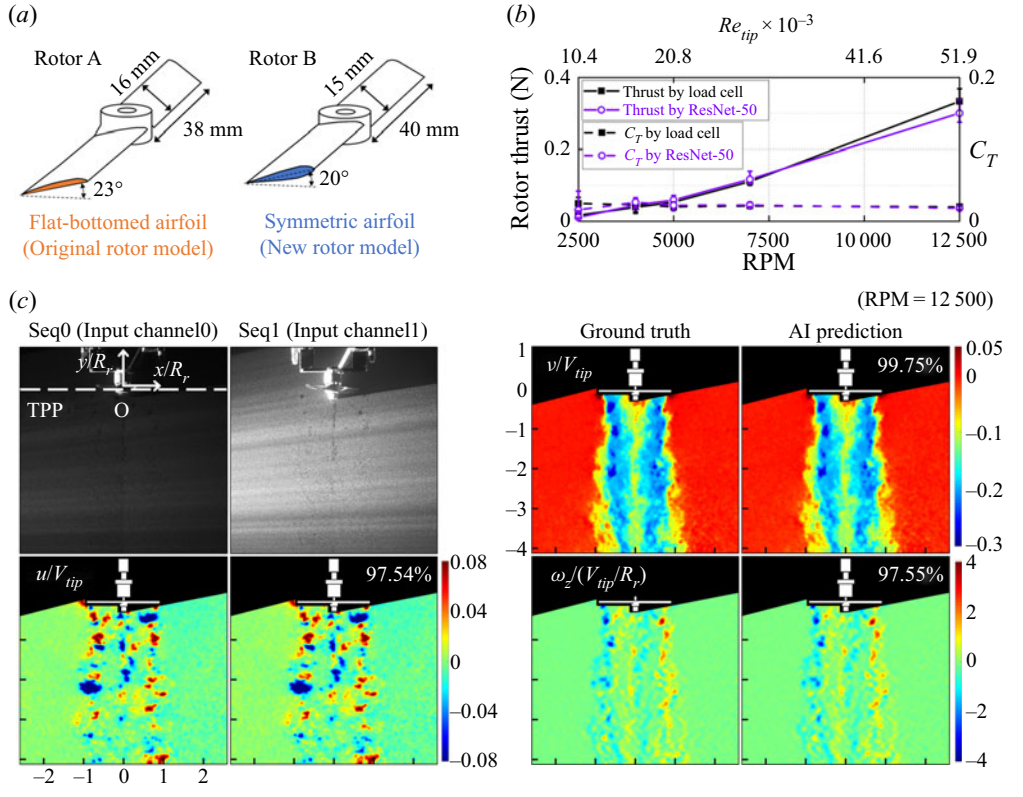


Figure 9. (*a*) Rotor models (left, rotor A – the original rotor model; right, rotor B – a new rotor model). (*b*) Thrust prediction result for rotor B using the trained CNN regressor. (*c*) Instantaneous flow-field prediction result for rotor B using the trained GAN translator.

Flow field	MAE	RMSE	R^2 (%)	PSNR	SSIM (%)
v	0.0599 m s^{-1} (0.00252)	0.114 m s^{-1} (0.00478)	99.46	48.29 dB	98.96
u	0.0419 m s^{-1} (0.00170)	0.0780 m s^{-1} (0.00314)	93.68	52.74 dB	99.48
ω_z	0.0240 (ms)$^{-1}$ (0.0404)	0.0430 (ms)$^{-1}$ (0.0714)	96.33	58.61 dB	99.56
(Mean)	—	—	96.49	53.21 dB	99.34

Table 5. Errors and accuracies of flow-field prediction under a new rotor (rotor B in figure 9*a*). Numbers in parentheses are non-dimensionalized errors.

minimal. The prediction metrics of MAE, RMSE, R^2, PSNR and SSIM are listed in table 5. Figure 9(*b*) presents the thrust prediction result, which showed good accuracy even with a smaller or larger Reynolds number than that used for training.

Note that the predictions for rotor B were safely conducted, as its data had a probability distribution similar to that of the existing rotor. (The geometric type of blade, rotor radius, blade pitch angle and chord length of rotor B were not significantly different from those of rotor A.) However, if a rotor with a radius of 150 mm is used (nearly 4 times bigger than rotor A), machine learning techniques, such as fine tuning or transfer learning, might

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

be required to effectively broaden visual field of AI models or to quickly adapt them to new targets (TensorFlow 2021). Such as example in computer vision is applying the prior knowledge obtained from learning to recognize cars to the task of recognizing buses. Specifically, it entails taking an AI model that has already been trained on some data and training it a little further using new data (and possibly with the original data). With this approach, it is possible to effectively broaden and deepen the knowledge of AI, and move toward generalization.

References

Adrian, L., Adrian, R.J. & Westerweel, J. 2011 *Particle Image Velocimetry*. Cambridge University Press.

Arjovsky, M., Chintala, S. & Bottou, L. 2017 Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pp. 214–223.

Blau, Y. & Michaeli, T. 2018 The perception-distortion tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6228–6237.

Bukka, S.R., Gupta, R., Magee, A.R. & Jaiman, R.K. 2021 Assessment of unsteady flow predictions using hybrid deep learning based reduced-order models. *Phys. Fluids* **33**, 013601.

Cai, S., Liang, J., Gao, Q., Xu, C. & Wei, R. 2019*a* Particle image velocimetry based on a deep learning motion estimator. *IEEE Trans. Instrum. Meas.* **69**, 3538–3554.

Cai, S., Zhou, S., Xu, C. & Gao, Q. 2019*b* Dense motion estimation of particle images via a convolutional neural network. *Exp. Fluids* **60**, 1–16.

Chae, S., Lee, S., Kim, J. & Lee, J.H. 2019 Adaptive-passive control of flow over a sphere for drag reduction. *Phys. Fluids* **31**, 015107.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. 2014 Generative adversarial nets. *Adv. Neural Inform. Proc. Syst.* **27**, 2672–2680.

Gross, S. & Wilber, M. 2016 Training and Investigating Residual Nets. Available at: http://torch.ch/blog/2016/02/04/resnets.html

Gu, J., Cai, H., Chen, H., Ye, X., Ren, J. & Dong, C. 2020 Image quality assessment for perceptual image restoration: a new dataset, benchmark and metric. arXiv:2011.15002

He, K., Zhang, X., Ren, S. & Sun, J. 2016 Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. & Hochreiter, S. 2017 Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Adv. Neural Inform. Proc. Syst.* **30**.

Hinton, G.E. & Salakhutdinov, R.R. 2006 Reducing the dimensionality of data with neural networks. *Science* **313**, 504–507.

Horn, R.A. & Johnson, C.R. 2012 *Matrix Analysis*. Cambridge University Press.

Ioffe, S. & Szegedy, C. 2015 Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167.

Isola, P., Zhu, J.-Y., Zhou, T. & Efros, A.A. 2017 Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1125–1134.

Johnson, J., Alahi, A. & Fei-Fei, L. 2016 Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pp. 694–711.

Kingma, D.P. & Ba, J.L. 2014. Adam: a method for stochastic optimization. arXiv:1412.6980

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S. & Houlsby, N. 2019. Big transfer (BiT): General visual representation learning. arXiv:1912.11370

Landgrebe, A.J. 1972 The wake geometry of a hovering helicopter rotor and its influence on rotor performance. *J. Am. Helicopter Soc.* **17**, 3–15.

Lecun, Y., Bottou, L., Bengio, Y. & Haffner, P. 1998 Gradient-based learning applied to document recognition. *Proc. IEEE* **86**, 2278–2324.

Ledig, C., Theis, L., Huszār, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J. & Wang, Z. 2017 Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer visIon and Pattern Recognition*, pp. 4681–4690.

Lee, S., Chae, S., Woo, S.Y., Jang, J. & Kim, J. 2021 Effects of rotor–rotor interaction on the wake structure and thrust generation of a quadrotor unmanned aerial vehicle. *IEEE Access* **9**, 85995–86016.

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press

LEE, Y., YANG, H. & YIN, Z. 2017 PIV-DCNN: cascaded deep convolutional neural networks for particle image velocimetry. *Exp. Fluids* **58**, 171.

LEISHMAN, J.G. 1998 Measurements of the aperiodic wake of a hovering rotor. *Exp. Fluids* **25**, 352–361.

LEISHMAN, J.G., BAKER, A. & COYNE, A. 1996 Measurements of rotor tip vortices using three-component laser Doppler velocimetry. *J. Am. Helicopter Soc.* **41**, 342–353.

LOSHCHILOV, I. & HUTTER, F. 2017 Decoupled weight decay regularization. arXiv:1711.05101

MAAS, A.L., HANNUN, A.Y. & NG, A.Y. 2013 Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning*, p. 3.

MAO, X., LI, Q., XIE, H., LAU, R.Y., WANG, Z. & PAUL SMOLLEY, S. 2017 Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2794–2802.

MIRZA, M. & OSINDERO, S. 2014 Conditional generative adversarial nets. arXiv:1411.1784

MIYANAWALA, T.P. & JAIMAN, R.K. 2017 An efficient deep learning technique for the Navier–Stokes equations: Application to unsteady wake flow dynamics. arXiv:1710.09099

MIYANAWALA, T.P. & JAIMAN, R.K. 2019 A hybrid data-driven deep learning technique for fluid-structure interaction. In *ASME 2019 38th International Conference on Ocean, Offshore and Arctic Engineering, Volume 2: CFD and FSI*.

MIYATO, T., KATAOKA, T., KOYAMA, M. & YOSHIDA, Y. 2018 Spectral normalization for generative adversarial networks. arXiv:1802.05957

MULA, S.M., STEPHENSON, J.H., TINNEY, C.E. & SIROHI, J. 2013 Dynamical characteristics of the tip vortex from a four-bladed rotor in hover. *Exp. Fluids* **54**, 1–14.

NAIR, V. & HINTON, G.E. 2010 Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*.

PATHAK, D., KRAHENBUHL, P., DONAHUE, J., DARRELL, T. & EFROS, A.A. 2016 Context encoders: feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536–2544.

PYTORCH 2021 Quantization. Available at: https://pytorch.org/docs/stable/quantization.html

RABAULT, J., KOLAAS, J. & JENSEN, A. 2017 Performing particle image velocimetry using artificial neural networks: a proof-of-concept. *Meas. Sci. Technol.* **28**, 125301.

RADFORD, A., METZ, L. & CHINTALA, S. 2015 Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv:1511.06434

SNOEK, J., LAROCHELLE, H. & ADAMS, R.P. 2012 Practical bayesian optimization of machine learning algorithms. arXiv:1206.2944

TENSORFLOW 2021 Transfer Learning and Fine-tuning. Available at: https://www.tensorflow.org/guide/keras/transfer_learning

ULYANOV, D., VEDALDI, A. & LEMPITSKY, V. 2016 Instance normalization: The missing ingredient for fast stylization. arXiv:1607.08022

WANG, T.-C., LIU, M.-Y., ZHU, J.-Y., TAO, A., KAUTZ, J. & CATANZARO, B. 2018 High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8798–8807.

WANG, Z., BOVIK, A.C., SHEIKH, H.R. & SIMONCELLI, E.P. 2004 Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**, 600–612.

YOON, S., LEE, H.C. & PULLIAM, T.H. 2016 Computational analysis of multi-rotor flows. In *54th AIAA Aerospace Sciences Meeting*, 0812.

ZEILER, M.D., KRISHNAN, D., TAYLOR, G.W. & FERGUS, R. 2010 Deconvolutional networks. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2528–2535.

ZHANG, H., GOODFELLOW, I., METAXAS, D. & ODENA, A. 2019 Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pp. 7354–7363.

ZHANG, M. & PIGGOTT, M.D. 2020 Unsupervised learning of particle image velocimetry. In *International Conference on High Performance Computing*, pp. 102–115.

ZHANG, R., ISOLA, P. & EFROS, A.A. 2016 Colorful image colorization. In *European Conference on Computer Vision*, pp. 649–666.

ZHANG, R., ISOLA, P., EFROS, A.A., SHECHTMAN, E. & WANG, O. 2018 The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595.

ZHOU, B., KHOSLA, A., LAPEDRIZA, A., OLIVA, A. & TORRALBA, A. 2016 Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2921–2929.

https://doi.org/10.1017/jfm.2022.135 Published online by Cambridge University Press