



American Society of
Mechanical Engineers

ASME Accepted Manuscript Repository

Institutional Repository Cover Sheet

Cranfield Collection of E-Research - CERES

ASME Paper

Title: Surface line integral convolution-based vortex detection using computer vision

Authors: Hazem Ashor Amran Abolholl, Tom-Robin Teschner, Irene Moulitsas

ASME Journal

Title: Journal of Computing and Information Science in Engineering

Volume/Issue: Volume 23, Issue 5

Date of Publication (VOR* Online): 11 January 2023

ASME Digital Collection <https://asmedigitalcollection.asme.org/computingengineering/article/doi/10.1115/1.4056660/1156034/SURFACE-LINE-INTEGRAL-CONVOLUTION-BASED-VORTEX>

DOI: <https://doi.org/10.1115/1.4056660>

*VOR (version of record)

SURFACE LINE INTEGRAL CONVOLUTION-BASED VORTEX DETECTION USING COMPUTER VISION

Hazem Ashor Amran Abolholl¹, Tom-Robin Teschner^{1,*}, Irene Moultsas¹

¹Cranfield University, College Rd, Cranfield, Wharley End, Bedford MK43 0AL

ABSTRACT

Vortex cores in fluid mechanics are easy to visualise, yet difficult to detect numerically. Precise knowledge of these allows fluid dynamics researchers to study complex flow structures and allow for a better understanding of the turbulence transition process and the development and evolution of flow instabilities, to name but a few relevant areas. Various approaches such as the *Q*, delta, and swirling strength criterion have been proposed to visualise vortical flows and these approaches can be used to detect vortex core locations. Using these methods can result in spuriously detected vortex cores and which can be balanced by a cut-off filter, making these methods lack robustness. To overcome this shortcoming, we propose a new approach using convolutional neural networks to detect flow structures directly from streamline plots, using the line integral convolution method. We show that our computer vision-based approach is able to reduce the number of false positives and negatives while removing the need for a cut-off. We validate our approach using the Taylor-Green vortex problem to generate input images for our network. We show that with an increasing number of images used for training, we are able to monotonically reduce the number of false positives and negatives. We then apply our trained network to a different flow problem where vortices are still reliably detected. Thus, our study presents a robust approach that allows for reliable vortex detection which is applicable to a wide range of flow scenarios.

Keywords: computer-vision, vortex detection, line integral convolution, computational fluid dynamics, turbulent flows

2010 MSC : 68T45, 76D17, 76F65, 76M27

1. INTRODUCTION

Vortex core detection is a challenging subject in fluid mechanics research and how to detect these reliably remains an open question to date. Knowing the exact location of vortex cores is useful during automated mesh refinement in computational fluid dynamics simulations but provides equally valuable insights, for example, to engineers trying to optimise aerodynamic devices to channel turbulent vortices around objects to reduce drag. Identifying single vortices may be done using simple velocity gradient-

based approaches, however, complex engineering applications see the merging and mixing of vortices where these simplistic approaches may fail. Knowing the precise location of vortices has far-reaching practical benefits, yet our numerical tools are still not able to predict these vortices reliably and they produce spurious detected vortex cores, negating their usefulness.

Vortices are usually identified based on the velocity-gradient tensor [1] and there is a range of different criteria available, among which the *Q* criterion [2], Lambda 2 criterion [3], the delta criterion [4], and the swirling strength criterion [5] are the most used ones and referred to here as local detection methods. While these criteria are cheap to compute, they lack robustness as they easily detect spurious vortices known as false positives (no vortex exists but is identified) and false negatives (a vortex exists but is not identified). To limit the number of false positives and negatives, the user can prescribe a threshold below which vortices are filtered. This threshold value is, however, flow-dependent and therefore requires re-calibration for each flow problem.

In order to overcome this problem, we present a vortex detection method based on Computer vision. Here, we focus our attention on finding bounding boxes that enclose vortex regions rather than exact vortex core locations. While this may seem limited at first, we may not need to know the exact vortex cores in order to perform local mesh refinement (which refines a volume and not a single line) or track a vortex by itself (or rather its bounding box). Rather, we focus our attention here on providing a more reliable, more accurate, and more robust way of predicting vortex regions which can then be used in engineering design processes with higher confidence. Our approach can be applied to any 3D data set of vortical flows; as long as 2D planes through a 3D flow volume can be constructed through which vortices are passing through and as long as these vortices can be visualised through surface streamline plots as discussed in Section 3.3, our proposed framework will be able to identify these vortices. The advantage here is that vortices do not have to be aligned with the generated 2D planes and can pass through these at an angle, providing a more generalised approach to detect vortices.

The recently introduced You Only Look Once (YOLO) [6] computer vision approach differentiates itself from similar feature detection methods by applying a single neural network to the whole image instead of making predictions on many regions within an

*Corresponding author: tom.teschner@cranfield.ac.uk

Documentation for asmeconf.cls: Version 1.30, January 13, 2023.

image. Moreover, it has a good balance between accuracy and speed [7, 8]. Therefore, it has been applied broadly to various applications such as the detection of humans [9], gesture recognition [7], and detection of objects through aerial observations [10]. The algorithm has been continuously improved [11] and we use here version 3 of the algorithm.

The remaining structure of this article is as follows: Section 2 reviews the literature on vortex core detection and computer vision. Section 3 introduces the methodology and tools used in the present study. Section 4 provides an overview of the numerical test case that is solved using computational fluid dynamics and used to produce input for our computer vision framework. Section 5 analyses how our computer vision approach compares against local vortex detection methods, and we provide a summary of the main points in Section 6.

2. LITERATURE REVIEW

The progress in transistor technology has established computational fluid dynamics as a tool to investigate turbulence flows down to the smallest scales of motion [12]. As was alluded to in the introduction, vortices are encountered in a range of engineering and scientific problems where they play an important role, such as aircraft navigation during nose-up or roll of a jet [13], medical [14, 15], as well as industrial applications [16]. Various definitions for a vortex can be found in the literature [17]. Lugt [18] states that “a vortex is the rotating motion of a multitude of material particles around a common centre”. This definition is difficult if not impossible, to implement into an algorithm [1]. Robinson [19] provided another definition and says that “a vortex exists when instantaneous streamlines mapped onto a plane normal to the vortex core exhibit a roughly circular or spiral pattern when viewed from a reference frame moving with the centre of the vortex”. This definition is self-referential, as we would need prior knowledge of the location, the orientation, and the motion of the vortex core in order to identify it.

Notwithstanding the lack of a formal definition, using available methods to identify vortices has been extensively investigated in recent years. There are two common types to detect vortices, based on local and global detection algorithms [20–22]. Local methods [2–4] are based on the decomposition of the local velocity gradient tensor. They provide rapid results and are easy to implement. However, these methods need to carefully select suitable parameters to get robust results, which increases the time to identify vortex cores. Global vortex identification methods generally utilize streamlines to identify vortex regions. The winding angle [20], instantaneous vorticity deviation [21], and elliptic objective Eulerian coherent structures method [22] are examples of global methods. In terms of robustness, global methods use neighbouring cells to identify vortical flows and thus tend to be more accurate than local methods, which comes at an expense of increased computational time for global methods [23]. Furthermore, global methods require user input which makes them unsuitable for automatic vortex core detection tasks. Therefore, both local and global methods have both drawbacks and cannot obtain fully robust and reliable detection results.

The recent advances made in the field of machine learning have seen a multitude of applications for physics and simulation-based

problems. Li et al. [24] proposed to use a super-resolution generative adversarial network (SRGAN) to reconstruct fluid phase fractions in turbulent, multiphase flows. Ayli et al. [25] used machine learning for active flow control applied to the flow around a circular cylinder. Warey et al. [26] used conditional invertible neural networks (cINNs) to optimise shapes. They applied it to 2D airfoil generation as a classical optimisation problem. Furthermore, Nabian and Meidani [27] proposed a more generic framework for general physical systems that utilises machine learning to aid in engineering design and analysis tasks. We follow a similar approach, by adopting a machine learning approach to identify vortices in simulation data. In order to address the existing problems within traditional vortex identification methods, recently Computer Vision was utilized to detect vortical structures in the flow field to obtain robust and reliable results. Many of the techniques developed in Computer Vision can be used to extract, track, and identify features [28, 29]. Visualisation aims to understand, analyse and identify physical phenomena or mathematical models. Therefore, the criteria for tracking features in a scientific area are different than for most computer vision applications. In a scientific simulation, evolving features may split, merge, or disappear [30]. Computer vision technology has been tested using deep neural networks to solve problems of reconstruction of velocity and pressure fields of a turbulent flow with high resolution and to identify the characteristic flow features [31]. In the past decade, deep neural network (DNN)-based machine learning has shown great success in natural language processing and computer vision [32–35] and has recently been shown to be able to identify vortical flows. Lguensat et al. [36] utilized a DNN to identify ocean eddies from sea surface height maps. They considered the entire visualization images of the flow field as inputs to detect the ocean eddies. Franz et al. [37] used a convolutional neural network (CNN) to detect and track ocean eddies and provided the labelled training data by local vortex identification that need suitable parameters to get robust results. Similarly, Ströfer et al. [38] developed a CNN for identifying the horseshoe vortices in a 3D flow over a wing-body junction. Deng et al. [39] presented a vortex identification method based on a CNN. They used the global instantaneous vorticity deviation method of Haller et al. [21] and a local method to obtain the labels of each point in the flow field for training. The local vortex detection methods are used to identify vortices and train the network which can be automated in this way. However, local methods are prone to provide spurious vortices and thus the network may be trained based on false positives and negatives. Expanding on the work of Deng et al., we propose an alternative way to use CNN to track and identify vortical structures. We replace local detection methods for labelling by manually tagging vortices on streamline plots. While this requires a substantial manual upfront cost, this needs to be done only once and with sufficient vortices labelled for different flow situations, a trained network can be applied for various flow scenarios. By adopting this approach, we are able to eliminate false positives and negatives and thus increase the overall robustness and accuracy of CNNs for vortex identification tasks. This approach is detailed in the next section.

3. METHODOLOGY

In the following section, we introduce and provide a brief description of the underlying methods we make use of in this study. We introduce the Navier-Stokes equations in the form it is solved for our turbulent flow, provide a description of local vortex detection algorithms as well as propose our computer vision-based approach to detect vortices based on the line integral convolution method.

3.1 Governing equations for incompressible computational fluid dynamics

The motion of fluids is entirely described by the Navier-Stokes equations. It is a set of conservation laws, namely conservation of mass, momentum, and energy and provides in total a set of five equations and six unknown quantities (density, pressure, energy (temperature), and three velocity components in the x-, y- and z-direction). For a compressible fluid (that is, with a variable density), we can close the Navier-Stokes equations through a suitable state equation such as the ideal gas law. For an incompressible fluid, such as we are considering in this study, the density is constant, and no such relation exists. In this case, various pressure-velocity coupling schemes have been proposed that work in a predictor-corrector fashion and iteratively find the solution to a given problem, of which the SIMPLE scheme of Patankar [40] is one of the most widely used one. The conservation of momentum is given by

$$\rho \frac{\partial U}{\partial t} + \rho \nabla \cdot (UU) = -\nabla p + 2\mu \nabla^2 S - \nabla \tau, \quad (1)$$

where U is the velocity vector (u_x, u_y, u_z), ρ the density, t the time, p is pressure, μ is dynamic viscosity and S and τ being the rate-of-strain and stress tensor, respectively. The rate-of-strain tensor can be defined in tensor notation as

$$S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right), \quad (2)$$

and the stress tensor as

$$\tau_{ij} - \frac{1}{3} \tau_{kk} \delta_{ij} = -2v_{SGS} S, \quad (3)$$

The conservation of mass, in turn, is given as

$$\nabla \cdot U = 0. \quad (4)$$

In the equations above, we have spatially filtered the Navier-Stokes equations where small-scale turbulence features that are smaller than a filter width Δ have to be modelled through a subgrid-scale model. The filtering operation produces a stress tensor τ that is commonly modelled with the relation provided in Eq.(3), where we can ignore the contribution of τ_{kk} . We replace the turbulent stresses through the strain-of-rate tensor S which is scaled by the turbulent subgrid-scale viscosity and which in turn accounts for the filtered smallest turbulent eddies. As we are now only resolving the largest turbulent flow structure and modelling the smallest scale, this approach is referred to as large-eddy simulations or LES in short. Using LES to resolve turbulence and thus vortical flows, we have to introduce a subgrid-scale model [41]

to calculate the turbulent viscosity to close the set of momentum equations, i.e. Eq.(1). In this study, we make use of the Smagorinsky-Lilly [42] model which is given by

$$v_{SGS} = C_S \Delta^2 \sqrt{2SS}, \quad (5)$$

where C_S is an empirical constant and tuned for a range of canonical turbulent flows and Δ the filter width below which turbulent eddies are filtered (typical related to the local mesh size). The advantage of LES is that we can generate turbulent vortical structures quickly while only removing the smallest length scales which can be easily modelled. The alternative is to model the full turbulent flow field which still requires significant computational resources. In this case, we do not filter the Navier-Stokes equations but instead ensure that our computational grid can capture all turbulent length scales (in which case the stress tensor τ does not need to be modelled and can be computed directly from the resolved velocity field). This approach is referred to as direct numerical simulations or DNS in short. Since we are only interested in detecting macroscopic vortical structures, LES provides excellent accuracy and sufficient information about turbulent flows, which is suitable to capture vortices for training our CNN. Therefore, LES has been used in this study to generate data quickly.

3.2 Local vortex identification methods

The detection methods rely on the velocity gradient tensor, ∇U , that can be written as

$$D_{ij} = \frac{\partial u_i}{\partial x_j}, \quad (6)$$

where $i = 1, 2, 3$ for x_i and u_i correspond to x, y, z and u, v, w, respectively. The velocity gradient tensor D_{ij} can be decomposed into a symmetric and anti-symmetric tensor as $D_{ij} = S_{ij} + \Omega_{ij}$ with

$$S_{ij} = \frac{1}{2} (D_{ij} + D_{ji}), \quad (7)$$

$$\Omega_{ij} = \frac{1}{2} (D_{ij} - D_{ji}), \quad (8)$$

and where S_{ij} is the symmetric part of the velocity gradient tensor which is commonly known as the rate-of-strain tensor and Ω is the anti-symmetric component representing the rate of rotation tensor. The characteristic equation for D is given as [43]

$$\sigma^3 + P\sigma^2 + Q\sigma + R = 0, \quad (9)$$

where P , Q , and R are the first, second, and third invariants of the velocity gradient tensor. Using the decomposition into symmetric and anti-symmetric tensors, these invariants can be expressed as

$$P = -\text{tr}(\nabla U) = 0, \quad (10)$$

$$Q = \frac{1}{2} \left(\text{tr}(\nabla U)^2 - \text{tr}(\nabla U^2) \right) = \frac{1}{2} \left(\|\Omega\|^2 - \|S\|^2 \right), \quad (11)$$

$$R = -\det(\nabla U). \quad (12)$$

Here we have $\|S\| = [\text{tr}(SS^t)]^{1/2}$, and $\|\Omega\| = [\text{tr}(\Omega\Omega^t)]^{1/2}$. The Q , delta, and swirling strength criteria are the most common local

detection methods based on the velocity gradient tensor D_{ij} . The Q criterion (Hunt et al. [2]) identifies vortices as flow regions with the positive second invariant of ∇U , i.e. $Q > 0$. The delta criterion is using the critical point theory of Chong et al. [4] and defines a vortex core for regions where D_{ij} has complex eigenvalues. These are identified using the discriminant of the characteristic equation, which is given by

$$\Delta = \left(\frac{1}{2}R\right)^2 + \left(\frac{1}{3}Q\right)^3 > \epsilon, \quad (13)$$

In a strict mathematical sense, ϵ is set to zero which can result in falsely detected vortices. To reduce these false positives, we can prescribe a value of $\epsilon > 0$ which increases the accuracy of the model but reduces the robustness of this method, as ϵ is a parameter specific to each vortex. Setting ϵ too low results in vortices that are detected in areas where no vortices exist (false positive). A value of ϵ too high results in vortices that are not detected even though they exist in physical space (false negative). The Swirling strength criterion of Zhou et al. [5] is similarly defined to the delta criterion as the region where D_{ij} has complex eigenvalues. Unlike the delta criterion, however, Zhou et al. [5] used the velocity gradient decomposition as follows

$$D_{ij} = [v_r \ v_{cr} \ v_{ci}]^T \begin{bmatrix} \lambda_r & 0 & 0 \\ 0 & \lambda_{cr} & \lambda_{ci} \\ 0 & -\lambda_{ci} & \lambda_{cr} \end{bmatrix} [v_r \ v_{cr} \ v_{ci}]^T. \quad (14)$$

Here, λ_r is defined as the real eigenvalue which is related to the real eigenvector v_r and we have the conjugate pair of complex eigenvalues $\lambda_{cr} \pm i\lambda_{ci}$ which is related to the corresponding eigenvectors $v_{cr} \pm iv_{ci}$. The local swirling strength of a vortex is defined through λ_{ci} and for values greater than zero, a vortex is identified. This definition, however, produces similar falsely detected vortices, and thus a more conservative approach would be to define $\lambda_{ci} > \epsilon$. Using this approach, we need to fine-tune the threshold parameter ϵ which is again a vortex specific parameter. Due to the singular nature of the threshold parameter, we are not able to avoid the detection of false positives and negatives, as vortices may be cut-off below the threshold or spurious vortices are allowed to be detected above the threshold.

The above described vortex detection methods were implemented by Lindner et al. [43] into their Vortex Fitting toolbox. We use this toolbox to detect vortices by fine-tuning the threshold parameter ϵ for each case individually. As we will show, the results are highly dependent on ϵ and despite searching for the optimum threshold value, false positives and negatives still exist. In order to overcome this shortcoming, we introduce a computer vision approach based on streamline-based vortex detection to reduce false positives and negatives.

3.3 Streamline detection using the line integral convolution algorithm

As the input to our computer vision framework, we generate streamline plots on predefined planes within our computational domain where we extract the velocity vector and compute from it the local streamlines for each pixel using the line integral convolution (LIC) algorithm, as proposed by Cabral and Leedom [44].

This technique starts by covering a surface with a random gray-scale value at the resolution at which streamline plots are to be generated. It then computes the following integral

$$C(r) = \int_{-L/2}^{L/2} k(s)N(\sigma(s))ds, \quad (15)$$

where $C(r)$ represents the colour value at location r , s being the arc length along the streamline, $k(s)$ is an appropriately chosen kernel function to limit colouring to local space and $N(\sigma(s))$ containing the gray-scale image. We integrate along the streamline ds for a distance L . We apply this for each pixel in the domain which produces LIC plots as shown in Figure 6. Compared to traditional streamline plots, LIC streamline plots provide the advantage of covering the entire image whereas traditional streamline plots require a starting location through which streamlines have to pass, which does not guarantee full coverage of the domain. Furthermore, in 3D, streamlines may leave a 2D plane on which visualisation is sought, making it a difficult task to generate surface streamline plots and extract vortex cores automatically. We generate these LIC streamline plots using ParaView and these are then further pre-processed using the Torchvision module — which is part of the PyTorch Python package — by applying rotation, cropping, and horizontal or vertical flipping to the images to make the training more robust. To construct the training and validation data set, the planes were randomly chosen at $-\pi/2$, 0 , or $\pi/2$ in the x , y , or z direction. For our supervised learning approach, we manually label vortex regions for 100 images where we identified a total of 1646 vortices. While this step represents a time-consuming step, it has to be done only once. We later show that this approach can then be transferred to different fluid problems for which no training has taken place and vortices can still be detected reliably.

3.4 Computer vision-based vortex detection with YOLO

Computer vision is a sub-field of artificial intelligence that is able to extract meaningful information from digital images and videos. In our case, we use it to extract vortices from streamline plots. In this work, we used YOLO version 3 [11] for enhancing the vortex core detection accuracy and reducing false positives and false negatives. YOLO passes the entire image into a Neural Network. This Neural Network divides the image into cells and provides probabilities for every cell. Then YOLO predicts the number of Bounding Boxes within the image and chooses the best ones based on the probabilities. The backbone network of YOLO v3 is called Darknet-53, which has 53 CNNs layers, to extract feature images from the input. The basic components of YOLO's architecture are shown in Figure 1 and consist of residual blocks, skip connections and up-sampling layers, where each convolutional layer is followed by a batch normalisation layer and Leaky ReLU activation function. YOLO uses three different resolutions, which are responsible for detecting objects of different scales, and at three separate places in the Network. Therefore, the original architecture of YOLO adds 53 additional layers and yields 106 layers for detection tasks. These separate places for detection are layers 82, 94, and 106. The size of the image input is divided by the network into 32, 16, and 8 [45] and referred to as the network's stride. Therefore, the input image to the Network

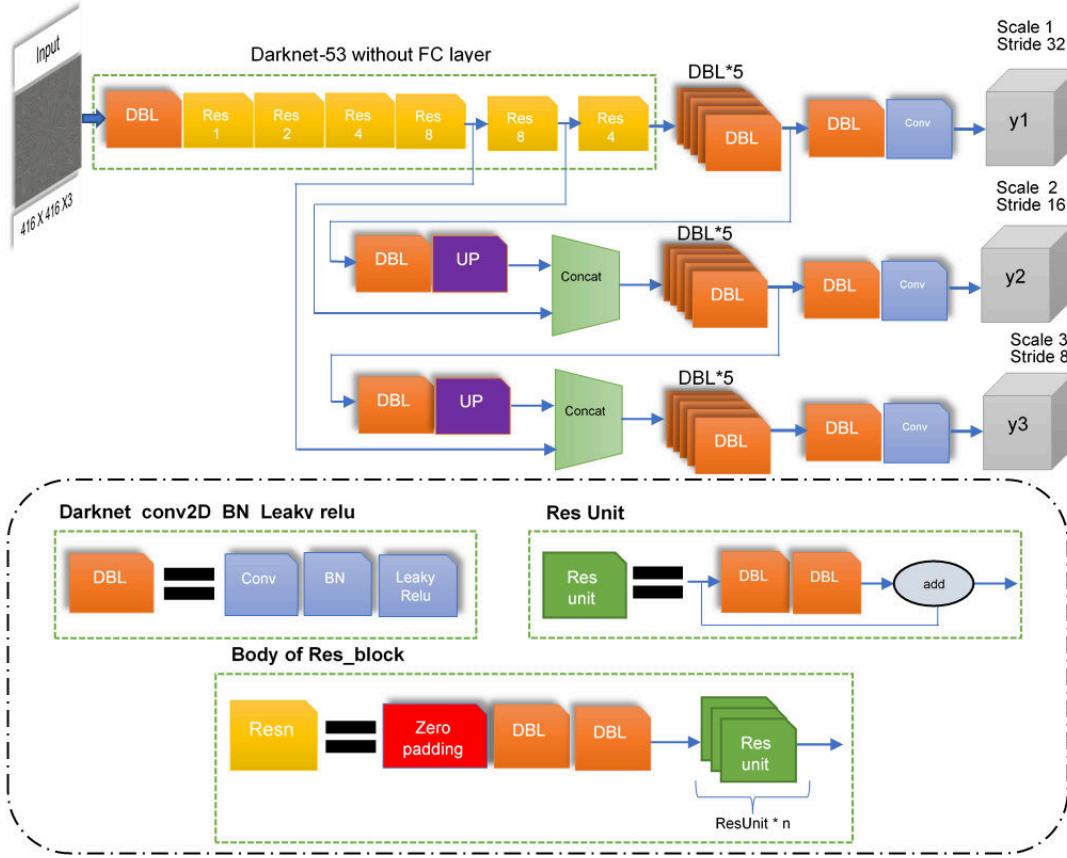


FIGURE 1: YOLOV3 ARCHITECTURE WITH INPUT IMAGE SIZE 416×416 AND 3 TYPES OF SCALES (13×13, 26×26, AND 52×52) AS OUTPUT; DARKNET-CONV2D-BN-LEAKY ("DBL" FOR SHORT) IS THE MAIN COMPONENT OF YOLOV3 WHICH CONTAINS ONE CONVOLUTION LAYER, ONE BATCH NORMALIZATION LAYER, AND ONE LEAKY RELU LAYER.; RESIDUAL-LIKE UNIT ("RESUNIT" FOR SHORT) IS TWO "DBL" STRUCTURES FOLLOWING ONE "ADD" LAYER; SEVERAL "RESUNIT" WITH ONE ZERO-PADDING LAYER AND "DBL" STRUCTURE FORWARD GENERATES A RESIDUAL-LIKE BLOCK, "RESBLOCK" FOR SHORT, WHICH IS THE MODULE ELEMENT OF DARKNET-53. FIGURE ADAPTED FROM WANG ET AL. [45]

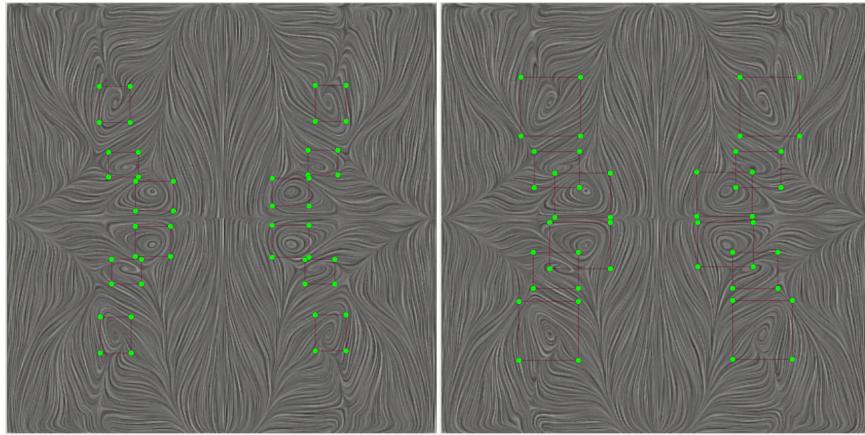


FIGURE 2: DIFFERENT LABELLING STRATEGIES: ON THE LEFT-HAND SIDE, WE USE NON-OVERLAPPING BOUNDING BOXES AND ON THE RIGHT-HAND SIDE WE ALLOW LABELLED BOUNDING BOXES TO OVERLAP, CAPTURING MORE OF THE VORTICES' STRUCTURES.

must be divisible by 32 without leaving a remainder, in our case the size of the input image is 416 pixels. Stride 32 will produce an output size of 13 by 13. Similarly, stride 16 will produce an output of size 26 by 26 and stride 8 of 52 by 52. Therefore, strides 8, 16, and 32 are responsible for detecting small, medium, and

large features in our images, respectively. In contrast, the previous versions (YOLOv1, YOLOv2) had difficulty detecting small objects since the images were divided into equal-sized grid cells with each grid cell detecting object if the central point of the object fell into that cell. YOLOv4 and subsequent versions generate

the same three different outputs of feature maps as YOLOv3 to achieve multi-scale prediction. In terms of accuracy, YOLOv4 and above may achieve slightly better accuracy than YOLOv3, especially in the MS COCO dataset, however, YOLOv3's detection speed is faster than that of YOLOv4 and versions above [46]. For that reason, we have chosen to use YOLOv3 in this study to detect vortex regions with good speed and high accuracy which is suitable to detect small areas of vortices.

3.5 Vortex regions labelling

Labelling of vortices poses some difficulties and needs to be considered carefully in order to avoid false positives and negatives. As vortices are not well-defined structures (where does a vortex start and where does it end?), we have implemented two different labelling strategies and compared their effect on the results. Both strategies are schematically shown in Figure 2. On the left-hand side, we show an approach in which the bounding boxes we use for labelling do not overlap. This ensures that each bounding box only considers one vortex and the influence from a close neighbouring vortex does not bleed into this bounding box. On the right-hand side, we do allow the labelled bounding boxes to overlap and concentrate here on capturing as much of the vortex as possible. This allows for neighbouring vortices to affect vortex detection. The idea here is that neighbouring vortices will have an effect on the shape and structure of the vortex itself and using this approach may help in predicting vortices of non-perfectly spherical shape better.

4. TEST CASE

We use the Taylor Green Vortex problem as a test case. The Taylor-Green vortex flow is simulated using LES at a Reynolds number of $\text{Re}=1600$ [47]. The large-scale turbulence is solved directly using the Navier-Stokes equations, and the small-scale turbulence is modelled using the Smagorinsky-Lilly subgrid-scale model. The geometry has been discretized by a uniform mesh of resolution 64^3 with a time-step of $\Delta t^* \approx 0.06$, which corresponds to a CFL value of 0.6. Here, we use the non-dimensional time t^* which is expressed as $t^* = tU_0/L$. All six sides of the cube of length $2\pi L$ (with $L = 1$) are set to periodic boundary conditions. The initial conditions are set in such a way that an initial distribution of large, energy-containing vortices exists which then decays through vortex stretching in three-dimensional space. The initial conditions are given by

$$u = U_0 \sin\left(\frac{x}{L}\right) \cos\left(\frac{y}{L}\right) \cos\left(\frac{z}{L}\right), \quad (16)$$

$$v = -U_0 \cos\left(\frac{x}{L}\right) \sin\left(\frac{y}{L}\right) \cos\left(\frac{Z}{L}\right), \quad (17)$$

$$w = 0, \quad (18)$$

$$p = p_0 + \frac{\rho_0 U_0^2}{10} \left(\cos\left(\frac{2x}{L}\right) + \cos\left(\frac{2y}{L}\right) \right) \left(\cos\left(\frac{2z}{L}\right) + 2 \right). \quad (19)$$

Here, we set $U_0 = 1 \text{ m s}^{-1}$, $p_0 = 0 \text{ Pa}$ and $\rho_0 = 1 \text{ kg m}^{-3}$. Setting the kinematic viscosity to $\nu = 1/1600 \text{ m}^2 \text{s}^{-1}$ then results in the desired Reynolds number of $\text{Re}=1600$. The left-hand side of Figure 3 shows the vortex distribution at $t^* = 0$

using the magnitude of the velocity vector. The right-hand side shows the state of the velocity vector at $t^* = 20$, after vortex stretching has redistributed the containing energy in space, with the energy contained at the smallest turbulent scales being dissipated into heat. From Figure 3, we can see that a range of vortices will develop from the initial conditions and this will provide us with a reliable mechanism to generate vortical flow patterns from which we can extract streamlines on the symmetry planes of the x-, y- and z-direction which are then used training data for our computer vision-based framework.

To validate our simulation setup, we run the Taylor-Green vortex problem until $t^* = 20$ and compare the computed overall kinetic energy with that of Beck and Gassner [48] and DeBonis [49]. The temporal evolution of kinetic energy, based on the instantaneous velocity vector, is computed as

$$E_k = \frac{1}{\rho_0 V} \int_V \rho \frac{U \cdot U}{2} dV. \quad (20)$$

The temporal evolution of the kinetic energy dissipation rate is computed as

$$\varepsilon = -\frac{dE_k}{dt}, \quad (21)$$

These two quantities are plotted and compared against reference data in Figure 4 and Figure 5, respectively. We compare our data here against direct numerical simulation (DNS) results, as well as LES results done on the same mesh size as our simulation. From Figure 4, we can see that our results prematurely capture an increase in dissipation rate compared to the DNS results. However, this trend is explained by the coarse resolution of the 64^3 grid cells in our simulation. Comparing our results against LES data obtained on a similar grid shows that we match that data much better. Looking at Figure 5, we capture the kinetic energy temporal evolution well and there are only minute differences between the LES and DNS reference data. It should be stressed here that our primary concern is to generate vortical structures quickly while capturing the overall physical process of turbulence and its decay. Our validation shows that this is the case here and that we can proceed with these simulations to generate our streamline plots for the vortex core detection task.

5. RESULTS AND DISCUSSIONS

A total of 335 time steps are required to advance the solution from $t^* = 0$ to $t^* = 20$ at $CFL = 0.6$. We store the velocity vector on the symmetry planes in the x-, y- and z-direction at each time-step. In the current investigation, we extract a total of 100 images from the symmetry plane norm to the x, y, and z-direction as training dataset and label vortices on these 100 images based which are visualised through the line integral convolution method. We then select two additional images from the symmetry plane, which are not part of the training data set, and use that for testing the accuracy of our vortex detection framework based on computer vision. The two images are shown in Figure 6, where test image 1, on the left-hand side, contains 16 vortices, and test image 2, on the right-hand side, contains 24 vortices. We observe that the test images contain mostly symmetrical vortex structures along the main axes.

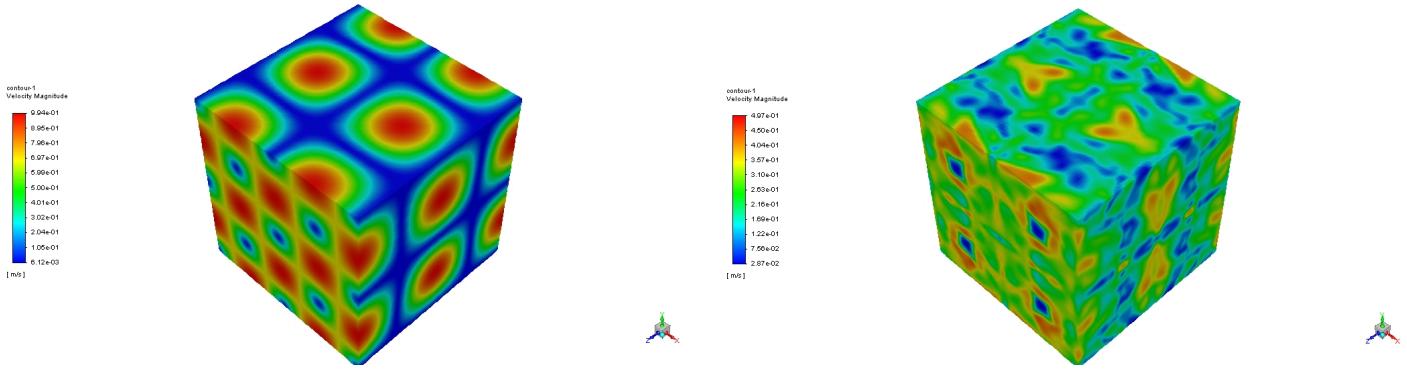


FIGURE 3: CONTOUR VELOCITY PLOT. THE LEFT-HAND SIDE AT THE BEGINNING INITIAL STATE AND THE RIGHT-HAND SIDE AT THE END TIME STEP.

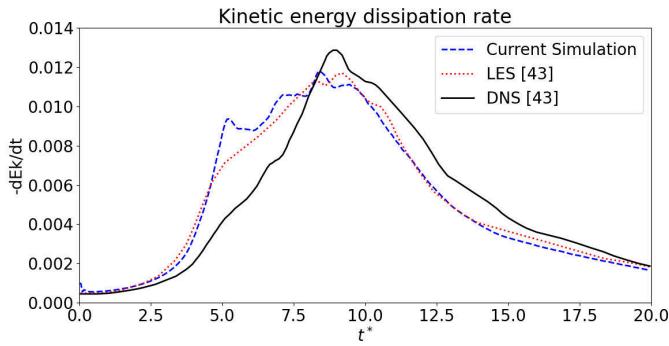


FIGURE 4: KINETIC ENERGY DISSIPATION RATE. THE REFERENCE LARGE EDDY SIMULATION (LES) AND DIRECT NUMERICAL SIMULATION (DNS) WAS TAKEN FROM [48], THE CURRENT SIMULATION RESULTS SHOWN IN THE DASHED BLUE LINE WERE OBTAINED USING LES ON A 64^3 GRID AT A REYNOLDS NUMBER OF RE=1600.

First, we will use these two test images to detect vortices using the Vortex fitting toolbox of Lindner et al. [43] and show results for that in the next section. Then, we apply the same test images to our trained computer vision-based framework and show how the number of images used for training influences the accuracy.

5.1 Local vortex detection methods

Figure 7 and Figure 8 show the results of the two test images after detecting vortices using the Vortex fitting toolbox [43]. We use the Q, delta, and swirling strength criterion here and vary the numerical discretisation schemes used to approximate the derivatives within the velocity gradient tensor. Here, we use second-order, fourth-order, and a least square gradient reconstruction approach. Furthermore, we fine-tune the threshold value to match the number of expected vortices for both cases, here 16 for test image 1 (Figure 7) and 24 vortices for test image 2 (Figure 8). This is only possible for the delta and swirling strength criterion, whereas the Q criterion does not allow for a threshold to be fine-tuned.

Here, the yellow dots in the images indicate clockwise rotat-

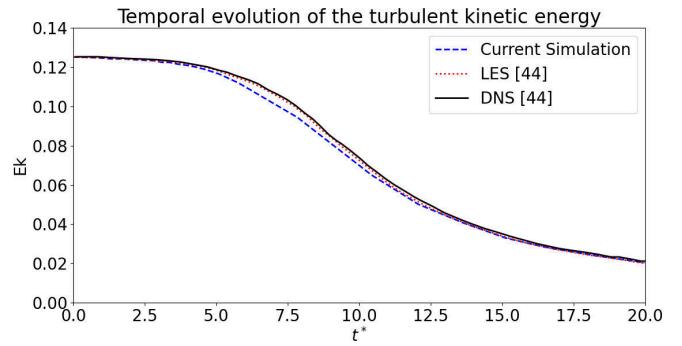
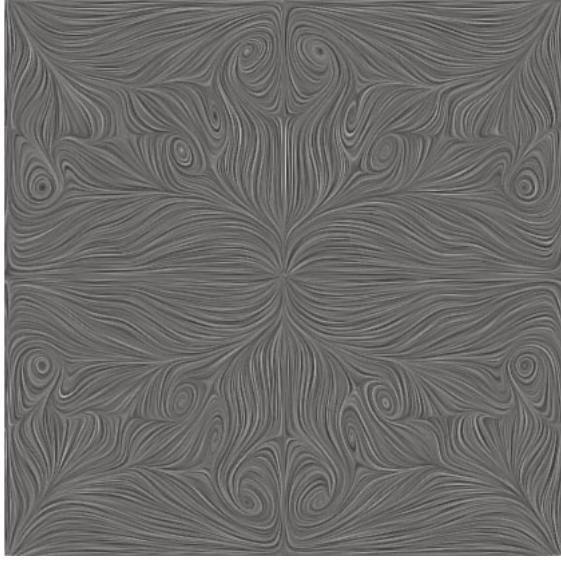


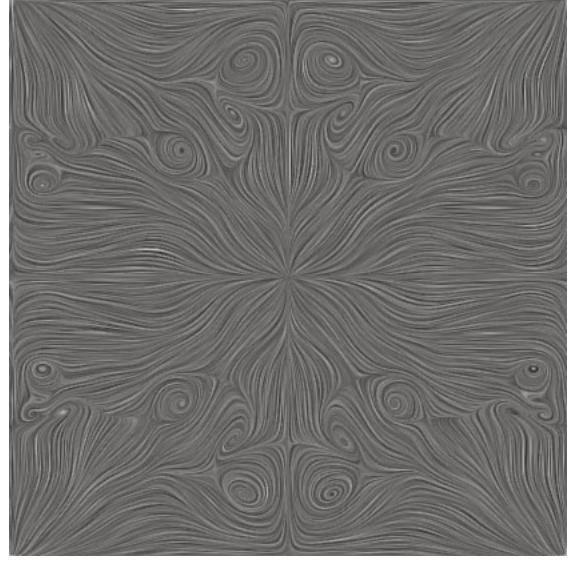
FIGURE 5: TEMPORAL EVOLUTION OF THE TURBULENT KINETIC ENERGY. THE REFERENCE LARGE EDDY SIMULATION (LES) AND DIRECT NUMERICAL SIMULATION (DNS) WAS TAKEN FROM [49], THE CURRENT SIMULATION RESULTS SHOWN IN THE DASHED BLUE LINE WERE OBTAINED USING LES ON A 64^3 GRID AT A REYNOLDS NUMBER OF RE=1600.

ing vortices, the green dots indicate counter-clockwise rotating vortices and the red dots show the actual location of vortex cores, which were manually extracted from Figure 6.

Both Figures show a similar behaviour in terms of performance for the three different detection algorithms. First of all, there is little coherence between the different methods and, to some extent, there is also a lack of coherence between the different gradient reconstruction schemes for the same method. What is interesting to note is that despite providing an almost symmetrical flow field to the vortex detection algorithms, this symmetry is not reflected in the predicted vortices. Some detection algorithms tend to a more symmetrical vortex core prediction but overall a non-symmetrical prediction pattern is established. This shows the sensitivity of the different methods to the threshold parameter ϵ (for the delta and swirling strength criterion) as fine-tuning this may remove some of the mirrored vortices due to a slight difference in vortex strength. Furthermore, we can observe that all these detection methods provide, at best, an approximation of actual vortex core regions. It can be noted from a qualitative point of view, that the delta criterion seems to predict regions of high vortical flow correctly, despite not being able to predict the



(a) *Test image 1 with 16 vortices.*



(b) *Test image 2 with 24 vortices.*

FIGURE 6: TEST IMAGES SHOWING VORTICAL STRUCTURES OBTAINED WITH THE LINE INTEGRAL CONVOLUTION-BASED STREAMLINE ALGORITHM.

TABLE 1: NUMBER OF PREDICTED VORTICES FOR THE Q, DELTA, AND SWIRLING STRENGTH CRITERION USING A SECOND-ORDER GRADIENT RECONSTRUCTION SCHEME, AS WELL AS THEIR FALSE POSITIVES AND NEGATIVES AND THE THRESHOLD VALUE REQUIRED TO MATCH THE NUMBER OF PREDICTED VORTICES TO THE NUMBER OF EXPECTED VORTICES (EV)

Test image	Second-order												
	Threshold value			Detected vortices			False positives			False negatives			
	Q	delta	swirl	Q	delta	swirl	Q	delta	swirl	Q	delta	swirl	
1	n/a	1.2e-09	0.9	12	16	16	4	8	4	8	7	4	16
2	n/a	4.3e-08	1.37	19	24	24	4	3	3	9	4	4	24

TABLE 2: NUMBER OF PREDICTED VORTICES FOR THE Q, DELTA, AND SWIRLING STRENGTH CRITERION USING A FOURTH-ORDER GRADIENT RECONSTRUCTION SCHEME, AS WELL AS THEIR FALSE POSITIVES AND NEGATIVES AND THE THRESHOLD VALUE REQUIRED TO MATCH THE NUMBER OF PREDICTED VORTICES TO THE NUMBER OF EXPECTED VORTICES (EV)

Test image	Fourth-order												
	Threshold value			Detected vortices			False positives			False negatives			
	Q	delta	swirl	Q	delta	swirl	Q	delta	swirl	Q	delta	swirl	
1	n/a	2.0e-10	0.0	11	16	15	2	2	3	7	2	4	16
2	n/a	2.4e-08	0.0	8	24	12	0	1	1	16	1	13	24

TABLE 3: NUMBER OF PREDICTED VORTICES FOR THE Q, DELTA, AND SWIRLING STRENGTH CRITERION USING A LEAST-SQUARE GRADIENT RECONSTRUCTION SCHEME, AS WELL AS THEIR FALSE POSITIVES AND NEGATIVES AND THE THRESHOLD VALUE ϵ REQUIRED TO MATCH THE NUMBER OF PREDICTED VORTICES TO THE NUMBER OF EXPECTED VORTICES (EV)

Test image	Least square												
	Threshold value			Detected vortices			False positives			False negatives			
	Q	delta	swirl	Q	delta	swirl	Q	delta	swirl	Q	delta	swirl	
1	n/a	1.4e-08	3.2	15	16	16	3	3	6	4	1	4	16
2	n/a	1.9e-07	0.0	6	24	9	0	4	0	18	2	15	24

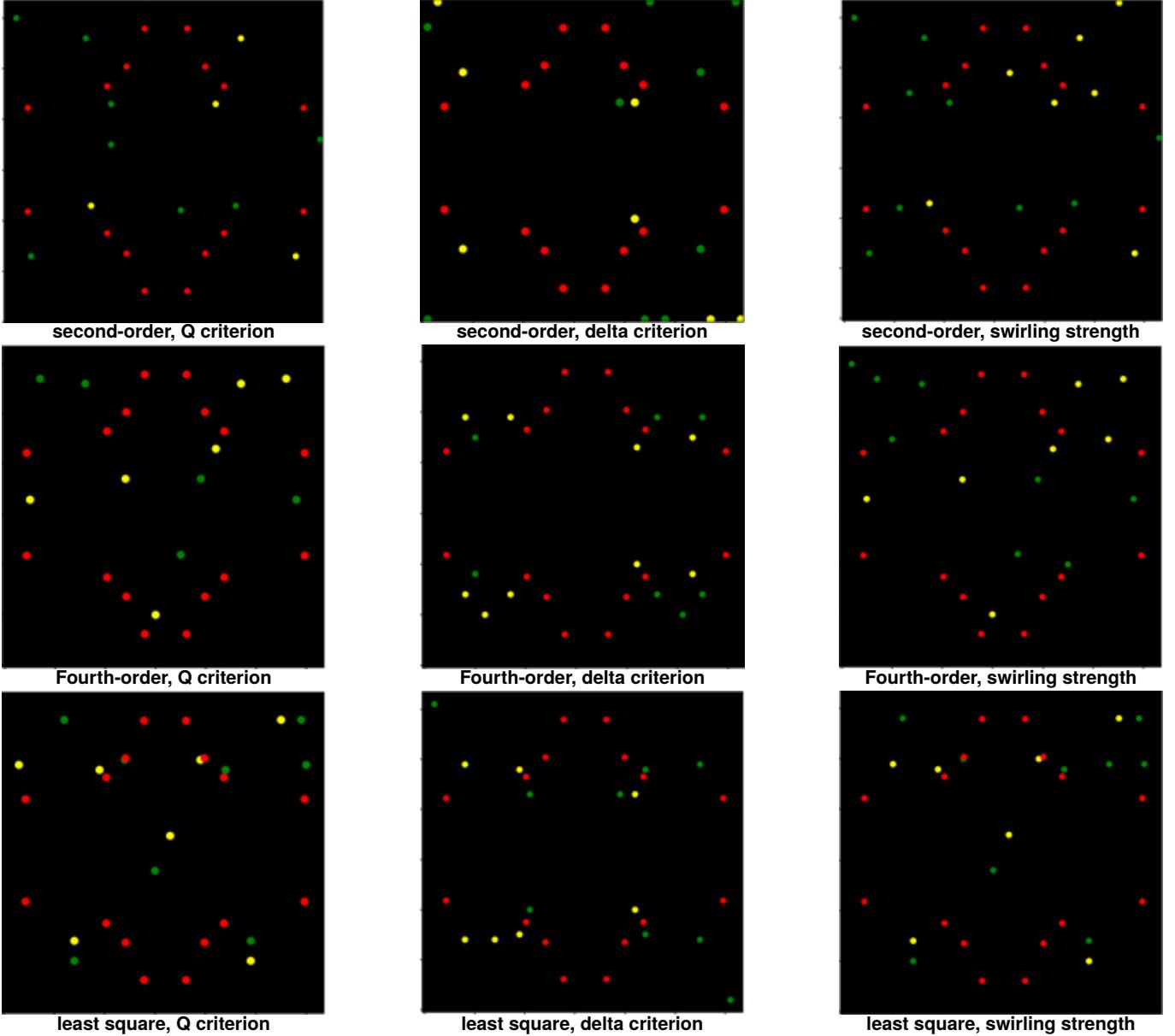


FIGURE 7: DETECTED VORTICES (SHOWN ON YELLOW AND GREEN) USING THE Q, DELTA, AND SWIRLING STRENGTH CRITERION WITH SECOND-ORDER, FOURTH-ORDER, AND LEAST SQUARE GRADIENT RECONSTRUCTION SCHEMES FOR THE FIRST TEST IMAGE (16 VORTICES). THE ACTUAL VORTEX CORE REGIONS ARE SHOWN BY THE RED CIRCLES.

exact location of vortex cores. This is especially noticeable for the second test image. Both the Q and swirling strength criterion are picking up on the rotational flow at the centre; we may have vortical flow here but within the plane we are observing the flow, the centre part of the streamlines cannot be classified as a vortex. Overall, we can say that we get a good initial prediction of areas where vortices are located but tracking down their precise location proves difficult. Without manually detecting where vortices are located, it is also difficult to judge if the reported results are physical or contain false positives and negatives.

Turning to Table 1–3, these summarise how the different methods compare in terms of predicting false positives and negatives for the three different gradient reconstruction schemes. Here, we count vortices as being correctly predicted if they are in the

vicinity of an actual vortex core. These three tables provide in numbers what has been discussed above but further show that a large number of false positives and negatives exist for all detection algorithms and reconstruction schemes and, apart from the Q criterion, we can see that arriving at these values requires fine-tuning of the threshold parameter which does not follow any identifiable pattern.

In order to quantify the spread of the vortex core detection of these local detection methods, we compute the root mean square error of all predicted vortices to the closest actual vortex that is

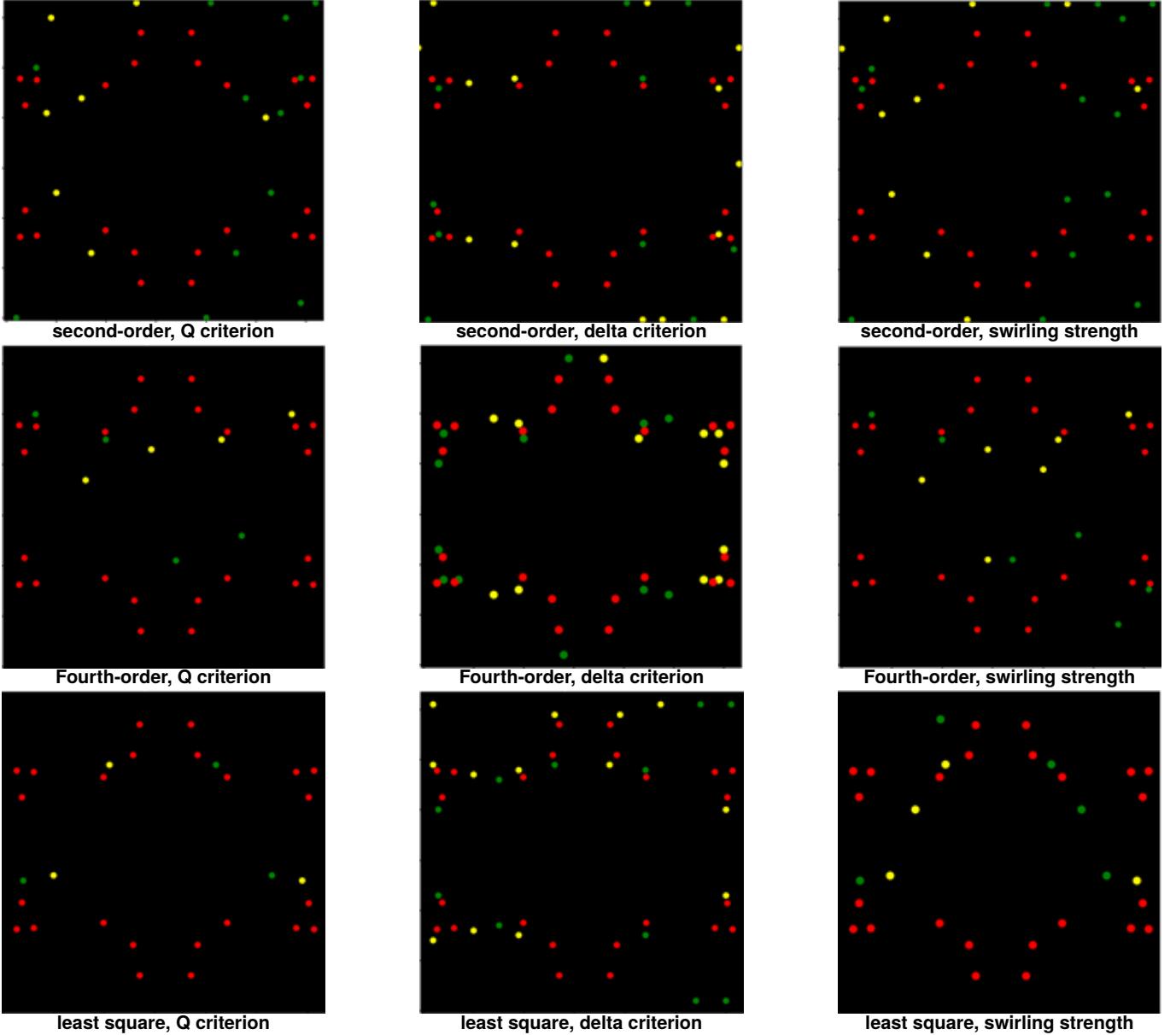


FIGURE 8: DETECTED VORTICES (SHOWN ON YELLOW AND GREEN) USING THE Q, DELTA, AND SWIRLING STRENGTH CRITERION WITH SECOND-ORDER, FOURTH-ORDER AND LEAST SQUARE GRADIENT RECONSTRUCTION SCHEMES FOR THE SECOND TEST IMAGE (24 VORTICES). THE ACTUAL VORTEX CORE REGIONS ARE SHOWN BY THE RED CIRCLES.

present in the domain using

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=0}^{n-1} (x_i - \hat{x}_i)^2}, \quad (22)$$

Where n is the total number of predicted vortices, x_i the predicted vortex core and \hat{x}_i being the vortex that is present in the domain and closest to x_i . Here, x_i represents the yellow and green dots in Figure 7 and Figure 8, while \hat{x}_i is represented by the red dots. The results are shown in Table 4. Interestingly, this Table shows that the smallest and highest errors are obtained using the delta criterion using the second- and fourth-order gradient reconstruction scheme for test images 1 and 2, respectively. This

is confirmed by taking the variance of all the results for each method where we have a variance of $5.5e-3$, $7.9e-4$, and $6.4e-4$ for the delta, Q, and swirling strength criterion, respectively. While the delta criterion does indeed provide an overall good prediction of the vortex core regions for test image 2, its prediction for the first test image is rather scattered. We can see that the swirling strength does not provide overall the lowest errors, but it is the most consistent method which has the lowest variance. The Q criterion performs similarly to the swirling strength in terms of overall errors and achieves this without the need to fine-tune the threshold parameter. Therefore, the Q criterion may be seen as a good compromise which may explain its popularity in the computational fluid dynamics community to visualise vortices, however, it still produces false positives and negatives.

TABLE 4: ROOT MEAN SQUARE ERRORS FOR THE PREDICTED VORTEX CORES TO THE ACTUAL VORTEX CORE REGIONS.

Test image	Root Mean Square Error values								
	Delta Criterion			Q Criterion			swirling strength Criterion		
	Second Order	Fourth Order	Least Square	Second Order	Fourth Order	Least Square	Second Order	Fourth Order	Least Square
1	0.275	0.1112	0.1347	0.1457	0.1603	0.158	0.1503	0.1603	0.1589
2	0.1419	0.0535	0.1115	0.1384	0.1046	0.0930	0.1389	0.1092	0.1022

TABLE 5: MAXIMUM DIFFERENCE BETWEEN THE PREDICTED VORTEX CORES AND THE ACTUAL VORTEX CORE REGIONS.

Test image	Maximum Difference values								
	Delta Criterion			Q Criterion			swirling strength Criterion		
	Second Order	Fourth Order	Least Square	Second Order	Fourth Order	Least Square	Second Order	Fourth Order	Least Square
1	1.129	0.919	1.089	1.080	0.989	1.025	1.080	1.080	1.025
2	1.190	1.016	1.161	1.190	1.0126	0.958	1.190	1.045	0.958

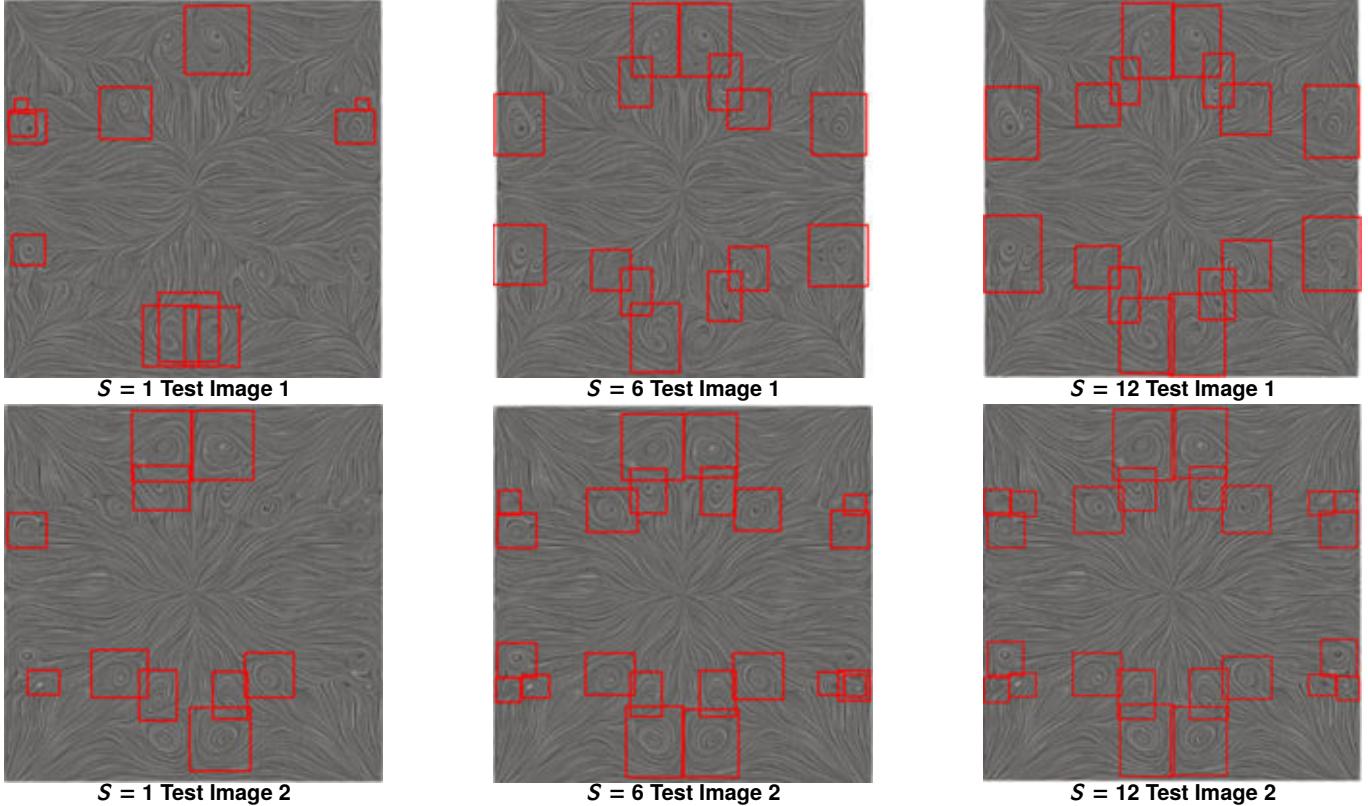


FIGURE 9: THE OUTPUT RESULT OF S=1 (1 IMAGE), S=6 (40 IMAGES) AND S=12 (100 IMAGES). THE RED BOX IS THE PREDICTED BOUNDING BOX BASED ON BOUNDING BOXES THAT ARE ALLOWED TO OVERLAP DURING LABELLING, SEE SECTION 3.5.

For each criterion, the maximum distance between the actual vortex cores and the predicted location is shown in Table 5 with the second-order, fourth-order, and least square gradient schemes. The maximum difference between all predicted vortices and the actual vortices in the domain is calculated as follows:

$$MaxDiff = \text{Max} \left(\sqrt{\Delta x_i^2 + \Delta y_i^2} \right) \quad i = 1, 2, \dots, n, \quad (23)$$

Δx_i represents the difference between the predicted and actual of a vortex core in x direction, while Δy_i represents the difference between the predicted and actual of a vortex core in the y direction. Based on Table 5, the Q criterion with the second-order scheme has the highest maximum distance value for test image one, as well as the highest RMSE value on the table 4. In test image 2, the highest maximum distance value is obtained with the second-order scheme of Q, delta, and swirling strength cri-

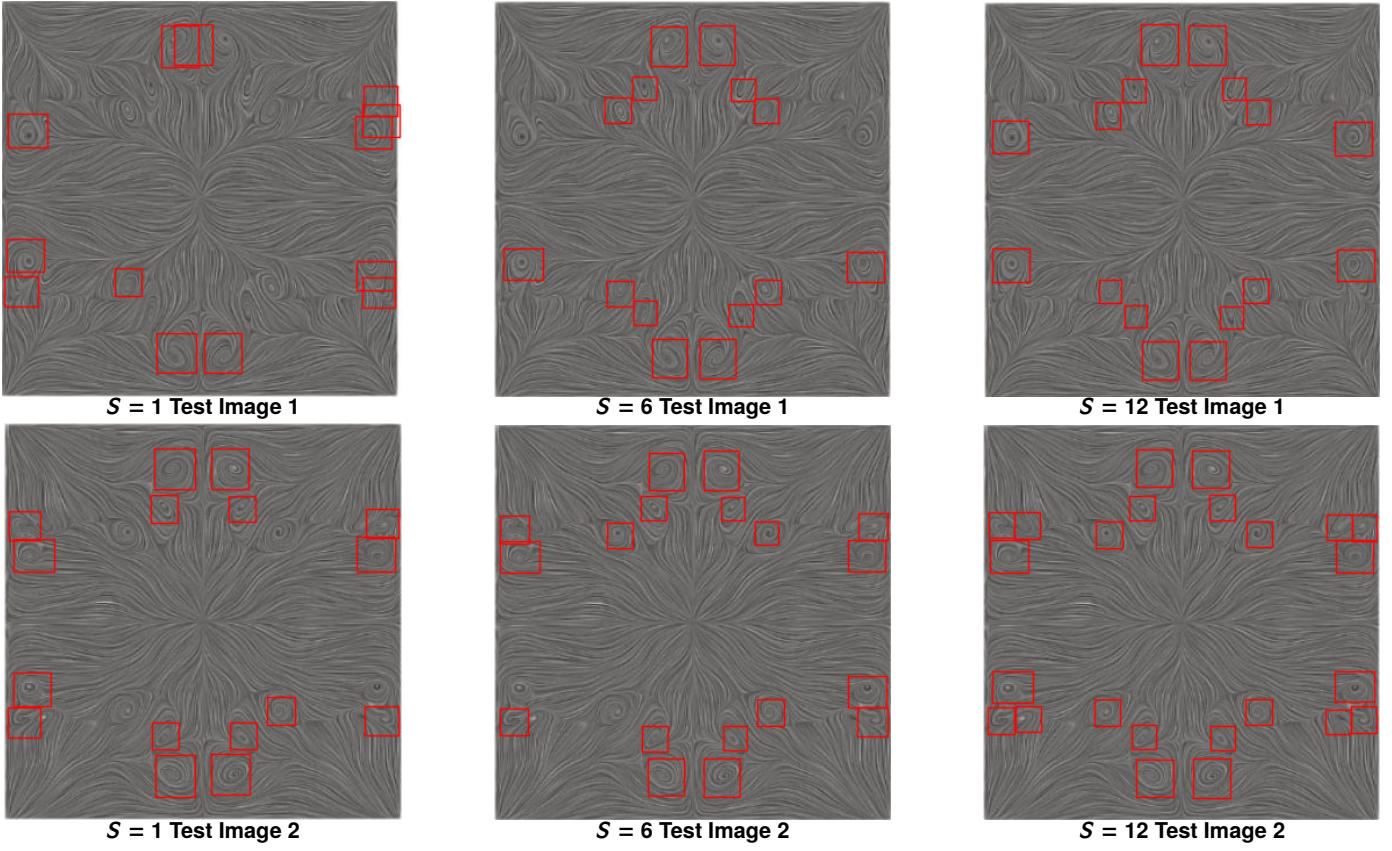


FIGURE 10: THE OUTPUT RESULT OF S=1 (1 IMAGE), S=6 (40 IMAGES) AND S=12 (100 IMAGES). THE RED BOX IS THE PREDICTED BOUNDING BOX BASED ON BOUNDING BOXES THAT ARE NOT ALLOWED TO OVERLAP DURING LABELLING, SEE SECTION 3.5.

terion. Table 4 also shows they have the highest RMSE value. The spread between the data is low, suggesting that all methods provide a similar difference in terms of maximum difference to the actual vortex cores. Next, we show that using computer vision can remove the need to specify this free parameter while reducing the false positives and negatives entirely.

5.2 Detecting vortices through computer vision

As we noted in the introduction, we focus here on predicting vortex regions (i.e. bounding boxes of vortices) and not the vortex cores themselves. This distinction is important as the test case that we employ is unsteady. For unsteady flows, the vortex core may not be inferred from streamline such as LIC surface streamline plots but rather path-lines must be studied. However, while the centre of rotation may be different, the vortex region remains approximately the same. Thus, by focusing on the bounding box of the vortex, we do not limit ourselves to studying only steady-state flows. This has the advantage that we can use simple geometries such as the Taylor-Green vortex problem which can produce vortices easily and in a wide range of sizes and shapes. In order to show how we are able to reduce false positives and negatives using computer vision, we train our YOLO-based CNN in a sequence of 12 steps, each sequence containing an increasing number of images used for labelling of vortices which are then used for training. The first sequence contains a single image for training which is then increased to 5, 10, 20, 30, 40, 50, 60, 70,

80, 90, and 100 images. The number of vortices that are manually labelled in each sequence are 18, 106, 202, 394, 562, 722, 926, 1086, 1246, 1326, 1486 and 1646, respectively. Using both test images, Figures 9 and figures 10 depict the evolution of training with overlapping and without overlapping bounding box labelling techniques, respectively, for sequence 1 (1 image), sequence 6 (40 images), and sequence 12 (100 images). The detected vortices are identified through a corresponding red bounding box. Our results indicate that whichever labelling method is used has no effect. Hence, we will discuss the overlapping bounding boxes as shown in figure 9 in the rest of this section since the bounding box labelling captures the entire vortex region, and the labelling way doesn't influence the results. Focusing on sequence 1 for the first test image (top left) in figure 9, we can see that a total of 11 vortices are predicted. 4 out of these 11 vortices are false positives and should be removed while there are 9 false negatives, i.e. vortices that are present but not detected. Moving to the right, i.e. increasing the number of images used in the sequence, we can see that we detect increasingly more vortices for sequences 6 and 12 and that we reduce the number of false positives and negatives entirely with a sufficiently large training set. Similar observations can be made as well for the second test image. Table 6 and Table 7 provide tabulated data about the reduction in false positives and negatives for each sequence of images. We can see that for test image 1 in Table 6 that it takes 60 images for training to reliably detected all physical vortices without detecting

any spurious vortices. For test image 2 in Table 7, we can see that it takes the full training set of 100 images to reduce all false positives and detect all physically present vortices correctly. This trend is shown also in Figure 11, where we plot the false negatives against the number of images used for training. We can see a monotonic decrease. We train our CNN on a single Nvidia V100 GPU and provide training times for both tables. Since both test images rely on the same 100 images that we use for training, we perform this step only once and hence have the same training times in both tables, here provided only for completeness. Table 6 and Table 7 also provide the mean Average Precision (mAP) and we train our model for each sequence until we reach a value of 0.9 except for the last sequence, where we increase the value to 0.96 to capture all vortices. Figure 12 shows that a few epochs are required initially to pick up any vortices after which we can observe a steady increase in the mAP value.

To obtain the mAP we need to calculate the Precision (measures the accuracy of the predictions) and Recall (measures how well all the true positives are found), which are defined as

$$\text{Precision} = \frac{\text{True positives}}{(\text{True positives} + \text{False positives})}, \quad (24)$$

and

$$\text{Recall} = \frac{\text{True positives}}{(\text{True positives} + \text{False negatives})}. \quad (25)$$

We count true positives as those identified vortices (red bounding boxes) that share an intersection over union (IoU) with the ground truth bounding box (the bounding box that is manually created to label vortices) which is above an IoU threshold value of 0.5 in our case. False positives and negatives take the same definition as in Table 6 and Table 7. We obtain Precision and Recall using IoU values for a given IoU threshold. Then we calculate the average of the area under the precision and recall curve.

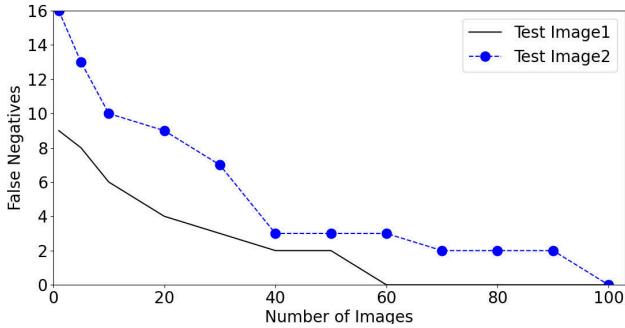


FIGURE 11: (THE BLACK SOLID AND BLUE DASH) FALSE NEGATIVES CURVES FOR TEST IMAGE 1 AND TEST IMAGE 2 RESPECTIVELY IN THE NUMBER OF IMAGES.

The error between the target value and the predicted value is calculated by the loss function, using the concept of error back-propagation in the neural network and altering the weight of each layer in the network to achieve the training of the model. During training, we calculate the loss function by summing the localisation loss (errors between the predicted bounding box and the

truth bounding box), the classification loss, and the confidence, as defined in [6, 50]. Figure 13 shows a monotonic decreasing loss function and we can say that after about 300 epochs, that our CNN is well fitted.

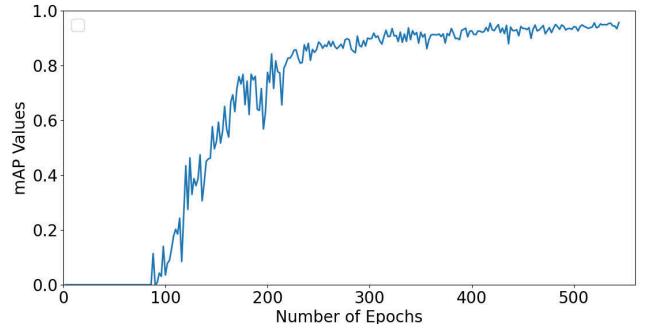


FIGURE 12: MEAN AVERAGE PRECISION CURVE IN THE TIME OF TRAINING EPOCHS

5.3 Application to turbulent boundary layer flows

Thus far, we have concentrated on homogeneously decaying turbulence in a box in Section 5.1 and Section 5.2. However, real fluid-dynamics applications feature a range of complex flow structures, in which hair-pin vortices, dean vortices, Görtler-Taylor vortices, jets, and streaks can occur (to name but a few). Therefore, the purpose of this section is to explore how our developed framework performs in cases with such structures. We investigate a case in which a turbulent boundary layer is developed over a flat plate that does feature, for example, streaks and hairpin vortices. Here, we use the LES data of Szöke et al. [51], where the turbulence within a channel with streamwise rods was simulated to investigate their role to attenuate pressure fluctuations near the wall. As the turbulent boundary layer is forming along the wall, we expect a flow structure to develop which our Taylor-Green Vortex setup did not contain. Thus, our trained framework has not seen any such structures during training. However, their vortical appearance may be similar enough so that vortices will be detected, regardless of their underlying shape. This investigation is performed to provide evidence of the universal nature of this framework.

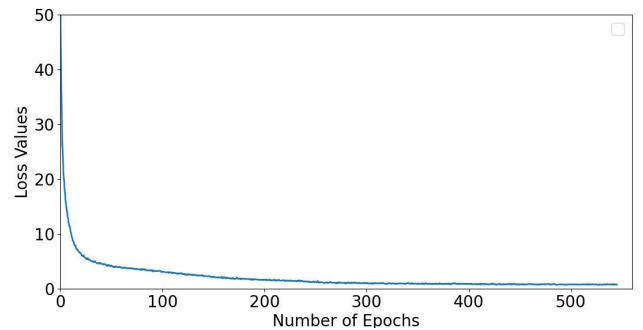


FIGURE 13: LOSS VALUES CURVE IN THE TIME OF TRAINING EPOCHS.

TABLE 6: NUMBER OF DETECTED AND EXPECTED VORTICES FOR THE FIRST TEST IMAGE, AS WELL AS FALSE POSITIVES AND NEGATIVES WITH AN INCREASING NUMBER OF IMAGES USED FOR TRAINING. THE COMPUTATIONAL TIME TAKEN FOR TRAINING IS PROVIDED ALONG WITH THE MEAN AVERAGE PRECISION (mAP)

S	Number of images	Number of vortices labelled	Expected vortices (EV)	Vortices detected	False positives	False negatives	Time of training	mAP
1	1	18	16	11	4	9	01:18:45	0.96
2	5	106	16	11	3	8	02:34:42	0.91
3	10	202	16	12	2	6	04:07:39	0.93
4	20	394	16	13	1	4	08:18:26	0.91
5	30	562	16	14	1	3	08:50:05	0.90
6	40	722	16	14	0	2	11:16:21	0.91
7	50	926	16	15	1	2	11:41:32	0.91
8	60	1086	16	16	0	0	12:05:55	0.90
9	70	1246	16	16	0	0	13:13:54	0.91
10	80	1326	16	16	0	0	13:50:58	0.92
11	90	1486	16	16	0	0	14:20:29	0.91
12	100	1646	16	16	0	0	22:00:38	0.96

TABLE 7: NUMBER OF DETECTED AND EXPECTED VORTICES FOR THE SECOND TEST IMAGE, AS WELL AS FALSE POSITIVES AND NEGATIVES WITH AN INCREASING NUMBER OF IMAGES USED FOR TRAINING. THE COMPUTATIONAL TIME TAKEN FOR TRAINING IS PROVIDED ALONG WITH THE MEAN AVERAGE PRECISION (mAP)

S	Number of images	Number of vortices labelled	Expected vortices (EV)	Vortices detected	False positives	False negatives	Time of training	mAP
1	1	18	24	12	4	16	01:18:45	0.96
2	5	106	24	14	3	13	02:34:42	0.91
3	10	202	24	17	3	10	04:07:39	0.93
4	20	394	24	17	2	9	08:18:26	0.91
5	30	562	24	18	1	7	08:50:05	0.90
6	40	722	24	22	1	3	11:16:21	0.91
7	50	926	24	22	1	3	11:41:32	0.91
8	60	1086	24	21	0	3	12:05:55	0.90
9	70	1246	24	22	0	2	13:13:54	0.91
10	80	1326	24	23	1	2	13:50:58	0.92
11	90	1486	24	24	2	2	14:20:29	0.91
12	100	1646	24	24	0	0	22:00:38	0.96

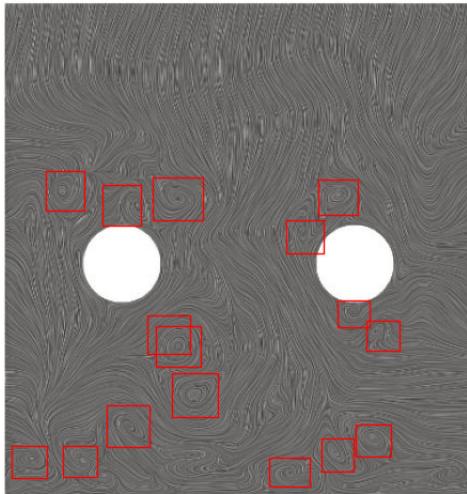


FIGURE 14: DETECTED VORTEX CORE REGIONS USING OUR COMPUTER VISION APPROACH FOR THE TURBULENT BOUNDARY LAYER CASE TAKEN FROM SZÖKE ET AL. [51]

Figure 14 shows the streamlines taken from the numerical data in [51]. This is looking straight onto the rods, where the bottom part is assigned a wall boundary condition (and hence this is where fundamentally different vortical structures are expected). Both the left and right sides are assigned periodic boundary conditions and the top is a slip wall. We use here our trained CNN and we can see that the vortices are still detected with good accuracy. We can spot at least one false negative below the right rod and at least one false positive in the bottom left corner, however, overall we are able to predict the correct vortex region and using additional images from turbulent boundary layer flows, it is likely that we will be able to remove false positives and negatives entirely. In addition, using different types of flow which produce different types of vortices will reduce the risk of over-fitting. In our case, the Taylor-Green Vortex problem produces a wide range of vortices but additional training would benefit the overall robustness.

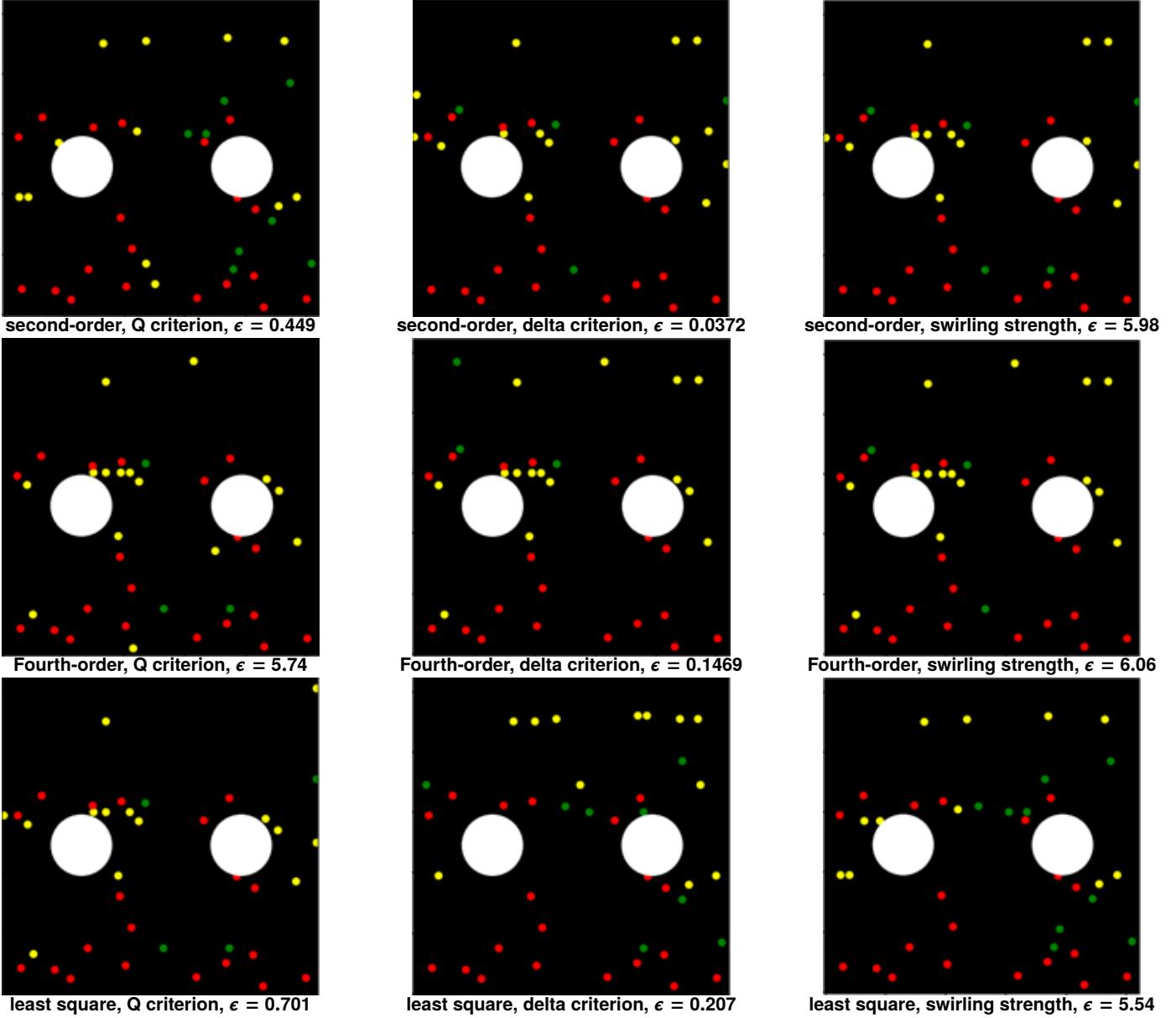


FIGURE 15: DETECTED VORTEX CORES USING THE Q, DELTA AND SWIRLING STRENGTH CRITERION WITH DIFFERENT GRADIENT RECONSTRUCTION SCHEMES FOR THE TURBULENT BOUNDARY LAYER CASE TAKEN FROM SZÖKE ET AL. [51]

for a different type of flows here.

For completeness, we also employ the Q, delta, and swirling strength criterion which is shown in Figure 15. Similar to observations made before, we can see that the results are scattered with false positives and negatives and that we have to fine-tune the threshold parameter. There is little coherence between the different methods and the computer vision-based approach provides better predictions, despite a lack of training for this particular case. Thus, these results suggest that our developed computer vision framework has a universal nature and can be applied to flow cases for which it has not been trained for. However, providing additional training would allow to detect even more features for attached, turbulent boundary layer flows and identify some vortices which are currently not detected.

6. CONCLUSIONS

We have presented a new computer vision-based approach to detect vortex cores in fluid dynamic applications through line integral convolution-based streamline plots. We used these streamline plots to train a convolutional neural network based on the YOLO algorithm. To test our new algorithm, we have performed large-eddy simulations of the Taylor-Green vortex problem at a Reynolds number of 1600 and extracted surface plots at the symmetry plane of the domain. We compute the bounding boxes of the vortex core regions and show that we are able to remove the false positives and negatives entirely for the Taylor-Green problem, regardless of the labelling procedure. We have also applied the trained network to a test case for which it has not been trained and found that most vortices are correctly detected, with only a few false positives and negatives. These

may be further removed through retraining using additional input images. We have also employed classical local vortex detection algorithms based on the Q, delta, and swirling strength criteria. We have shown that these methods are able to approximate the region of vortex cores. There are a number of false positives and negatives that are generated and the fine-tuning of a threshold parameter is required to keep the number of detected vortices bounded. Thus, local vortex detection algorithms require *a priori* knowledge of the flow field to set up the correct threshold parameter but even then produce spuriously detected vortex cores. We show that using our computer vision-based algorithm that we are able to remove these entirely.

While we are able to remove false positives and negatives, our computer vision approach only provides bounding boxes of vortex regions. To extract the precise vortex core, one could extend our approach through a hybrid computer-vision and deep neural network approach or a modified local vortex core detection algorithm that only searches in detected vortex core regions. We leave these to future work activities.

ACKNOWLEDGEMENTS

The author Hazem Ashor Amran Abolholl has received funding through the Academic Relations Department of the Libyan Embassy – London, and their financial support is greatly acknowledged.

REFERENCES

- [1] Jiang, Ming, Machiraju, Raghu and Thompson, David. “Detection and visualization of vortices.” *Visualization Handbook* (2005): pp. 295–309 DOI [10.1016/B978-012387582-2/50016-2](https://doi.org/10.1016/B978-012387582-2/50016-2).
- [2] Hunt, J C R, Wray, a a and Moin, P. “Eddies, streams, and convergence zones in turbulent flows.” *Center for Turbulence Research, Proceedings of the Summer Program* No. 1970 (1988): pp. 193–208.
- [3] Liu, Chao Qun, Wang, Yi Qian, Yang, Yong and Duan, Zhi Wei. “New omega vortex identification method.” *Science China: Physics, Mechanics and Astronomy* Vol. 59 No. 8. DOI [10.1007/s11433-016-0022-6](https://doi.org/10.1007/s11433-016-0022-6).
- [4] Chong, M. S., Perry, A. E. and Cantwell, B. J. “A general classification of three-dimensional flow fields.” *Physics of Fluids A* Vol. 2 No. 5 (1990): pp. 765–777. DOI [10.1063/1.857730](https://doi.org/10.1063/1.857730).
- [5] Zhou, J., Adrian, R. J., Balachandar, S. and Kendall, T. M. “Mechanisms for generating coherent packets of hairpin vortices in channel flow.” *Journal of Fluid Mechanics* Vol. 387 (1999): pp. 353–396. DOI [10.1017/S002211209900467X](https://doi.org/10.1017/S002211209900467X).
- [6] Redmon, Joseph, Divvala, Santosh, Girshick, Ross and Farhadi, Ali. “You only look once: Unified, real-time object detection.” *Proceedings of the IEEE conference on computer vision and pattern recognition*: pp. 779–788. 2016.
- [7] Ni, Zihan, Chen, Jia, Sang, Nong, Gao, Changxin and Liu, Leyuan. “LIGHT YOLO FOR HIGH-SPEED GESTURE RECOGNITION Key Laboratory of Ministry of Education for Image Processing and Intelligent Control , School of Automation , Huazhong University of Science and Technology , Wuhan 430074 , China . NERCEL , Central China Norma.” *2018 25th IEEE International Conference on Image Processing (ICIP)* (2018): pp. 3099–3103.
- [8] Jiang, Mingxin, Hai, Tao, Pan, Zhigeng, Wang, Haiyan, Jia, Yinjie and Deng, Chao. “Multi-Agent Deep Reinforcement Learning for Multi-Object Tracker.” *IEEE Access* Vol. 7 (2019): pp. 32400–32407. DOI [10.1109/ACCESS.2019.2901300](https://doi.org/10.1109/ACCESS.2019.2901300).
- [9] Boudjit, Kamel and Ramzan, Naeem. “Human detection based on deep learning YOLO-v2 for real-time UAV applications.” *Journal of Experimental and Theoretical Artificial Intelligence* Vol. 00 No. 00 (2021): pp. 1–18. DOI [10.1080/0952813X.2021.1907793](https://doi.org/10.1080/0952813X.2021.1907793). URL <https://doi.org/10.1080/0952813X.2021.1907793>.
- [10] Lu, Junyan, Ma, Chi, Li, Li, Xing, Xiaoyan, Zhang, Yong, Wang, Zhigang and Xu, Jiuwei. “A Vehicle Detection Method for Aerial Image Based on YOLO.” *Journal of Computer and Communications* Vol. 06 No. 11 (2018): pp. 98–107. DOI [10.4236/jcc.2018.611009](https://doi.org/10.4236/jcc.2018.611009).
- [11] Farhadi, Ali and Redmon, Joseph. “Yolov3: An incremental improvement.” *Computer Vision and Pattern Recognition*: pp. 1804–2767. 2018. Springer Berlin/Heidelberg, Germany.
- [12] Wu, Xiaohua and Moin, Parviz. “Direct numerical simulation of turbulence in a nominally zero-pressure-gradient flat-plate boundary layer.” *Journal of Fluid Mechanics* Vol. 630 (2009): pp. 5–41. DOI [10.1017/S0022112009006624](https://doi.org/10.1017/S0022112009006624).
- [13] Kenwright, David and Haimes, Robert. “Vortex identification - applications in aerodynamics: a case study.” *Proceedings of the IEEE Visualization Conference* (1997): pp. 413–415 DOI [10.1109/visual.1997.663910](https://doi.org/10.1109/visual.1997.663910).
- [14] Kohler, Benjamin, Gasteiger, Rocco, Preim, Uta, Theisel, Holger, Gutberlet, Matthias and Preim, Bernhard. “Semi-automatic vortex extraction in 4D PC-MRI cardiac blood flow data using line predicates.” *IEEE Transactions on Visualization and Computer Graphics* Vol. 19 No. 12 (2013): pp. 2773–2782. DOI [10.1109/TVCG.2013.189](https://doi.org/10.1109/TVCG.2013.189).
- [15] Oeltze-Jafra, Steffen, Cebral, Juan R., Janiga, Gábor and Preim, Bernhard. “Cluster Analysis of Vortical Flow in Simulations of Cerebral Aneurysm Hemodynamics.” *IEEE Transactions on Visualization and Computer Graphics* Vol. 22 No. 1 (2016): pp. 757–766. DOI [10.1109/TVCG.2015.2467203](https://doi.org/10.1109/TVCG.2015.2467203).
- [16] Gutak, A. D. “Experimental investigation and industrial application of Ranque-Hilsch vortex tube.” *International Journal of Refrigeration* Vol. 49 No. 0 (2015): pp. 93–98. DOI [10.1016/j.ijrefrig.2014.09.021](https://doi.org/10.1016/j.ijrefrig.2014.09.021). URL <http://dx.doi.org/10.1016/j.ijrefrig.2014.09.021>.
- [17] Epps, Brenden P. “Review of vortex identification methods.” *AIAA SciTech Forum - 55th AIAA Aerospace Sciences Meeting* No. January. DOI [10.2514/6.2017-0989](https://doi.org/10.2514/6.2017-0989).
- [18] Förste, J. “Lugt, H. J., Vortex Flow in Nature and Technology. New York et al., John Wiley & Sons 1983. XV, 297 S., zahlr. Abb., £ 47.45. ISBN 0-471-86925-2.” *ZAMM - Journal of Applied Mathematics*

- and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik* Vol. 64 No. 11 (1984): p. 502. DOI <https://doi.org/10.1002/zamm.19840641116>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/zamm.19840641116>.
- [19] Robinson, S K. “Coherent Motions in the Turbulent Boundary Layer.” *Annual Review of Fluid Mechanics* Vol. 23 No. 1 (1991): pp. 601–639. DOI [10.1146/annurev.fl.23.010191.003125](https://doi.org/10.1146/annurev.fl.23.010191.003125). URL <https://doi.org/10.1146/annurev.fl.23.010191.003125>.
- [20] Sadarjoen, I. Ari, Post, Frits H., Ma, Bing, Banks, David C. and Pagendarm, Hans Georg. “Selective visualization of vortices in hydrodynamic flows.” *Proceedings of the IEEE Visualization Conference* Vol. 98 (1998): pp. 419–422. DOI [10.1109/visual.1998.745333](https://doi.org/10.1109/visual.1998.745333).
- [21] Haller, G., Hadjighasem, A., Farazmand, M. and Huhn, F. “Defining coherent vortices objectively from the vorticity.” *Journal of Fluid Mechanics* Vol. 795 (2016): pp. 136–173. DOI [10.1017/jfm.2016.151](https://doi.org/10.1017/jfm.2016.151). URL [1506.04061](https://doi.org/10.1017/jfm.2016.151).
- [22] Serra, Mattia and Haller, George. “Objective eulerian coherent structures.” *Chaos* Vol. 26 No. 5. DOI [10.1063/1.4951720](https://doi.org/10.1063/1.4951720). URL [1512.02112](https://doi.org/10.1063/1.4951720), URL <http://dx.doi.org/10.1063/1.4951720>.
- [23] Zhang, Yuning, Liu, Kaihua, Xian, Haizhen and Du, Xiaozhe. “A review of methods for vortex identification in hydroturbines.” *Renewable and Sustainable Energy Reviews* Vol. 81 No. March 2017 (2018): pp. 1269–1285. DOI [10.1016/j.rser.2017.05.058](https://doi.org/10.1016/j.rser.2017.05.058).
- [24] Li, Matthew and McComb, Christopher. “Using Physics-Informed Generative Adversarial Networks to Perform Super-Resolution for Multiphase Fluid Simulations.” *Journal of Computing and Information Science in Engineering* Vol. 22 No. 4. DOI [10.1115/1.4053671](https://doi.org/10.1115/1.4053671). URL https://asmedigitalcollection.asme.org/computingengineering/article-pdf/22/4/044501/6850327/jcise_22_4_044501.pdf, URL <https://doi.org/10.1115/1.4053671.044501>.
- [25] Ayli, Ulku Ece, Kocak, Eyup and Turkoglu, Hasmet. “MACHINE LEARNING BASED DEVELOPING FLOW CONTROL TECHNIQUE OVER CIRCULAR CYLINDERS.” *Journal of Computing and Information Science in Engineering* (2022): pp. 1–19 DOI [10.1115/1.4054689](https://doi.org/10.1115/1.4054689). URL <https://asmedigitalcollection.asme.org/computingengineering/article-pdf/doi/10.1115/1.4054689/6884286/jcise-22-1024.pdf>, URL <https://doi.org/10.1115/1.4054689>.
- [26] Warey, Alok, Raul, Vishal, Kaushik, Shailendra, Han, Taeyoung and Chakravarty, Rajan. “Generative Inverse Design of Aerodynamic Shapes Using Conditional Invertible Neural Networks.” *Journal of Computing and Information Science in Engineering* (2022): pp. 1–18 DOI [10.1115/1.4054715](https://doi.org/10.1115/1.4054715). URL <https://asmedigitalcollection.asme.org/computingengineering/article-pdf/doi/10.1115/1.4054715/6884933/jcise-22-1055.pdf>, URL <https://doi.org/10.1115/1.4054715>.
- [27] Nabian, Mohammad Amin and Meidani, Hadi. “Physics-Driven Regularization of Deep Neural Networks for Enhanced Engineering Design and Analysis.” *Journal of Computing and Information Science in Engineering* Vol. 20 No. 1. DOI [10.1115/1.4044507](https://doi.org/10.1115/1.4044507). URL https://asmedigitalcollection.asme.org/computingengineering/article-pdf/20/1/011006/6437409/jcise_20_1_011006.pdf, URL <https://doi.org/10.1115/1.4044507>. 011006.
- [28] Samtaney, Ravi, Silver, Deborah, Zabusky, Norman and Cao, Jim. “Visualizing Features and Tracking Their Evolution.” *Computer* Vol. 27 No. 7 (1994): pp. 20–27. DOI [10.1109/2.299407](https://doi.org/10.1109/2.299407).
- [29] Hsu, William M, Corporation, Digital Equipment and Hunt, Neil. “SIGGRAPH ’91 Workshop Report Integrating Computer Graphics , Computer Vision , and Image Processing in Scientific Applications Ingrid Carlbom , Digital Equipment Corporation Indranil Chakravarty , Schlumberger Laboratory for Computer Science.” *Computer* (1992): pp. 8–17.
- [30] Silver, Deborah and Wang, Xin. “Tracking and visualizing turbulent 3D features.” *IEEE Transactions on Visualization and Computer Graphics* Vol. 3 No. 2 (1997): pp. 129–141. DOI [10.1109/2945.597796](https://doi.org/10.1109/2945.597796).
- [31] Abdurakipov, S., Tokarev, M., Butakov, E. and Dulin, V. “Application of computer vision and neural network analysis to study the structure and dynamics of turbulent jets.” *Journal of Physics: Conference Series* Vol. 1421 No. 1. DOI [10.1088/1742-6596/1421/1/012018](https://doi.org/10.1088/1742-6596/1421/1/012018).
- [32] Wang, Yueqing, Deng, Liang, Yang, Zhigong, Zhao, Dan and Wang, Fang. “A rapid vortex identification method using fully convolutional segmentation network.” *Visual Computer* Vol. 37 No. 2 (2021): pp. 261–273. DOI [10.1007/s00371-020-01797-6](https://doi.org/10.1007/s00371-020-01797-6). URL <https://doi.org/10.1007/s00371-020-01797-6>.
- [33] Kim, Byungsoo and Günther, Tobias. “Robust reference frame extraction from unsteady 2D vector fields with convolutional neural networks.” *Computer Graphics Forum* Vol. 38 No. 3 (2019): pp. 285–295. DOI [10.1111/cgf.13689](https://doi.org/10.1111/cgf.13689). URL [1903.10255](https://doi.org/10.1111/cgf.13689).
- [34] Krizhevsky, Alex, Sutskever, Ilya and Hinton, Geoffrey E. “ImageNet classification with deep convolutional neural networks.” *Communications of the ACM* Vol. 60 No. 6 (2017): pp. 84–90. DOI [10.1145/3065386](https://doi.org/10.1145/3065386).
- [35] Lawrence, Steve, Giles, C. Lee, Tsoi, Ah Chung and Back, Andrew D. “Face recognition: A convolutional neural-network approach.” *IEEE Transactions on Neural Networks* Vol. 8 No. 1 (1997): pp. 98–113. DOI [10.1109/72.554195](https://doi.org/10.1109/72.554195).
- [36] Lguensat, Redouane, Sun, Miao, Fablet, Ronan, Mason, Evan, Tandeo, Pierre and Chen, Ge. “EddyNet: A deep neural network for pixel-wise classification of oceanic eddies.” *International Geoscience and Remote Sensing Symposium (IGARSS)* Vol. 2018-July No. June 2021 (2018): pp. 1764–1767. DOI [10.1109/IGARSS.2018.8518411](https://doi.org/10.1109/IGARSS.2018.8518411). URL [1711.03954](https://doi.org/10.1109/IGARSS.2018.8518411).
- [37] Franz, Katharina, Roscher, Ribana, Milioto, Andres and Wenzel, Susanne. “OCEAN EDDY IDENTIFICATION AND TRACKING USING NEURAL NETWORKS Katharina Franz, Ribana Roscher, Andres Milioto, Susanne Wenzel, Jürgen Kusche University of Bonn, Institute of Geodesy and Geoinformation, Nussallee 15+17,

- 53115 Bonn, Germany.” (2018): pp. 1–4URL [arXiv:1803.07436v2](https://arxiv.org/abs/1803.07436v2).
- [38] Ströfer, Carlos Michelén, Wu, Jinlong, Xiao, Heng and Paterson, Eric. “Data-Driven, Physics-Based Feature Extraction from Fluid Flow Fields.” DOI [10.4208/cicp.OA-2018-0035](https://doi.org/10.4208/cicp.OA-2018-0035). URL [1802.00775](https://arxiv.org/abs/1802.00775), URL [http://arxiv.org/abs/1802.00775%0Ahttp://dx.doi.org/10.4208/cicp.OA-2018-0035](https://arxiv.org/abs/1802.00775%0Ahttp://dx.doi.org/10.4208/cicp.OA-2018-0035).
- [39] Deng, Liang, Wang, Yueqing, Liu, Yang, Wang, Fang, Li, Sikun and Liu, Jie. “A CNN-based vortex identification method.” *Journal of Visualization* Vol. 22 No. 1 (2019): pp. 65–78. DOI [10.1007/s12650-018-0523-1](https://doi.org/10.1007/s12650-018-0523-1). URL <https://doi.org/10.1007/s12650-018-0523-1>.
- [40] Patankar, Suhas. “Numerical heat transfer and fluid flow: Computational methods in mechanics and thermal science.” (1980).
- [41] DeBonis, James R. “A high-resolution capability for large-eddy simulation of jet flows.” *40th AIAA Fluid Dynamics Conference* Vol. 1 No. July (2010): pp. 1–22. DOI [10.2514/6.2010-5023](https://doi.org/10.2514/6.2010-5023).
- [42] Lesieur, Marcel and Métais, Olivier. “New trends in large-eddy simulations of turbulence.” *Annual Review of Fluid Mechanics* Vol. 28 (1996): pp. 45–82. DOI [10.1146/annurev.fl.28.010196.000401](https://doi.org/10.1146/annurev.fl.28.010196.000401).
- [43] Lindner, Guilherme, Devaux, Yann and Miskovic, Sanja. “VortexFitting: A post-processing fluid mechanics tool for vortex identification.” *SoftwareX* Vol. 12 (2020): p. 100604. DOI [10.1016/j.softx.2020.100604](https://doi.org/10.1016/j.softx.2020.100604). URL <https://doi.org/10.1016/j.softx.2020.100604>.
- [44] Cabral, Brian and Leedom, Leith. “Imaging vector fields using line integral convolution.” *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1993* (1993): pp. 263–270DOI [10.1145/166117.166151](https://doi.org/10.1145/166117.166151).
- [45] Wang, Qiwei, Bi, Shusheng, Sun, Minglei, Wang, Yuliang, Wang, Di and Yang, Shaobao. “Deep learning approach to peripheral leukocyte recognition.” *PLoS ONE* Vol. 14 No. 6 (2018): pp. 1–18. DOI [10.1371/journal.pone.0218808](https://doi.org/10.1371/journal.pone.0218808).
- [46] Nepal, Upesh and Eslamiat, Hossein. “Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs.” *Sensors* Vol. 22 No. 2. DOI [10.3390/s22020464](https://doi.org/10.3390/s22020464). URL <https://www.mdpi.com/1424-8220/22/2/464>.
- [47] Diosady, L and Murman, S. “Case 3.3: Taylor green vortex evolution.” *Case Summary for 3rd International Workshop on Higher-Order CFD Methods*. 2015.
- [48] Beck, Andrea D. and Gassner, Gregor J. “Numerical Simulation of the Taylor-Green Vortex at Re=1600 with the Discontinuous Galerkin Spectral Element Method for well-resolved and underresolved scenarios.” *50th AIAA Aerospace Sciences Meeting* (2012): pp. 1–4.
- [49] DeBonis, James R. “Solutions of the Taylor-Green vortex problem using high-resolution explicit finite difference methods.” *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition 2013*. January: pp. 1–20. 2013. DOI [10.2514/6.2013-382](https://doi.org/10.2514/6.2013-382).
- [50] Li, Jing, Gu, Jinan, Huang, Zedong and Wen, Jia. “Application research of improved YOLO V3 algorithm in PCB electronic component detection.” *Applied Sciences (Switzerland)* Vol. 9 No. 18. DOI [10.3390/app9183750](https://doi.org/10.3390/app9183750).
- [51] Szoke, Máté, Nurani Hari, Nandita, Devenport, William J., Glegg, Stewart A. and Teschner, Tom-Robin. “Flow Field Analysis Around Pressure Shielding Structures.” (2021): pp. 1–25DOI [10.2514/6.2021-2293](https://doi.org/10.2514/6.2021-2293).

,

2023-01-11

Surface line integral convolution-based vortex detection using computer vision

Abolholl, Hazem Ashor Amran

American Society of Mechanical Engineers

Abolholl HAA, Teschner T-R, Moultsas I. (2023) Surface line integral convolution-based vortex detection using computer vision. Journal of Computing and Information Science in Engineering, Volume 23, Issue 5, October 2023, Article number 051002, Paper number JCISE-22-1222

<https://doi.org/10.1115/1.4056660>

Downloaded from Cranfield Library Services E-Repository