

GPPS-TC-2024-0100

Extracting Quantitative Streamline Information from Surface Flow Visualization Images in a Linear Cascade using Convolutional Neural Networks

Xingyu Liu
Durham University
xingyu.liu2@durham.ac.uk
Durham, United Kingdom

Grant Ingram
Durham University
g.l.ingram@durham.ac.uk
Durham, United Kingdom

David Sims-Williams
Durham University
d.b.sims-williams@durham.ac.uk
Durham, United Kingdom

Toby P. Breckon
Durham University
toby.breckon@durham.ac.uk
Durham, United Kingdom

ABSTRACT

Surface flow visualization (SFV), specifically surface oil flow visualization, is an experimental technique that involves coating the surface with a mixture of oils and dyes before applying the flow to the subject. While investigating the surface flow, the surface topology must be analysed to determine the flow field near the surface. For this, numerous flow visualization and image processing techniques have been proposed, showing good performance. Nonetheless, their accuracy is largely contingent on human expertise, and the overall processing cost is elevated because they necessitate the trial-and-error optimization of thresholding parameters, which are not applicable universally across all experimental conditions.

Convolutional Neural Networks (CNN) are deep learning models designed primarily for tasks involving grid-like data, particularly image and video analysis. Inspired by the outstanding feature extraction performance of deep neural networks, in this work, we trained a CNN-based model to develop an automated streamline detection and flow field reconstruction tool which works well on surface oil flow visualization images. The accuracy of streamline detection can be customized through the use of a threshold function. The predictive outcomes of the flow field and the distribution of shear pressure will be compared to the results obtained through Computational Fluid Dynamics (CFD) in the same case.

INTRODUCTION

Throughout the historical evolution of aerodynamic and hydrodynamic research, the development of methods to visualize flow has been essential (Maltby, 1962). These techniques allow experimenters to grasp the general nature of the flow before delving into detailed measurements or mathematical analyses. This paper uses contemporary image processing and machine learning techniques applied to classical surface flow visualisation techniques to extract additional information.

PREVIOUS WORK

The role of flow visualization in experimental fluid-mechanical research has been thoroughly examined, with numerous reviews covering the entire field or specific applications (Merzkirch, 1987). Flow visualization offers precise information about flow characteristics in both spatial and temporal dimensions, enabling investigators to theorize and verify flow behaviour (Etminan et al., 2022).

As Lu (2010) emphasized in their review, establishing a coherent link between surface patterns and the overarching flow field requires a solid foundation in the theoretical underpinnings of the flow which is absent in a number of publications.

Flow visualization techniques, including methods such as surface oil flow visualization (SOFV), tuft analysis, Particle Image Velocimetry (PIV), and others, are employed to study fluid dynamics. SOFV, as an experimental approach, entails coating the surface of interest with an oil and dye mixture before subjecting it to a flow. In areas with high shear stress, the oil/dye mixture is removed, while in regions with low shear stress, it persists or accumulates. The resulting pattern is then analysed to infer the structure near the examined surface. Despite the simplicity, intuitive outcomes, and minimal experimental equipment requirements of SOFV, current research predominantly utilizes it as an indirect observational tool, emphasizing qualitative insights over quantitative analysis.

Zierke et al. (1994) used an oil-paint method to obtain patterns of skin-friction lines on the rotor blades surfaces of a high-Reynolds-number pump. The flow visualization results reveals that the existence of a trailing-edge separation vortex, which migrates radially upward along the trailing edge and then turns in the circumferential direction near the casing, moving in the opposite direction of blade rotation. The resulting patterns qualitatively confirmed the unsteadiness of the vortex and helped in establishing the trajectory of the tip leakage vortex core.

The abilities of the thin oil film technique with its application in a short duration wind tunnel can avoid the intrusiveness of the measuring device and the corresponding flow disturbance while exploring the skin friction as emphasized by Schülein (2004). For instance, Aunapu et al. (2000) compared the flow visualization results from two different established methods utilizing ink dots and solvent, and oil and black powder. The flow features captured using these techniques include the horseshoe vortex and its migration across the passage to impinge on the neighbouring blade, endwall cross flow and the endwall saddle point.

Flow visualisation images are captured using digital cameras and the subsequent digital image can be processed to extract additional information. When analysing these images topological rules are pivotal in connecting the surface footprint to the broader flow. Edge detection is a recurring task in classical computer vision processes such as segmentation (Zhang et al., 2015) and image recognition (Yang et al., 2002, Shotton et al., 2008) it is also used in contemporary tasks like image-to-image translation (Zhu et al., 2017) and photo sketching (Li et al., 2019). These techniques are indispensable for numerous applications such as that considered in this paper. Fields such as medical image analysis (Pourreza et al., 2018) or remote sensing (Isikdogan et al., 2017) heavily rely on edge detectors for core activities. Despite the substantial body of work on edge detection, it persists as an open problem with room for new contributions.

Since the introduction of the Sobel operator (Sobel, 1970), numerous edge detection techniques have been proposed, with classical approaches like Canny (Canny, 1986) remaining in use today. In the era of Deep Learning (DL), Convolutional Neural Networks (CNN) have given rise to edge detectors such as DeepEdge (Bertasius et al., 2015), HED (Xie and Tu, 2015), RCF (Liu et al., 2017), BDCN (He et al., 2019), among others. These models predict edge-maps from given images, akin to low-level methods, but with superior performance (Ziou and Tabbone, 1998). The efficacy of these methods stems from the application of CNN at different scales to a diverse set of images, coupled with training regularization techniques (Poma et al., 2020).

The genesis of the current work is a paper by Abdelsalam et al. (2017) who developed an algorithm based on edge detection to determine the orientation of vectors based on flow visualisation images. However, Abdelsalam was unable to discern the direction of the streaks and to avoid formation of scattered noise vectors. This provides the motivation for the current work: to apply more advanced CNN based edge detection methods to improve upon this work.

EXPERIMENTAL METHODS

The so-called “Durham Cascade” (Fig. 1), a low-speed linear cascade of a high pressure turbine blade was used to generate a number of flow visualisation images for the work. This cascade has been extensively used for secondary flow research and for this paper was used with and without a tip clearance.

Table 1: Key Cascade Parameters

Parameter	Unit
Inlet Flow Angle	42.75°
Exit Blade Angle	-68.70°
Blade Pitch	191 mm
Blade Axial Chord	181 mm
Tip Clearance	3.75 mm
Freestream Velocity	19.1 m/s
Dynamic Pressure	215 Pa
Re number	4 x 10 ⁵

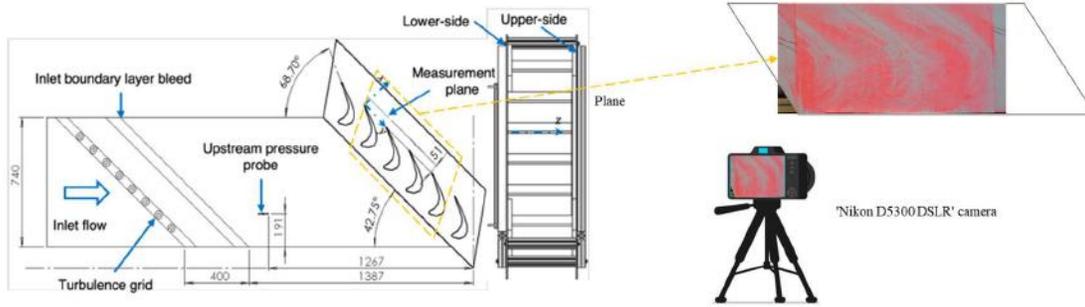


Figure 1: Durham Cascade layout third angle projection including definitions of cascade coordinates and image capture.

Upon completion of the entire experiment, the endwall would be carefully extracted for photography. High-resolution images (6000x4000 pixels) were captured using a ‘Nikon D5300 DSLR’ camera with an ‘AF-S Nikkor 35mm 1:1.8G’ lens, mitigating optical distortions. The camera was mounted on a horizontal tripod arm, and the endwall maintained upright during capture. A blend of paraffin and red UV fluorescent dye powder (5.5 ml paraffin: 1 ml dye) was uniformly applied on the endwall center, covering at least two cascade passages, using a perpendicular paintbrush orientation.

In the specific tip clearance endwall experiment, after the wind tunnel had been initiated and the flow velocity had stabilized for approximately 25 minutes, the endwall was removed from the cascade, and image acquisition was conducted. Subsequent inspection of the endwall following the experiment revealed that the majority of its surface had become dry, with minimal oil residue present. However, certain areas near the projection of the blades still retained oil deposits. It is quite evident result since it is conventionally expected that surface oil is readily entrained by the airflow along the streamlines, with less facile removal occurring at saddle points.

IMAGE PROCESSING METHOD

Subsequent to image acquisition, a multi-stage image processing pipeline implemented in software using the ‘Python’ language. was employed for image processing, utilizing functions sourced from an open-source computer vision library module named Pytorch (Paszke et al., 2019). Supplementary functions were drawn from other modules such as Numpy (Harris et al., 2020), SciPy (Virtanen et al., 2020), Matplotlib (Hunter, 2007), etc. The principal stages of the algorithm are shown in Fig. 2.

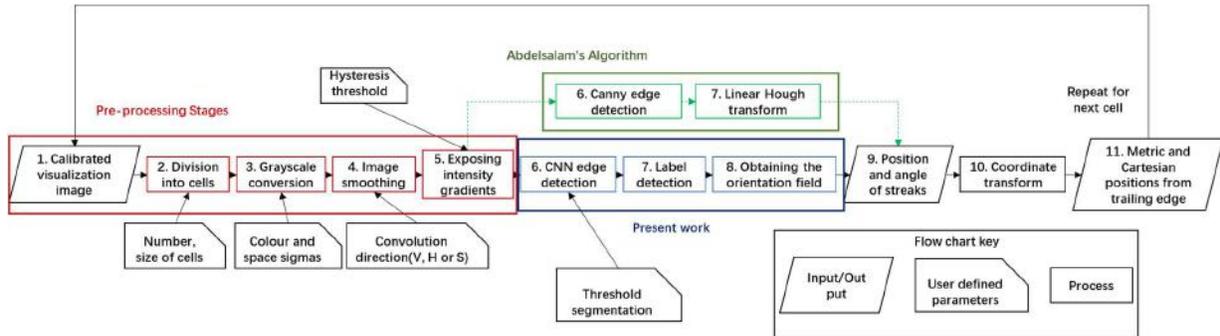


Figure 2: Flow chart of streakline detection algorithm

The algorithm is based on the processing method of Abdelsalam et al. (2017), if steps 6 and 7 were replaced with Canny Edge Detection and Linear Hough Transform, and step 8 was skipped (Green box in Figure 2), the method would be identical to Abdelsalam’s technique. Subsequently, as well as the differences in edge detection algorithms compared to Abdelsalam, this work also employs a distinct methodology for the subsequent construction of the flow field.

Pre-processing Stages (Steps 1-5)

To pre-process images, the method provided by Abdelsalam et al. (2017) was applied. The calibration phase addressed radial and tangential distortions inherent to the camera lens by estimating constant parameters through image capture of a calibration pattern, utilizing Zhang (2000). Following calibration, the visual image was partitioned into user-defined rectangular cells for grayscale conversion, with grid size influencing detection accuracy. Noise reduction was then applied using the bilateral filter (Tomasi and Manduchi, 1998), which was particularly effective in preserving edges. Finally, intensity gradients were exposed using an algorithm based on the approach of Rao and Schunck (1991) for estimating the orientation field from flow visualization images.

CNN Edge Detection (Step 6)

After calibration and smoothing, streaks were detected using a CNN based edge detection function. (Step 6 in Fig. 2). This section would introduce the proposed network architecture and give details on the different modules. The proposed model is based on Lightweight Dense Convolutional network (LDC) by Soria et al. (2022), so below we are going to present in details the proposed modifications. The LDC architecture comprises two subnets: the Dense Extreme Inception Network (Dexi) and the upsampling network (USNet). Dexi consists of 4 blocks, which acts as an encoder, and the skip-connections couple the third and fourth blocks as well as their sub-blocks. USNet, on the other hand, is a conditional CNN, which acts as a decoder and transforms feature maps into edge maps, matching the size of the input image of the Dexi subnet.

As shown in Fig. 3, each block contains two convolutional layers. The size of all convolutional kernels is 3×3 , and the number of kernels is indicated accordingly. In addition, the first block has a notation 's2', which signifies that the stride of the convolutional layer is 2. Each convolutional layer is followed by batch normalization and a Rectified Linear Unit (ReLU). Starting from block-3, the last convolutional layer of the last sub-block does not include the ReLU function. The red rectangles represent the use of 3×3 convolutional kernels with a stride of 2 in the max-pooling operation. Compared to LDC, this paper increases the number of convolutional kernels in block-4. Additionally, Inspired by Poma et al. (2020)'s work, the four intermediate edge-map predictions are fused to generate a single edge-map. The configuration of the loss function for the network mentioned in this paper is similar to that of the LDC network. Detailed information can be found in Soria et al. (2022).

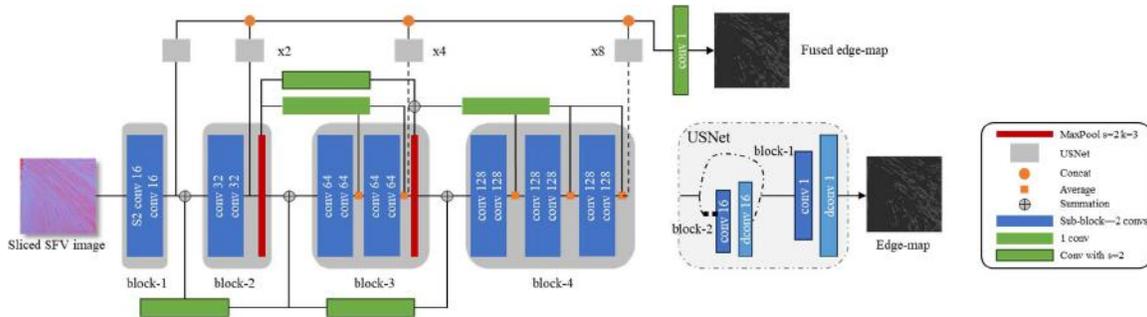


Figure 3: Architecture of the proposed CNN based model.

Due to the limited number of images obtained from the experiment, the network was first pre-trained using the BIPED dataset to better leverage its performance. The Barcelona Images for Perceptual Edge Detection (BIPED) has 250 images in high definition. The BIPED dataset has just one annotation in the edge level that has been carefully validated. The images in the dataset used in this work come from the SFV experiments described in experiment method part. On the other hand, the edge map is derived by initially applying Canny edge detection on the experimental image followed by manual removal of excessively short and overlapping streaks.

The sliced SFV images are then input into this pre-trained network, the output of which is its edge feature map. This provides all the edge information in the image and is a binary image (Show later in Fig. 5b). The values of all edge positions are 1, and the background values are 0. However, this information cannot be directly used to compute the flow field.

Label Detection (Step 7)

Labelling (also known as Connected component labelling, connected component analysis, or region labelling) represents an algorithmic implementation of graph theory employed for ascertaining the connectivity of regions resembling "blobs" within a binary image. In a binary image, such as a grayscale image, if two adjacent pixels have the same value (both 0 or 1), these two pixels are considered to be part of a mutually connected region. This relationship is transitive, and the "labelling" process involves assigning the same value to all pixels within a connected region.

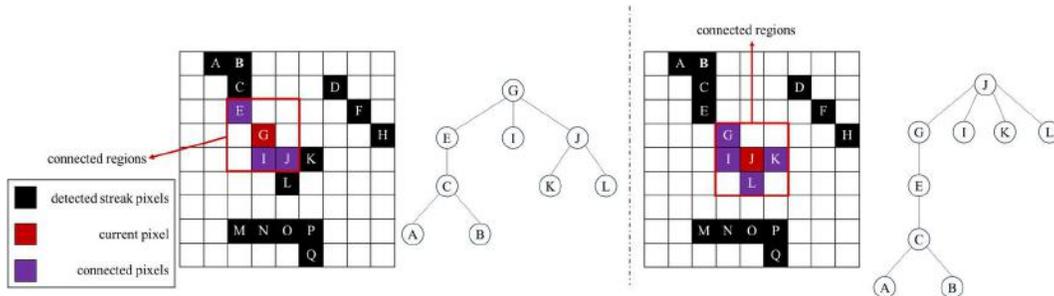


Figure 4: Label and store the detected streaks. (left) start with G; (right) start with J.

Specifically, as illustrated in Fig. 4 (for clarity, a simplified lettering scheme is used in the figure), for each pixel, the presence of other pixels within its surrounding eight cells is examined. This evaluation is recursively applied to the detected pixels until no suitable candidates remain. Throughout this process, all pixels involved are labelled as part of the same streak. The whole process can be started from any point on the streak, such as G (Fig. 4 left) or J (Fig. 4 right). Initiating the labelling process for one of the streaks, starting from G (Fig. 4 left), reveals that E, I and J are adjacent to it. Subsequently, C is adjacent to point E, while K and L is adjacent to J. Continuing in this sequential manner, a tree-like structure can be employed to record the results, ultimately resulting in the "labelling" of A, B, C, E, G, I, J, K as a cohesive entity. Suppose the process starts from J (Fig. 4 right), similarly, it would reveal that G, I, K and L are adjacent to J. Finally, a different tree-like structure with same content could be obtained. In fact, starting from any point on the streak will eventually result in the entire streak being labelled.

Fig. 5 provides a visual summary of Steps 6 and 7 applied to a single cell. The segmented grayscale image cell is shown in Fig. 5(a) and underwent edge detection to transform into the binary image shown in Fig. 5(b). Labelling was then performed on this edge feature map. Although the actual map was more intricate than the example in Fig. 4, the underlying principles remained consistent. Ultimately, each "independent" edge was labelled, and for ease of distinction, different colours are used in Fig. 5(c) to show the detected streaks.

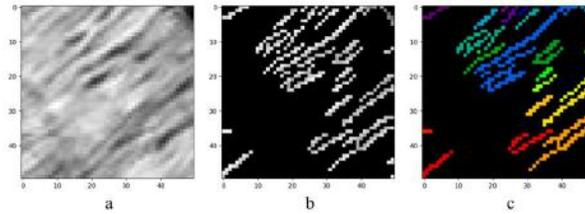


Figure 5: The label detection result from a cell of flow visualization image: (a) grey scale image; (b) image after edge detection; (c) image after labelled

Obtaining the Orientation Field (Step 8)

In Step 7, despite the extraction of all streaks, which constitutes the most direct observational data, this information cannot be directly used to generate the directional field. This limitation stems from the heterogeneous distribution and clutter of streaks throughout the image, which obscures relevant information. Additionally, on the scale of the entire image, the identified streaks form only a sparse matrix. Therefore, alternative approaches are required.

The algorithm used to estimate the orientation field from a surface oil flow visualization image relies on the gradients of the resulting images after post-processing, which is described in detail by Rao and Schunck (1991). Specifically, it targets the gradients obtained from detected streaks, aiming to reduce interference from background noise.

The gradient at a point (i, j) is represented in polar coordinates as $G_{ij}e^{i\theta_{ij}}$. In Step 7, employing label detection, one can obtain the set L of streaks present on a given cell. Therefore, for an $N * N$ cell, the dominant orientation, $\hat{\theta}_{mn}$ of its centre point (m, n) can be represented as below:

$$\hat{\theta}_{mn} = \tan^{-1} \left(\frac{\sum_{(i,j) \in L} G_{ij}^2 \sin 2\theta_{ij}}{\sum_{(i,j) \in L} G_{ij}^2 \cos 2\theta_{ij}} \right) / 2 \quad (1)$$

The estimated orientation angle is denoted as $\hat{\theta}_{mn} + \pi/2$, as the gradient vector is perpendicular to the direction of anisotropy.

A key point is that the orientation estimation algorithm yields an orientation field rather than a vector field. The flow at a specific point could correspond to either direction: θ or $\theta + \pi$. As a result, any method depending on an accurate vector field as input will inevitably fail to achieve the intended outcomes.

Direction Decision

Additional measures were required to aid in determining the specific flow direction of cells. In this work, the flow direction assessment of cells is based on the following assumptions: at the inlet of the flow field, the flow direction of cells should align with the direction of the wind tunnel's outflow; for any cell at arbitrary positions, due to the sufficiently small size of the cell, its flow direction should not exhibit significant "discontinuities" compared to neighbouring cells. For instance, in Fig 6(a), a cell with an undetermined orientation exhibits two potential directions, denoted as (b) and (c). Clearly, the predominant flow direction at this juncture is upward. Hence, direction (b) appears more plausible, while direction (c) is disregarded due to the presence of a significant "discontinuity". To quantitatively describe the coherence in the flow direction at the centre (m, n) of a given cell, consider another cell whose centre is (i, j) , where i and j are chosen so that this cell resides within a defined region W surrounding the target cell. In practice, the selected entities comprise the eight neighbouring cells surrounding the target cell. The directional relationship between two vectors can be

represented by the cosine of the angular displacement difference. The smaller the absolute difference in angular displacements, the smaller the angle between the vectors, indicating a closer alignment in direction. The measure of coherence could be defined by:

$$Score_{mn} = \sum_{(i,j) \in W} \cos|\theta_{ij} - (\varepsilon_{mn} + \theta_{mn})| \quad (2)$$

where $\varepsilon_{mn} = \begin{cases} 0 \\ \pi \end{cases}$

As a result, the problem shifts to maximizing the overall coherence across all cells, whose centre points fall within the set F , with the boundary condition being the predetermined direction at the inlet position. By solving for the maximum of (3), the distribution of the vector field can be determined.

$$\varepsilon_{mn}^{\square} = \underset{\varepsilon_{mn}}{\operatorname{argmax}} \sum_{(m,n) \in F} \sum_{(i,j) \in W} \cos|\theta_{ij} - (\varepsilon_{mn} + \theta_{mn})| \quad (3)$$

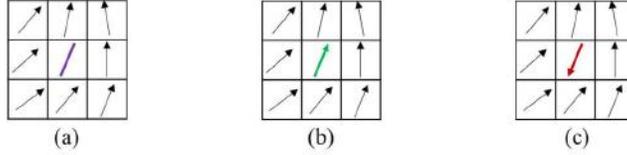
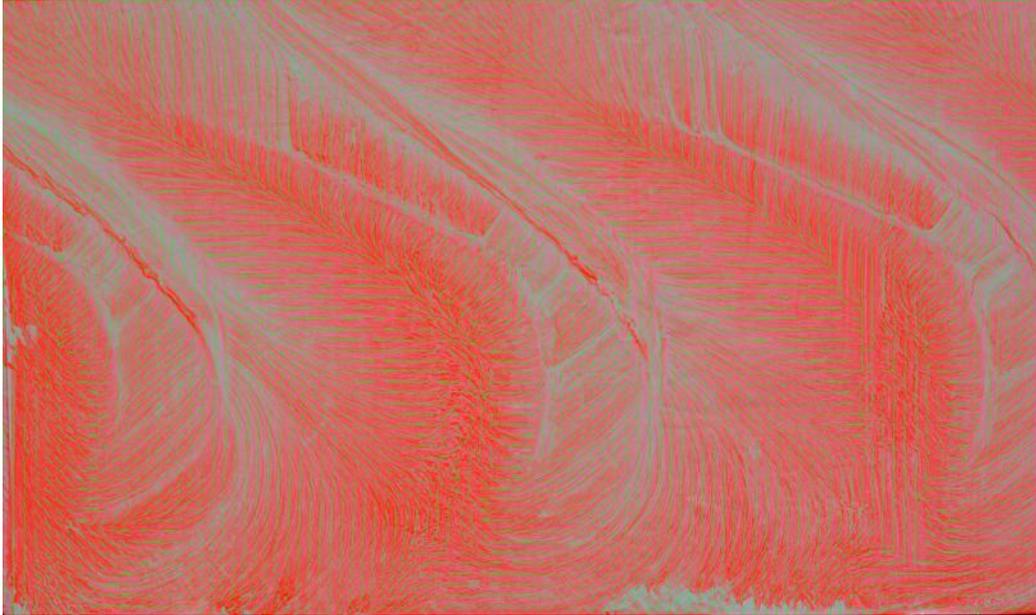


Figure 6: Possible case for ambiguity. (a) Original orientation to be inferred; (b), (c) Probable estimations

RESULT AND DISCUSSION

In order to demonstrate the improvements found in the new algorithm the images created by Abdelsalam were run both with his original algorithm. The disparities between the results obtained from the present algorithm and those of Abdelsalam original approach are as follows:

The lines detected in the flow visualisation images with a 3.75 mm tip clearance using CNN-based algorithm is shown in Fig 7 (top). The result of using linear Hough transform on the detected streaklines is exhibited in Fig. 7 (bottom). The algorithm proposed in this paper significantly mitigates the formation of noise vectors, which often manifest as seemingly vertical, horizontal, or diagonally oriented at 45° (Fig. 7 (bottom)). Moreover, the algorithm in this work calculates the average information of each labelled streak using Eq. (1) to obtain directional information, which is then redrawn as streaks. This approach differs from fitting the maximum possible line segments within cells (as shown in the right panel of Fig. 7). Therefore, it is evident that the distribution of streaks in the top panel of Fig. 7 is more uniform, with minimal intersection between streaks.



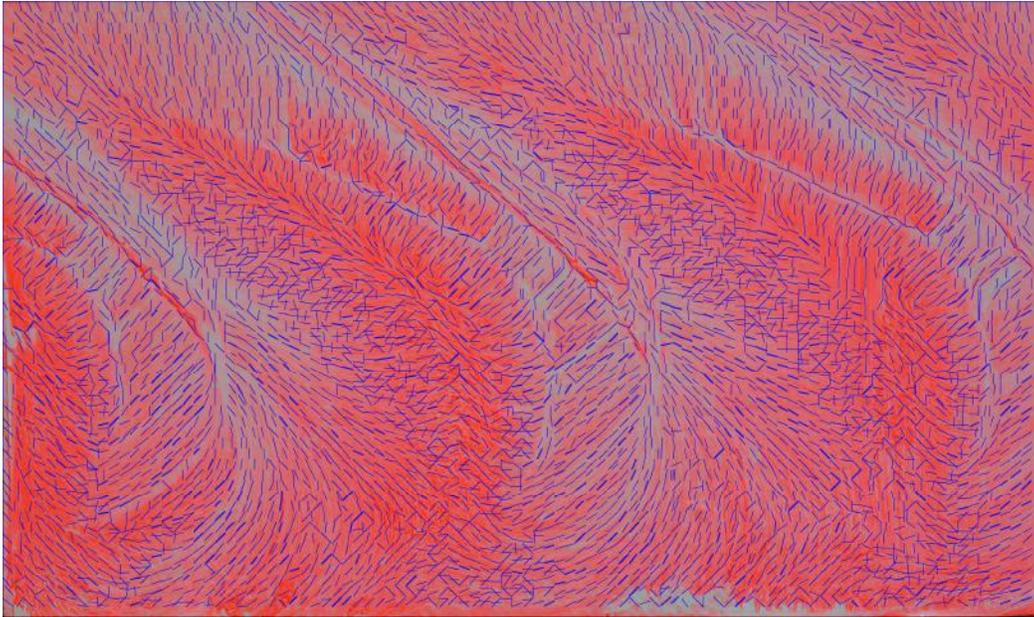


Figure 7: Image processing results; (top) CNN-based algorithm result in this work; (bottom) image processing solution from Abdelsalam et al. (2017)

Fig 8 shows the calculated streamlines from streaks using streamslice function in matlab. Precisely locating the visualization image facilitates direct plotting of blade profiles onto the image. The obtained quantitative flow field results from the input of a flow visualization photograph are notably encouraging. At the inlet, streamlines exhibit approximate uniform flow. However, in proximity to the cascade blades, the streamlines undergo separation following the expected pattern of the tip leakage vortex. A close examination of the results shows that regions within the original image displayed limited clarity which may introduce distortions into the results. In the dotted box area of Fig. 8, a number of vectors point in apparent non-physical directions at certain points. Some of these were attributed to surface irregularities or contaminants. A few anomalies were also attributed to deficiencies in the processing algorithms. In regions with complex flow conditions, especially near saddle points and areas with nearly horizontal flow, the existing direction recognition algorithm struggled to provide accurate results.

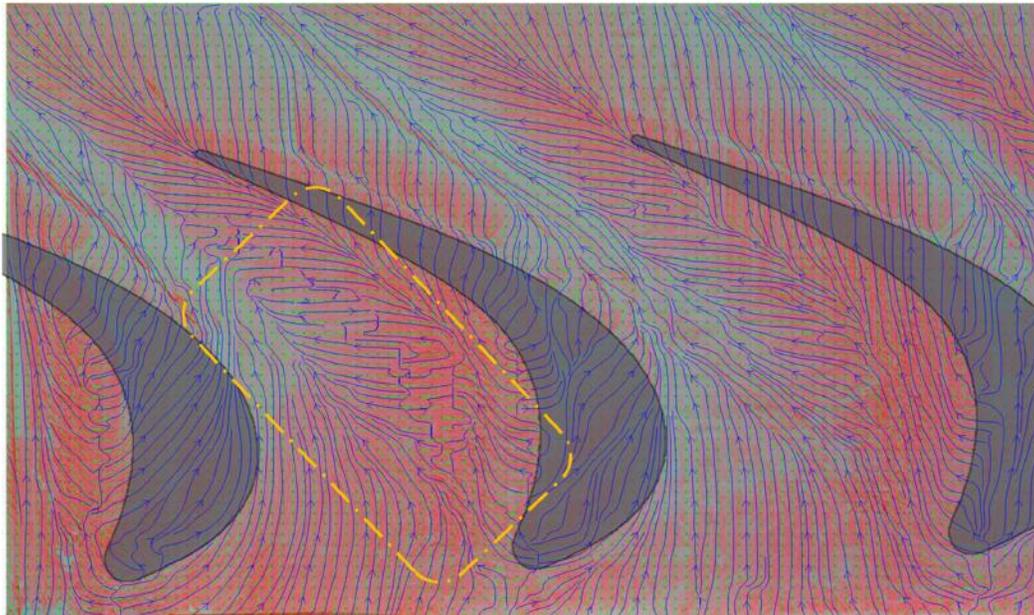


Figure 8: Calculated streamlines from streaks.

This approach was not exclusively applicable to the experimental results obtained from the cascade; in fact, it yielded promising qualitative outcomes for surface flow visualization images sourced from other sources as well.

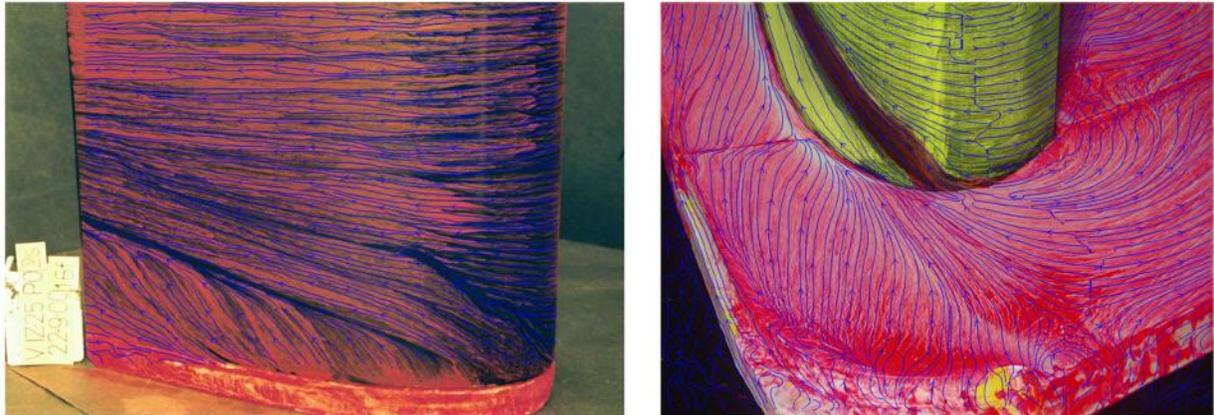


Figure 9: Calculated streamlines from streaks from other surface flow visualization images.

As shown in Fig 9, the application of this approach to surface flow visualization images sourced from other experiments resulted in flow patterns that are consistent with expert interpretation of the image in the literature (Ingram et al., 2005) and with other measurements (pressure probe, wool tufts etc) in the cascade. As well as a qualitative description of the image that aids interpretation of the flow field the technique offers quantitative data from surface oil flow visualisation.

CONCLUSION

An image processing pipeline has been enhanced to efficiently detect and extract data from endwall streaklines visualized through an oil film surface flow visualization. The optimization of the algorithm has improved its accuracy in capturing fluid dynamics features, making the extraction of information from complex flow fields more reliable. The development of this algorithm provides researchers with a powerful tool to delve into the motion and distribution of endwall streaklines, opening up new possibilities for studies in fluid dynamics and engineering.

Compared to previous work, the main advantage of the algorithm in this paper is the avoidance of scattered noise vector formation, thereby eliminating the interference of a large amount of irrelevant information on the results. This work successfully delineates the extraction of quantitative data from surface flow visualization using a CNN based image processing procedures.

The key advances show in this paper are:

- Automated Detection and Flow Field Reconstruction.
- Reduced Noise Vectors.
- Quantitative Data Extraction.

All datasets and code are available for download at https://github.com/Dehaka/Streamline_detector

REFERENCES

- ABDELSALAM, T. I., WILLIAMS, R. & INGRAM, G. 2017. Exploiting modern image processing in surface flow visualisation. Global Power and Propulsion Society (GPPS).
- AUNAPU, N. V., VOLINO, R. J. & FLACK, K. A. 2000. Surface flow visualization of a scaled-up turbine blade passage. *Measurement Science and Technology*, 11, 987.
- BERTASIUS, G., SHI, J. & TORRESANI, L. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015. 4380-4389.
- CANNY, J. 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, 679-698.
- ETMINAN, A., MUZYCHKA, Y. S., POPE, K. & NYANTEKYI-KWAKYE, B. 2022. Flow visualization: state-of-the-art development of micro-particle image velocimetry. *Measurement Science and Technology*, 33, 092002.
- HARRIS, C. R., MILLMAN, K. J., VAN DER WALT, S. J., GOMMERS, R., VIRTANEN, P., COURNAPEAU, D., WIESER, E., TAYLOR, J., BERG, S. & SMITH, N. J. 2020. Array programming with NumPy. *Nature*, 585, 357-362.
- HE, J., ZHANG, S., YANG, M., SHAN, Y. & HUANG, T. Bi-directional cascade network for perceptual edge detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019. 3828-3837.
- HUNTER, J. D. 2007. Matplotlib: A 2D graphics environment. *Computing in science & engineering*, 9, 90-95.
- INGRAM, G., GREGORY-SMITH, D. & HARVEY, N. 2005. Investigation of a novel secondary flow feature in a turbine cascade with end wall profiling. *J. Turbomach.*, 127, 209-214.

ISIKDOGAN, F., BOVIK, A. & PASSALACQUA, P. 2017. RivaMap: An automated river analysis and mapping engine. *Remote Sensing of Environment*, 202, 88-97.

LI, M., LIN, Z., MECH, R., YUMER, E. & RAMANAN, D. Photo-sketching: Inferring contour drawings from images. 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), 2019. IEEE, 1403-1412.

LIU, Y., CHENG, M.-M., HU, X., WANG, K. & BAI, X. Richer convolutional features for edge detection. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017. 3000-3009.

LU, F. 2010. Surface oil flow visualization. *The European Physical Journal Special Topics*, 182, 51-63.

MALTBY, R. 1962. Flow visualization in wind tunnels using indicators. Advisory group for aeronautical research and development paris (France).

MERZKIRCH, W. 1987. Techniques of flow visualization. Advisory group for aerospace research and development neuilly-sur-seine (France).

PASZKE, A., GROSS, S., MASSA, F., LERER, A., BRADBURY, J., CHANAN, G., KILLEEN, T., LIN, Z., GIMELSHEIN, N. & ANTIGA, L. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

POMA, X. S., RIBA, E. & SAPPA, A. Dense extreme inception network: Towards a robust cnn model for edge detection. Proceedings of the IEEE/CVF winter conference on applications of computer vision, 2020. 1923-1932.

POURREZA, R., ZHUGE, Y., NING, H. & MILLER, R. Brain tumor segmentation in MRI scans using deeply-supervised neural networks. Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: Third International Workshop, BrainLes 2017, Held in Conjunction with MICCAI 2017, Quebec City, QC, Canada, September 14, 2017, Revised Selected Papers 3, 2018. Springer, 320-331.

RAO, A. R. & SCHUNCK, B. G. 1991. Computing oriented texture fields. *CVGIP: Graphical Models and Image Processing*, 53, 157-185.

SCHÜLEIN, E. Development and Application of the Thin Oil Film Technique for Skin Friction Measurements in the Short-Duration Hypersonic Wind Tunnel. In: BREITSAMTER, C., LASCHKA, B., HEINEMANN, H.-J. & HILBIG, R., eds. New Results in Numerical and Experimental Fluid Mechanics IV, 2004// 2004 Berlin, Heidelberg. Springer Berlin Heidelberg, 407-414.

SHOTTON, J., BLAKE, A. & CIPOLLA, R. 2008. Multiscale categorical object recognition using contour fragments. *IEEE transactions on pattern analysis and machine intelligence*, 30, 1270-1281.

SOBEL, I. E. 1970. *Camera models and machine perception*, stanford university.

SORIA, X., POMBOZA-JUNEZ, G. & SAPPA, A. D. 2022. Ldc: Lightweight dense cnn for edge detection. *IEEE Access*, 10, 68281-68290.

TOMASI, C. & MANDUCHI, R. Bilateral filtering for gray and color images. Sixth international conference on computer vision (IEEE Cat. No. 98CH36271), 1998. IEEE, 839-846.

VIRTANEN, P., GOMMERS, R., OLIPHANT, T. E., HABERLAND, M., REDDY, T., COURNAPEAU, D., BUROVSKI, E., PETERSON, P., WECKESSER, W. & BRIGHT, J. 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature methods*, 17, 261-272.

XIE, S. & TU, Z. Holistically-nested edge detection. Proceedings of the IEEE international conference on computer vision, 2015. 1395-1403.

YANG, M.-H., KRIEGMAN, D. & AHUJA, N. 2002. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis & Machine Intelligence*.

ZHANG, K., ZHANG, L., LAM, K.-M. & ZHANG, D. 2015. A level set approach to image segmentation with intensity inhomogeneity. *IEEE transactions on cybernetics*, 46, 546-557.

ZHANG, Z. 2000. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22, 1330-1334.

ZHU, J.-Y., PARK, T., ISOLA, P. & EFROS, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. Proceedings of the IEEE international conference on computer vision, 2017. 2223-2232.

ZIERKE, W., FARRELL, K. & STRAKA, W. Measurements of the tip clearance flow for a high Reynolds number axial-flow rotor: Part 1—Flow visualization. Turbo Expo: Power for Land, Sea, and Air, 1994. American Society of Mechanical Engineers, V001T01A140.

ZIOU, D. & TABBONE, S. 1998. Edge detection techniques-an overview. *Pattern Recognition and Image Analysis C/C of Raspoznaniye Obrazov I Analiz Izobrazhenii*, 8, 537-559.