Experiments in Fluids (2022) 63:166
https://doi.org/10.1007/s00348-022-03516-4

RESEARCH ARTICLE

Automatic separation of laminar-turbulent flows on aircraft wings and stabilisers via adaptive attention butterfly network

Rıdvan Salih Kuzu[1] · Philipp Mühlmann[2] · Xiao Xiang Zhu[3]

Received: 6 February 2022 / Revised: 12 August 2022 / Accepted: 23 September 2022 / Published online: 18 October 2022

Abstract

Many of the laminar-turbulent flow localisation techniques are strongly dependent upon expert control even-though determining the flow distribution is the prerequisite for analysing the efficiency of wing & stabiliser design in aeronautics. Some recent efforts have dealt with the automatic localisation of laminar-turbulent flow but they are still in infancy and not robust enough in noisy environments. This study investigates whether it is possible to separate flow regions with current deep learning techniques. For this aim, a flow segmentation architecture composed of two consecutive encoder-decoder is proposed, which is called Adaptive Attention Butterfly Network. Contrary to the existing automatic flow localisation techniques in the literature which mostly rely on homogeneous and clean data, the competency of our proposed approach in automatic flow segmentation is examined on the mixture of diverse thermographic observation sets exposed to different levels of noise. Finally, in order to improve the robustness of the proposed architecture, a self-supervised learning strategy is adopted by exploiting 23.468 non-labelled laminar-turbulent flow observations.

1 Introduction

Building aircraft wings and stabilisers necessitates comprehensive design, test, and optimisation iterations in order to satisfy the expected criteria for efficiency, safety and robustness in aeronautics. Among those iterative steps, some of them are more data-centric and require profound human effort to reach a conclusion. For instance, investigation for determining laminar-turbulent boundaries is mostly dependent upon manual or semi-automated annotation of images captured from the wings and stabilisers. Thus, the main objective of this study is to seek a reliable automation method using recent artificial intelligence approaches for facilitating the localisation of laminar-turbulent flow regions on the captured measurement images.

Before elaborating on the objective, having a clear picture of high-level workflow for design, test and analysis iterations would be useful as illustrated in Fig. 1:

- In the first step, the design team devises or revises wings, stabilisers or blades.
- In the second step, a new measurement system is installed to examine the devised or revised aircraft body components. The installation environment might be in wind tunnels on the ground, or on flight as detailed in Sect. 3.1.
- In the third step, thermographic measurements are collected from the region of interest on the aircraft body, as some of the examples can be seen in Sect. 5.1.
- In the fourth step, experts in the investigation team determine the laminar-turbulent regions on the measurement images. However, those investigations might be entirely manual, or semi-automatic such that experts can draw the flow boundary lines or they can benefit from some image processing tools to determine the parameters which will be later utilised for separating the laminar-turbulent regions. Nevertheless, the main issue here, each time the

The source code of the study: https://github.com/ridvansalihkuzu/butterflynet.

✉ Rıdvan Salih Kuzu
ridvan.kuzu@dlr.de

Philipp Mühlmann
philipp.muehlmann@dlr.de

Xiao Xiang Zhu
xiaoxiang.zhu@tum.de

[1] Remote Sensing Technology Institute, German Aerospace Center (DLR), 82234 Wessling, Germany

[2] Institute of Aerodynamics and Flow Technology, German Aerospace Center (DLR), 38108 Braunschweig, Germany

[3] Department of EO Data Science, Technical University of Munich, 80333 München, Germany

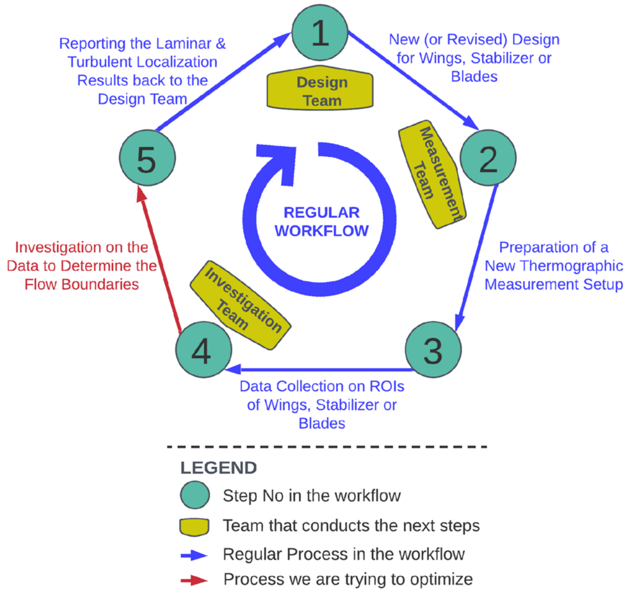
1
Design Team
New (or Revised) Design for Wings, Stabilizer or Blades
2
Measurement Team
Preparation of a New Thermographic Measurement Setup
3
Data Collection on ROIs of Wings, Stabilizer or Blades
4
Investigation Team
Investigation on the Data to Determine the Flow Boundaries
5
Reporting the Laminar & Turbulent Localization Results back to the Design Team
REGULAR WORKFLOW
LEGEND
Step No in the workflow
Team that conducts the next steps
Regular Process in the workflow
Process we are trying to optimize

Fig. 1 High-level workflow for building new wings, stabilisers or blades (Please note that the workflow ignores some of the side steps which are out of context for this study)

experts may need to re-calibrate the parameters in the processing tool because those parameters may become obsolete if the observation data varies due to changing conditions during the course of the measurements. That is why our main focus is to optimise this step in this study.

- After a long analysis and annotation period on those measurement images, the investigation team delivers the results (such as laminar-turbulent boundaries, or coordinate points of those regions corresponding to the reference markers as explained in Sect. 3.2) to the design team back.
- Finally, the design team considers those processed images and investigation results to decide if the original design is satisfactory or if it needs some revisions.

In the light of the workflow above, Sect. 1.1 explains the laminar-turbulent flow phenomenon for those who has no fluid dynamics background, Sect. 1.2 details the issues in the automation of laminar-turbulent flow localisation, and Sect. 1.3 lists the contributions of our study to overcome the issues mentioned in Sect. 1.2.

1.1 What is laminar and turbulent flow?

Laminar-Turbulent flow is a fluid dynamics phenomenon that refers to the motion of particles as they move through a substance. The distinction between laminar and turbulent flow is eminently important in aerodynamics and hydrodynamics because the type of flow has a profound impact on how momentum and heat are transferred. In principle, when particles of fluid have a property of a streamlined flow, and they follow a smooth path without interfering with one another, it is called *laminar flow*. On the other hand, *turbulent flow* means the chaotic and rough movement of particles through a substance (Emmons 1951), such as mixing and shifting between the flow layers and whirlpool-like patterns as illustrated in Fig. 2.

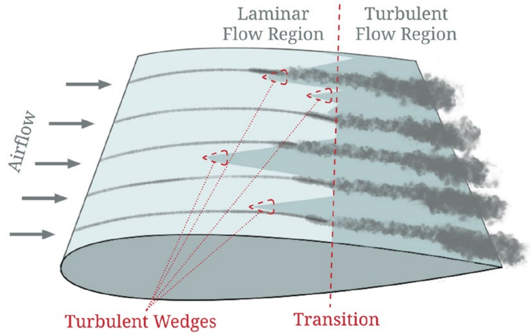

Fig. 2 Airflow over a wing exhibiting transition from laminar to turbulent

Specifically, detection of laminar and turbulent flow regions and transition locations in between are of crucial interest in a range of aviational applications, since the achievement of a fuel-efficient wing, airfoil or rotor blade design and reducing the instability caused by drag forces are strongly related with the way of handling laminar and turbulent flow exposed by the wing and blade surfaces. To illustrate, friction resistance accounts for about half of the total drag exposing to an aircraft in cruise (Schrauf 2005), and extending the laminar flow regions decrease frictional resistance, so it has the potential to significantly reduce fuel combustion. That is why sophisticated experimental techniques to determine the surface friction distribution have been applied in the industry and academia for the purpose of validating the consistency among the measurements, performance assessments and design methods (Bégou et al. 2017).

Observation techniques for determining the laminar-turbulent flow regions in aviational applications fall into two broad categories, which are (i) intrusive methods (e.g. hot films, temperature-sensitive paint, oil-flow), and (ii) non-intrusive methods (Joseph et al. 2016). In this study, we have applied to one of the non-intrusive methods, which is *infrared thermography*, a known technique for visualising different flow states by utilising temperature differences on an object surface. Basically, heat transfer between an object surface and an external flow is proportional to the friction that occurs at the surface, which is exposed to this flow (Quast 2006; De Luca et al. 1990) . In other words, the surface exposed to a laminar boundary layer will develop a different temperature than the surface exposed to a turbulent boundary layer. Thus, the temperature difference can be observed through a thermographic camera in case of applying the method of infrared (IR) thermography.

1.2 What are the main issues in flow localisation?

Once the temperature distributions caused by laminar and turbulent flows are measured, those data should be interpreted in a various manner for determining the flow regions and transition locations, which is traditionally handled by human experts. However, such a visual interpretation is not always effective due to (i) the induced bias by the experts with different experience levels, and (ii) the time-consuming nature of manual observation and interpretation (Crawford et al. 2015). Even, at some point, noise and low contrast will make analysis difficult for a human.

Although there have been some recent efforts, as detailed in Sect. 2.1, to automatise the separation or localisation of flow states, they often lack reproducibility due to their dependence on clean, high-contrast and homogeneous IR data. For instance, IR images may look different depending on surface material (metal vs. composite), surface heating (none vs. internal vs. external), observed object (fixed-wing vs. rotor), environment (wind tunnel vs. flight test), camera sensor, view angle and the distance between camera and object to mention just a few influences. For this reason, a classical processing technique - such as differential thermography or thresholding - may work for one setup, but may fail at a different task. Hence, a more general approach is necessary to automatically localise the flow states in images from a wide range of test setups.

1.3 Our contributions in flow localisation

The contributions of the study are:

- We introduce the challenging problem of separating the laminar flow regions from the remaining parts of a thermographic measurement image which has a low signal-to-noise ratio, and a low contrast value. It is worth noting that there is only a single image from a specific test point in most cases in our experiments. Nevertheless, in many of the former efforts, multiple images from a single test point were mostly desired for proper separation of flows.
- For the first time in the literature, we propose an automatic laminar flow segmentation system based on a variant of encoder-decoder-based deep neural networks, namely *Adaptive Attention Butterfly Network*, providing reproducibility in different flow measurement scenarios. It is also worth adding that our proposed network is a lightweight but better-performing alternative to some well-known segmentation networks as demonstrated in benchmark comparison in Sect. 6.2.
- We show that the robustness of the laminar flow segmentation can be significantly improved with the self-supervised learning strategies in the presence of a limited amount of ground-truth data due to the difficulty of manual labelling of thermographic measurement images.

With those contributions, the presented approach aims to interact as a bridge between the raw image and the post-processing. If it would be possible to extract information from the original input image about the location of the laminar flow area within the image, the non-laminar flow regions can be masked out, because the main interest in the image analysis is detecting where the laminar flow states are.

The rest of this paper is organised as follows: after a review of the state of the art in Sect. 2, the setup for collecting the thermographic laminar-turbulent flow measurements are explained in Sect. 3. Sect. 4 outlines the proposed method for automatic flow segmentation. The experimental framework and the discussion about the obtained results are given in Sects. 5 and 6. Eventually, the conclusions and future works are drawn in Sect. 7.

2 Related work

2.1 Automatic localisation of laminar-turbulent flow

It is worth pointing out that the segmentation of laminar-turbulent flow regions has been mainly handled by human experts so far, who observed the thermographic images and defined the flow transition locations by manually (or semi-automatically) drawing the boundary lines, which is a very inefficient and time-consuming task. For this reason, some of the efforts in the literature of laminar-turbulent flow measurement have focused on processing the IR thermography data in an automated way.

In one of the pioneering attempts, Gartenberg and Wright (1994) have proposed an image subtraction technique to illustrate the potential of image processing approaches for automatic detection of flow regions. By following a similar principle, Raffel and Merz (2014) have proposed a technique that provides high-contrast images using differential IR thermography between two successive time steps, which are suitable for automated processing. Later, Grawunder et al. (2016) have extended differential IR thermography approach for low-enthalpy flows where the temperature difference between laminar and turbulent regions are low, while Simon et al. (2016) and Wolf et al. (2019) have made such an extension for unsteady flow measurements. The major drawback in these attempts is the need for multiple IR thermography measurements to detect the flow region boundaries.

On the other hand, Richter and Schülein (2014) have put forth an automatic flow localisation method based on a single instantaneous thermal image using the chordwise temperature and infrared signal intensity distribution. Later, Crawford et al. (2015) have exploited classical image processing techniques like median and Gaussian filtering, thresholding, contour finding and extraction of quadrilaterals via the Ramier-Douglas-Peuker algorithm. Nevertheless, image inhomogeneities have been partially corrected or ignored in the aforementioned efforts, limiting the distinguishability of flow states.

In some of the recent studies, Dollinger et al. (2018) have applied discrete Fourier transform to evaluate the mean amplitude of the temporal fluctuations which is less affected by spatial inhomogeneities within the flow states. Similarly, Gleichauf et al. (2020) have proposed a flow transition localisation system that increased the distinguishability between laminar and turbulent flow states employing non-negative matrix factorisation. Nonetheless, the main drawbacks of these approaches are the need for a priori knowledge such as frequency range of temperature fluctuations in the former study, or approximate location of the flow states in the latter one, in order to determine the optimal boundaries. Another recent effort from Gleichauf et al. (2021) exploited principal component analysis to reduce the image artefacts and temperature gradients within the flow states, achieving lower measurement error in the laminar-turbulent transition localisation.

A common thing among the aforementioned approaches, which have tried to develop automatic localisation of flow transition regions, is the dependence upon clear IR images. They usually require:

- High signal-to-noise ratio (SNR), which depends upon the temperature resolution of camera and the temperature difference between laminar and turbulent regions;
- Low thermal response time, which is determined by the heat capacity and conductivity of the surface material;
- High spatial resolution, and low smearing in IR data.

Nonetheless, those requirements are difficult to meet in most of the measurement cases, as illustrated in Fig. 3, due to the possible existence of noise, low contrast, artefacts, reflections as well as structure elements below the observed surface. For this reason, more advanced methods are needed to cope with laminar-turbulent flow localisation under such noisy and heterogeneous measurement cases. Thence, in this study, we examined the feasibility of deep learning-based segmentation techniques to robustly localise the flow regions observed in the IR thermographic measurements on wings and stabilisers.

2.2 Image segmentation via deep learning

Although traditional laminar-turbulent flow localisation and segmentation techniques have mostly relied on classical image processing and transformation algorithms, the impact of the recent deep learning-based approaches has been never investigated in this domain to the best of our knowledge. For this reason, in this section, we briefly mention some of the recently emerging deep learning-based segmentation approaches which have yielded remarkable performance improvement in various computer vision problems ranging from autonomous driving, video surveillance to medical diagnosis, as well as we recall the fundamental principles of the neural networks resorted for image segmentation.

2.2.1 Fundamentals of encoder-decoder networks

For segmentation tasks, the most widespread deep learning-based approaches originated from encoder-decoder networks. Let $\{\mathbf{x}, \mathbf{y}\}$ be the data-label pair, and $E(.)$ and $D(.)$ be the encoder and decoder functions, respectively. Thus, the encoder is responsible for producing hidden code $\mathbf{h} = E(\mathbf{x})$, and the decoder for constructing the output $\hat{\mathbf{y}} = D(\mathbf{h})$. Here, minimisation of $\|D(E(\mathbf{x})) - \mathbf{y}\|$ is the main learning objective of the encoder-decoder network to achieve $\hat{\mathbf{y}} \approxeq \mathbf{y}$.

2.2.2 U-Net: a special type of encoder-decoder networks

In principle, U-Net is an encoder-decoder-based network, but its distinction stems from the skip connections between the symmetrically arranged network layers of encoder and decoder (Ronneberger et al. 2015). This novel arrangement efficiently extracts and assembles multi-scale feature maps in which encoded features propagate to decoder blocks via skip connections and a bottleneck layer, as illustrated in Fig. 4. To elaborate on the standard U-Net architecture:

- *Encoding Phase* consists of a series of operations involving 3×3 convolution followed by a ReLU activation function as explained in Fig. 4b. The obtained feature maps are downsampled with the max pooling operation, as depicted in Fig. 4c. Throughout the layer blocks, the number of feature channels are increased by a factor of 2; whereas convolution, activation and max pooling lead to spatial contraction of the feature maps until the bottleneck layer.
- *Decoding Phase* consists of sequences of up-convolutions which map each feature vector to the 2×2 pixel output window, as explained in Fig. 4d. Later, they are concatenated with high-resolution features coming from the corresponding encoded layers with skip connections. Contrary to the operations in encoder, throughout the layer blocks, the number of feature channels are

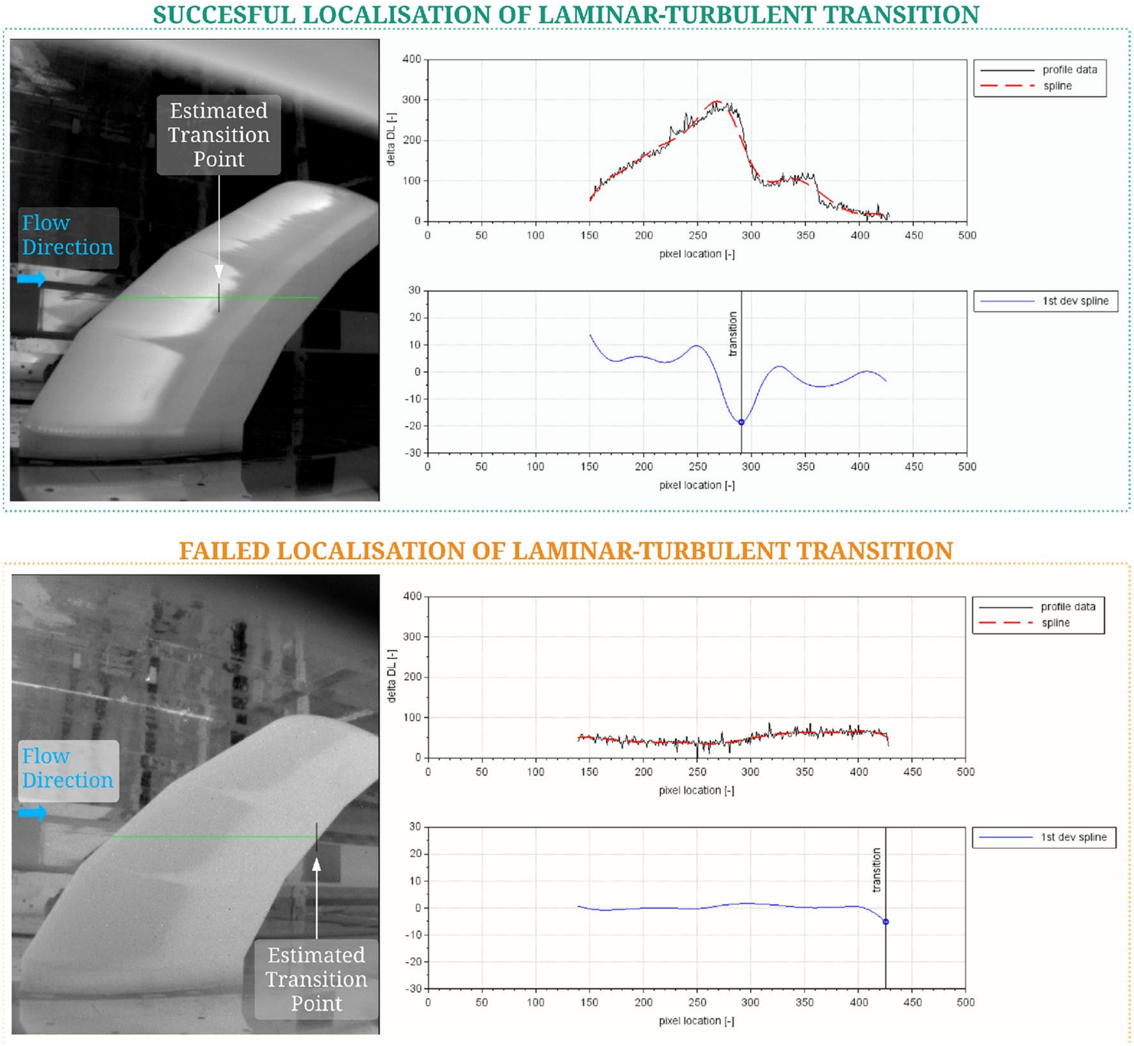

Fig. 3 Success and failure examples for localisation of laminar-turbulent flow transition point on the flow direction. Note that a localisation approach based on extrema of profile gradients works well with high-contrast/low-noise IR imaging (above) but might fail with low-contrast/high-noise IR imaging (below). The qualitative examples were composed with the data taken from the project described in Kruse et al. (2018)

decreased by a factor of 2; whereas up-convolution, convolution and ReLU activation operations result in spatial expansion of the feature maps until the output layer. Finally, the output layer generates a mask comprising segmented background and foreground.

2.2.3 U-Net variants for image segmentation

Even though U-Net has been proposed firstly for medical image segmentation (Ronneberger et al. 2015), it has become a widespread solution in many segmentation tasks due to its data augmentation capabilities for effective learning in case of having a limited amount of annotated images. Nevertheless, its main drawbacks are a large number of parameters and performance losses when having different input shapes. To mitigate those issues, many variants of U-Net have emerged in recent years, which can broadly fall into the five categories (Punn and Agarwal 2022): (i) *Improved U-Nets*, (ii) *Inception U-Nets*, (iii) *Attention U-Nets*, (iv) *Transformer U-Nets*, and (v) *Ensemble U-Nets*, depending on the architectural differences:

- *Improved U-Nets*: are composed of U-Net models performing better than the standard U-Net through slight modifications, such as adding dense or residual blocks, exploiting transfer learning or multi-stage training. Among these, it is worth mentioning Tong et al. (2018) with their improved U-Net which includes mini-residual connections within encoder-decoder phases, Alom et al. (2019) with their R2 U-Net which is based on a recurrent

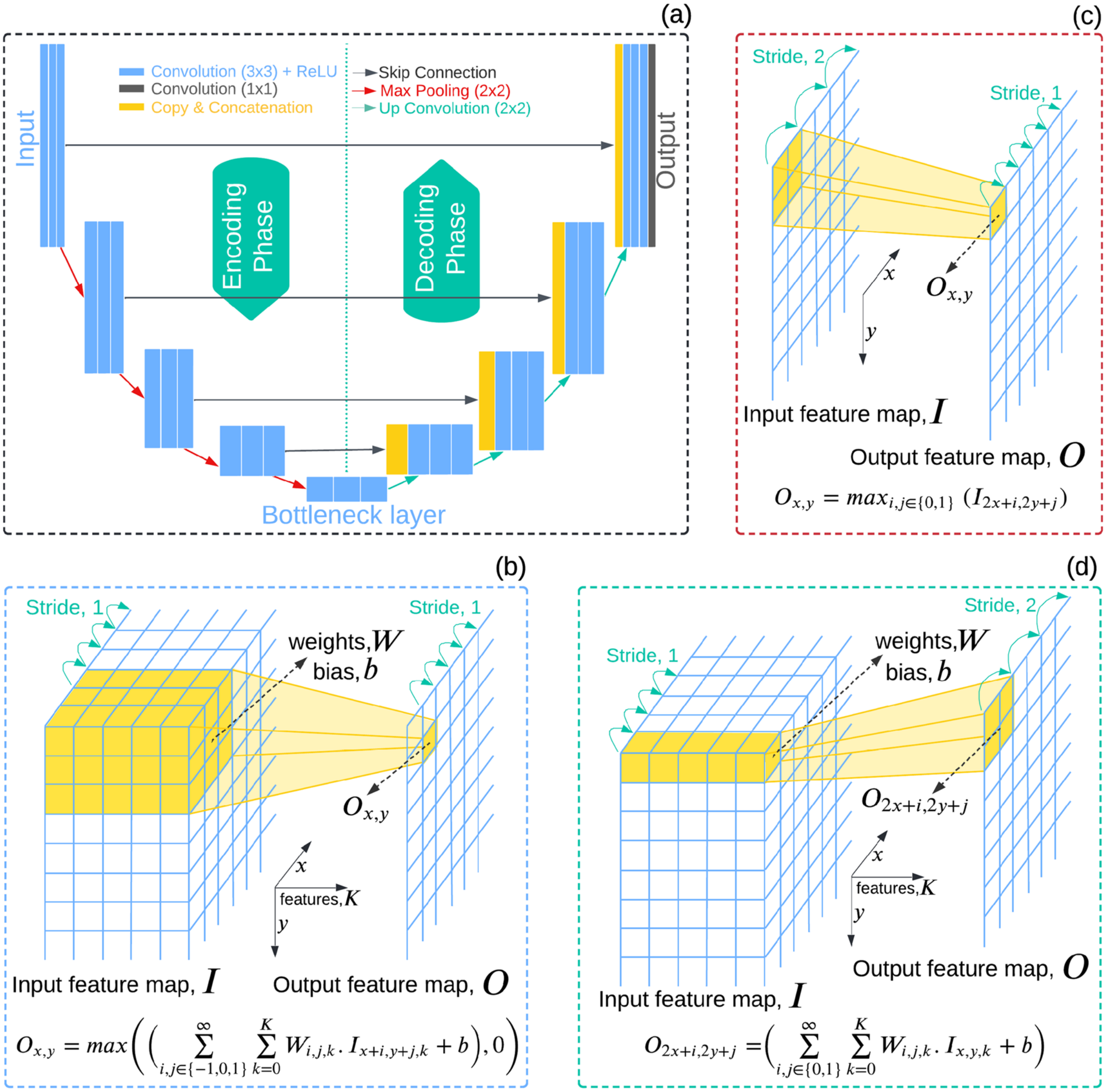

Fig. 4 Standard U-Net architecture and summary of operations in it: **a** U-Net high-level representation, **b** 3×3 convolution and ReLU operation, **c** 2×2 max pooling operation, **d** 2×2 up-convolution operation. Note that, I and O stands for input and output feature maps, respectively, of any layer in U-Net

residual convolutional neural network, and Huang et al. (2020) with their U-Net 3+ which is based on full-scale skip connections and deep supervisions in order to incorporate low-level details with high-level semantics from feature maps in different scales.

- *Inception U-Nets*: use multi-scale feature fusion strategies to learn the feature representations effectively. For instance, Zhang et al. (2021) have built dual encoder models with a densely connected recurrent convolutional neural network, and their DEF-U-Net performed better in extracting the spatial features. Recently, Xia et al. (2022) have put forth MC-Net which exploits the residual attention approach in multi-scale context extraction to model the global and local semantic information in the regions to be segmented.
- *Attention U-Nets*: make use of attention mechanisms together with convolutional layers in feature mapping to filter the most relevant features, such as spatial attention, channel attention or mixed attention. For instance, Schlemper et al. (2019) have introduced Attention U-Net for medical imaging that automatically learns to focus on target structures of varying shapes and sizes, and Ren et al. (2020) have combined dual attention mechanism (Fu et al. 2019) with capsule networks (Hinton et al. 2018) for precise extraction of road network from remote sensing images.

- *Transformer U-Nets:* replaces classical convolutional layers with transformer-based blocks which are known better for capturing a long-distance pixel relation due to the self-attention mechanisms. For instance, Swin U-Net (Cao et al. 2021) is composed of pure hierarchical vision transformer blocks with shifted windows inspired by Swin Transformer (Liu et al. 2021). Similarly, Wang et al. (2022) have introduced Mixed-Transformer U-Net (MT-U-Net) which refined the self-attention mechanism by simultaneously obtaining intra- and inter-correlations. Last but not least, Trans U-Net (Chen et al. 2021) has combined convolutional blocks with transformers in the encoder part.
- *Ensemble U-Nets*: exploits multiple models and submodels with or without the other enhancements. For instance, to address the issue of dealing with different input shapes, Isensee et al. (2018) introduced nnU-Net which is a robust and self-adapting framework based on 2D and 3D vanilla U-Nets. On the other hand, Qin et al. (2020) have proposed U2-Net, which is a nested version of the original U-Net with residual U-blocks. Recently, Jha et al. (2020) have proposed a sequence of two U-Nets, namely DoubleU-Net, with Atrous Spatial Pyramid Pooling (Chen et al. 2017) in between encoder-decoder parts, for eliminating the artefacts observed in medical image segmentation.

2.2.4 Other approaches for image segmentation

Among the other most popular deep learning-based segmentation models resorting to encoder-decoder architecture: Noh et al. (2015) introduced DeConvNet, and Badrinarayanan et al. (2017) proposed SegNet, which are semantic segmentation models based on transposed convolution. The encoder parts of both architectures were adopted from the VGG-16 network (Simonyan and Zisserman 2014), but the novelty of the latter one was to perform nonlinear upsampling in decoder layers by using pooling indices in the corresponding encoder layers.

Multiscale and pyramid-based architectures are also common in image segmentation. Some of the most prominent models are Feature Pyramid Network introduced by Lin et al. (2017) for object detection but later applied to segmentation, and the Pyramid Scene Parsing Network proposed by Zhao et al. (2017) to better learn the global context representation of a scene. After that, atrous convolution, which brought the dilation rate to the convolution operation, and its combination with pyramid architectures have become very popular due to their ability in addressing the decreasing resolution issue, and in robustly segmenting objects at multiple scales. For instance, the core component of the family of DeepLab architectures introduced by Chen et al. (2017) is Atrous Spatial Pyramid Pooling (ASPP).

Semi-supervised and self-supervised learning methodologies have been also exploited for image segmentation. Souly et al. (2017) and Hung et al. (2019) applied to semi-supervised learning by using generative adversarial networks (GANs). Goel et al. (2018) proposed a deep reinforcement learning approach for moving object segmentation in videos. Wang et al. (2020) put forth self-supervised equivariant attention mechanism for semantic segmentation.

Other very prominent image segmentation architectures are the variants of region-based convolutional neural networks (R-CNNs) (Ren et al. 2016; He et al. 2017), CNNs with active contour models (Chen et al. 2019; Gur et al. 2019), and panoptic segmentation variants (Kirillov et al. 2019; Li et al. 2019).

3 Thermographic measurement setup

In this part, Sect. 3.1 summarises widespread use cases in aerodynamics requiring thermographic measurement, Sect. 3.2 elaborates on a particular measurement setup conducted in our lab, as one of the data sources for our flow localisation architecture proposed in this study, Sect. 3.3 lays out the common features of the diverse measurement setups for the audience with limited aerodynamics background.

3.1 Common use cases for thermographic measurement

IR thermography in aerodynamics is nowadays commonly used in a vast variety of measurement setups. Applications can be found at wind turbines (rotor blades), wind tunnel experiments (huge variety of measurement setups) as well as on ground and flight tests on helicopters (rotating blades) and fixed wing aircrafts (e.g. wings, flaps, stabilisers). Thus, the experimental setup for a thermographic measurement will almost always differ from one to another.

To have a general insight into the common use cases for the application of IR thermography measurements, the reader might have a look at the listed references: (Quast 1987; Dollinger et al. 1992; Raffel and Merz 2014; Traphan et al. 2015; Bakunowicz and Szewczyk 2015; Joseph et al. 2016; Koch et al. 2020; Gardner et al. 2020; Schrauf and von Geyr 2021).

For wind turbines, an IR camera might be installed on the ground or directly at the wind turbine. The usual region of interest (ROI) in these cases is the observation of a wind turbine rotor part.

Wind tunnel projects also may differ in the number of featured IR cameras. It is quite common to have one or two IR cameras installed; however, one of the authors of this manuscript has participated in a wind tunnel test featuring five IR cameras of different types. As the IR images

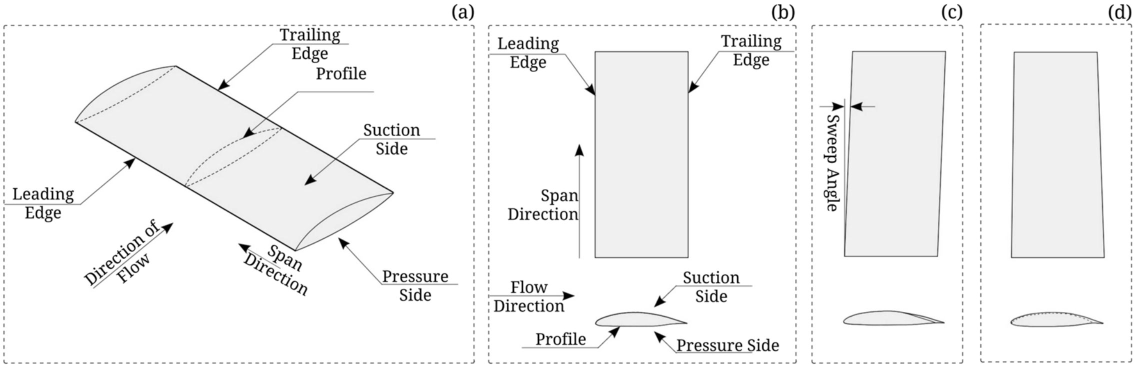

Fig. 5 **a** Schematic iso-view on a generic 2D-wing profile. Top and side view on different wing model types: **b** 2D-model, **c** 2.5D-model, and **d** 3D-model, respectively

presented in this work are linked to aerodynamic research, the ROI in wind tunnel project is often the suction and/or the pressure side of the wind tunnel model. The model itself might be (Fig. 5):

- a 2D-wing model, featuring only a profile of constant width, which is stretched perpendicular to the profile plane to generate a span width;
- a 2.5D-wing model, which is a 2D-model, but with a sweep angle;
- a 3D-wing model, featuring sweep angle, wing taper, twist and bend;
- a 3D-wing model including moveable devices like slats and flaps;
- an aircraft half-span model (e.g. half of fuselage + complete wing);
- a full span model (e.g. complete aircraft model).

To mention some other examples, flight tests on sail planes (gliders) (Seitz 2007; Barth 2021), have been conducted using only one IR camera. On the other hand, flight tests on small and large motorised aircrafts of very different kinds have featured at least one or two IR cameras. Besides, laminar-turbulent transition detection is used in industry and research facilities worldwide for the sub-, trans- and supersonic flight tests (Frederick et al. 2015).

3.2 Specifications of thermographicmeasurement setup for a use case

The examples in the previous section put forth the diversity of measurement setups featuring IR thermography. Even though the existence of such widespread applications, most of the measurement results may not be published due to binding confidentiality agreements. This is especially the case for the research collaborations including partners from the industry. Nevertheless, the authors would like to portray a single example from a recent flight test setup conducted in their institute, under the scope of the AFLoNext Project (AFLoNext 2018a), which made its test flights in spring 2018 (AFLoNext 2018b). The description of the test setup is given below:

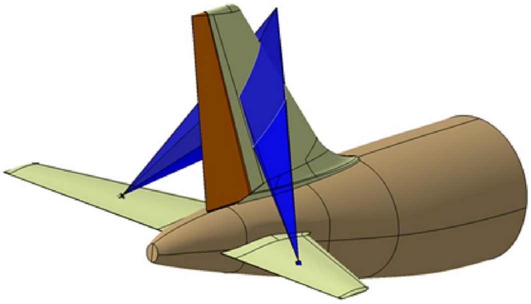

Fig. 6 AFLoNext, digital mockup field of view (FoV) study using 20° lens

Beside other flight test instrumentation, two IR cameras had been installed into the horizontal tail plane (HTP) of the test aircraft, which was an AIRBUS A320. One IR camera was installed at the port (left hand) and one at the starboard (right hand) side of the aircraft. Both cameras were orientated in such a way that they observed a ROI at the vertical tail plane (VTP). As a consequence, IR images could be taken during flight simultaneously both from the left- and right-hand side of the VTP, as illustrated in Fig. 6.

The cameras were of type FLIR SC3000, which featured a 320×240 pixel QWIP sensor. This camera type is specified to detect temperature differences of 20 mK at $30°C$ and operates in the spectral range of $8 - 9$ μm. Both cameras were equipped with a 20° lens and connected to a data acquisition (DAQ) system, which was used to operate the cameras and display and record the IR images. A set of IR images of

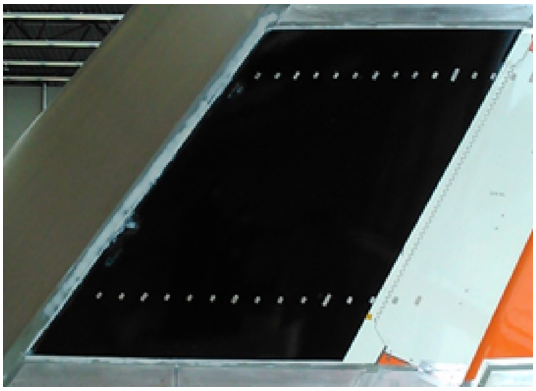

Fig. 7 Black foil within the AFLoNext IR-FOV port side, showing the reference markers. Flow direction is from left to right

this flight test campaign has been exploited within this work together with the images from other test campaigns as seen in Fig. 11, to train our proposed neural network for laminar-turbulent flow separation.

Since only sun illumination could be used to increase image contrast between laminar and turbulent flow, a thin black foil was applied onto the surface to increase image quality as observed in Fig. 7. The foil will not act so much as an insulation to the underlying structure - which would prevent the appearance of the structure in the IR image - but more as a heating device. The black foil absorbs the energy of the solar radiation, resulting in a surface temperature increase. Thus, the surface temperature difference between the airflow and the surface will increase, resulting in a higher image contrast on the VTP side that is faced to the sun.

Reference markers have been also applied to measure the laminar-turbulent transition location as also seen in Fig. 7.

3.3 Generalisation of measurement setup

As described in Sects. 3.1 and 3.2, measurement setups may differ from use case to use case. However, at least three things are shared between all setups: (i) a surface, (ii) an airflow to which the surface is exposed to, and (iii) at minimum, one IR camera observing the surface.

Parameters like model orientation (horizontal vs. vertical), camera view angle, object-camera-distance, field of view, surface heating, surface coating, surface material, thermographic system (camera type) and many more are in their combination most likely unique for every test setup.

As illustrated in Fig. 8, a simplified laminar-turbulent flow measurement setup applied in the variety of our experiments can be described as follows: A surface of an object is observed by an IR camera. The camera together with its optical system (lenses) spans a field of view (FoV) onto the object surface. Everything on this FoV is visible within the recorded image. As camera and object have their own reference system, coordinate systems for the object and camera exist. Modern IR cameras work with a focal plane array (FPA) being the sensor, hence the image plane is put into the sensor plane. Perpendicularly from the camera sensor plane, a view axis can be defined, which is the optical axis. The intersection of the optical axis with the surface is the focal point of the camera. Reference markers located in the FoV allow the image transformation from image to object coordinates. Since the work presented here is based on images in the aerodynamic field, the object of interest is exposed to a flow with a dedicated flow direction. Eventually, a laminar and turbulent region may develop inside the FoV if the object (flight-wings and vertical stabiliser in our experiments) is exposed to a flow.

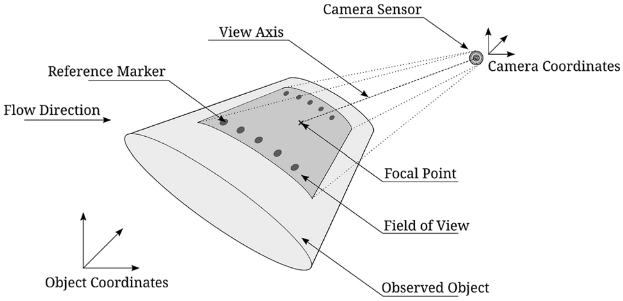

Fig. 8 A simplified laminar-turbulent flow measurement setup

In the scope of this paper, the laminar-turbulent-transition type (e.g. Tollmien-Schlichting or Cross-Flow-Transition) is of no interest. Other flow phenomena, such as the presence of laminar separation bubbles, play a minor role. Considering these effects will require different training data and may change the neural network weights, but won't change the essence of the automatic flow localisation/segmentation approach proposed in this paper.

4 Proposed flow segmentation methodology

As already mentioned in Sect. 1, three major problems in the thermographic measurement of laminar-turbulent regions led us to the proposed approach in this study. Those are (i) noisy observations with some artefacts, reflections or low contrast among the flow regions which makes the automatic localisation of flow regions difficult, (ii) limited amount of labelled data preventing to build an efficient segmentation system based on neural networks, and (iii) the computational complexity of our early deep learning-based segmentation attempts for our regular workflow depicted in Fig. 1.

To deal with the first and the last problems, we proposed Adaptive Attention Butterfly Network (shortly *ButterflyNet*) for the effective separation of laminar flow

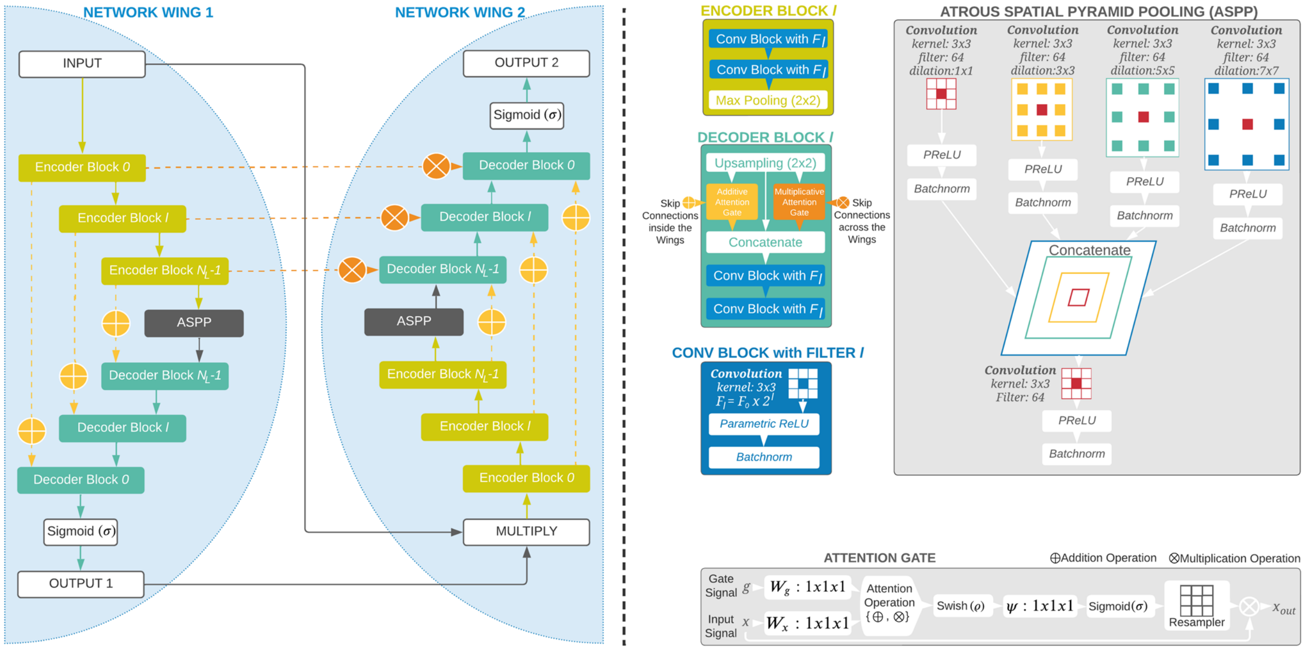

Fig. 9 The Adaptive Attention Butterfly Network (ButterflyNet) architecture (left), and the details of the blocks utilised in the ButterflyNet (right)

from the other flow regions, as detailed in Sect. 4.1. The proposed ButterflyNet falls into the category of *Ensemble U-Nets* already described in Sect. 2.2.3, with the inspiration from the following studies: (i) Attention U-Net variants (Schlemper et al. 2019; Fu et al. 2019; Ren et al. 2020) which increase the model sensitivity and prediction accuracy with minimal computational overhead, (ii) DoubleU-Net (Jha et al. 2020) which is useful to eliminate the small artefacts in the segmented areas, and (iii) nnU-Net (Isensee et al. 2018) which enables dynamic adaptation of network topologies to different image modalities and geometries, and compensates the issue of having a large number of network parameters and of inefficient throughput time in case of using cascaded networks as similar to our proposed ButterflyNet.

Nevertheless, our second problem, having few amounts of labelled images, might reduce the robustness of the segmentation due to the model overfitting or the lack of generalisation ability. That is why we adopted a self-supervised learning strategy, SimCLR (Chen et al. 2020), to exploit non-labelled thermographic measurements along with the labelled ones for more reliable flow segmentation, as detailed in Sect. 4.2.

4.1 Adaptive attention butterfly network

The ButterflyNet is composed of two cascaded networks ($WING_1$ and $WING_2$) as shown on the left side of Fig. 9. Input image $\mathbf{i}_I \in \mathbb{R}^{w_I \times h_I \times c_i}$ is fed into the $WING_1$ of the network to get the segmented output image $\mathbf{i}_{o_1} \in \mathbb{R}^{w_I \times h_I \times c_o}$ where w_I and h_I are the image width and height, and c_i and c_o are the number of channels at the input and output images, respectively. Later, element-wise multiplication ($\odot$) of input $\mathbf{i}_I$ and output $\mathbf{i}_{o_1}$ is fed into the $WING_2$ of the network to get the final output, $\mathbf{i}_{o_2} \in \mathbb{R}^{w_I \times h_I \times c_o}$, such that:

$$\begin{aligned} \mathbf{i}_{o_1} &= WING_1(\mathbf{i}_I) \\ \mathbf{i}_{o_2} &= WING_2(\mathbf{i}_I \odot \mathbf{i}_{o_1}) \end{aligned} \tag{1}$$

To elaborate on the wings, each of them contains N_L number of encoder blocks (E_l) and decoder blocks (D_l) with $0 \leq l < N_L$, where l stands for the block order. Each E_l and D_l blocks are linked with the skip connections, and an *ASPP* (Chen et al. 2017) is placed between the blocks E_{N_L-1} and D_{N_L-1}. When a skip connection takes place inside a wing, an additive attention gate (AG_a) filters the features propagated through the connection. However, if a skip connection is between $WING_1$ and $WING_2$, a multiplicative attention gate (AG_m) filters the features.

Each encoder block, E_l, is composed of two consecutive convolutional blocks ($CONV_l$) and one 2×2 max pooling operation ($POOL_{2x2}$) for downsampling. Thus, each encoder block E_l can map a feature vector $\mathbf{x}_e^l$ to the next block with the following series of functions:

$$E_l \overset{\text{def}}{=} POOL_{2x2} \circ CONV_l \circ CONV_l : \mathbf{x}_e^l \mapsto \mathbf{x}_e^{l+1} \tag{2}$$

Conversely, each decoder block, D_l, starts with one 2×2 upsampling operation ($UP_{2\times2}$). After that, the outputs of $UP_{2\times2}$ and the attention gates, AG_a and AG_m, are concatenated and later fed into two consecutive $CONV_l$ blocks. Thus, a decoder block D_l can be formulated as:

$$D_l \overset{\text{def}}{=} CONV_l \circ CONV_l \circ CONCAT : \mathbf{x}_d^{l+1} \mapsto \mathbf{x}_d^l, \quad \text{where:}$$
$$CONCAT \overset{\text{def}}{=} \begin{cases} [\, UP_{2\times2}, (AG_a \circ UP_{2\times2}) \,] & \text{if on } WING_1 \\ [\, UP_{2\times2}, (AG_a \circ UP_{2\times2}), (AG_m \circ UP_{2\times2}) \,] & \text{if on } WING_2 \end{cases} \tag{3}$$

As illustrated in Fig. 9, each $CONV_l$ block contains:

- Convolution function, $f_{conv}(\mathbf{x};\boldsymbol{\phi})$, where $\boldsymbol{\phi}$ is 3×3 kernel parameter with filter size $F_l = F_0 \times 2^l$ where $F_0 \in \mathbb{Z}^+$,
- Batch normalisation (BN) layer,
- And parametric rectified linear unit ($PReLU$).

Thus, $CONV_l$ block is formulated as:

$$CONV_l \overset{\text{def}}{=} BN \circ PReLU \circ f_{conv} : \mathbf{x}^l \mapsto \hat{\mathbf{x}}^l, \tag{4}$$

Furthermore, attention gate (AG) is a mechanism that identifies salient image regions and prune relevant feature responses to a specific task by scaling feature vector $\mathbf{x}$ with attention coefficient $\boldsymbol{\alpha}$. Thus, the output of an AG is $\hat{\mathbf{x}} = \boldsymbol{\alpha}^\top \mathbf{x}$. In our proposed ButterflyNet, attention coefficient $\boldsymbol{\alpha}$ is computed as follows by adapting the approach in (Schlemper et al. 2019):

$$\alpha = \sigma\big(AG(\mathbf{x}, \mathbf{g}; \boldsymbol{\phi}_{att})\big), \quad \text{where}$$
$$AG = \begin{cases} \boldsymbol{\psi}^\top \big(\varrho\big((\mathbf{W}_x^\top \mathbf{x} + b_x) \odot (\mathbf{W}_g^\top \mathbf{g} + b_g)\big)\big) + b_\psi & \text{if multiplicative} \\ \boldsymbol{\psi}^\top \big(\varrho\big((\mathbf{W}_x^\top \mathbf{x} + b_x) + (\mathbf{W}_g^\top \mathbf{g} + b_g)\big)\big) + b_\psi & \text{if additive} \end{cases} \tag{5}$$

where AG is characterised by set of parameters $\boldsymbol{\phi}_{att}$ consisting of linear transformations $\mathbf{W}_x$, $\mathbf{W}_g$, $\boldsymbol{\psi}$ and bias terms $\boldsymbol{b}_x$, $\boldsymbol{b}_g$, $\boldsymbol{b}_\psi$. Moreover, $\sigma(\mathbf{x}) = {}^{1}\!/_{1+e^{-x}}$ and $\varrho(\mathbf{x}) = {}^{x}\!/_{1+e^{-x}}$ are the *sigmoid* and *swish* nonlinear activation functions, respectively. Note that, in AG, the features coming from skip connections are considered as input signal $\mathbf{x}$, whereas the features mapped from the previous layer are considered as gate signal $\mathbf{g}$.

Besides, our proposed network is subject to the mean-absolute error function to be minimised:

$$\mathcal{L}_{MAE} = \sum_i \sum_j |\hat{p}_{ij} - p_{ij}|, \tag{6}$$

where p_{ij} and $\hat{p}_{ij}$ are the ground-truth pixel label and the predicted pixel label belonging to a segmentation output, respectively.

4.2 Self-supervised learning framework

Self-supervised learning methodologies are aiming at designing pretext tasks (*e.g.* relative position prediction, image colourisation, or spatial transformation prediction) to generate labels instead of using a large amount of annotated labels to train a network (Wang et al. 2020). Similarly, *SimCLR*, which learns visual representations by maximising agreement between differently augmented views of the same data via a contrastive loss, is counted as one of the simplest and most powerful techniques for self-supervision (Chen et al. 2020). That is why we rely on the SimCLR architecture with some task-based modifications in our study.

As illustrated in Fig. 10, our self-supervised learning framework is composed of the following components:

- A stochastic *data augmentation* module, $\mathcal{T}$, to randomly transform a data into correlated views of it,
- The ButterflyNet without the decoder part at the second wing, as a *base encoder* module of the framework, $f(.)$,
- A *projection head* module, $h(.)$, composed of multilayer perceptron (MLP) with two hidden layers, followed by ℓ_2 normalisation to extract feature representations,
- A *binary classifier* with a fully-connected layer, $g(.)$.

To elaborate on, let $\mathcal{B} = \{\mathbf{i}_1, .., \mathbf{i}_j, .., \mathbf{i}_{N_\mathcal{B}}\}$ be the batch of $N_\mathcal{B}$ random image samples, and let two augmentation operators, $t' \sim \mathcal{T}$ and $t'' \sim \mathcal{T}$, be sampled from the data augmentation module to obtain two different views of images in $\mathcal{B}$, such that $t' : \mathbf{i}_j \mapsto \mathbf{i}'_j$, $t'' : \mathbf{i}_j \mapsto \mathbf{i}''_j$; therefore, the feature representations ($\mathbf{x}_j$'s) for those images are obtained as:

$$\ell_2 \circ h \circ f : \mathbf{i}'_j \mapsto \mathbf{x}'_j, \mathbf{i}''_j \mapsto \mathbf{x}''_j, \text{ where:}$$
$$\ell_2(\mathbf{x}) = \frac{\mathbf{x}}{\|\mathbf{x}\|_2} \quad \text{and} \quad h(\mathbf{x}) = \varrho\big(\mathbf{W}_{(2)}\big(\varrho(\mathbf{W}_{(1)}^\top \mathbf{x})\big)\big) \tag{7}$$

where $\mathbf{W}_{(1)}$ and $\mathbf{W}_{(2)}$ are the linear transformations in MLP and ϱ is the *swish* nonlinear activation. Moreover, binary decision about if two separate feature representations stems from the same input is obtained as follows:

$$g(\mathbf{z}) = \sigma(\mathbf{W}_c^\top \mathbf{z}), \quad \mathbf{z} = \mathbf{x}'_i - \mathbf{x}''_j, \quad y(\mathbf{z}) = \begin{cases} 0 & \text{if } i = j \\ 1 & \text{if } i \neq j \end{cases} \tag{8}$$

where $\mathbf{W}_c$ is the linear transformation in the binary classifier, σ is the *sigmoid* nonlinear activation, and y is the pretext label for output $\mathbf{z}$.

Therefore, the self-supervised learning framework is subject to the following loss function to be minimised:

$$\mathcal{L}_{total} = \beta \cdot \mathcal{L}_{CL} + \omega \cdot \mathcal{L}_{BCE}, \tag{9}$$

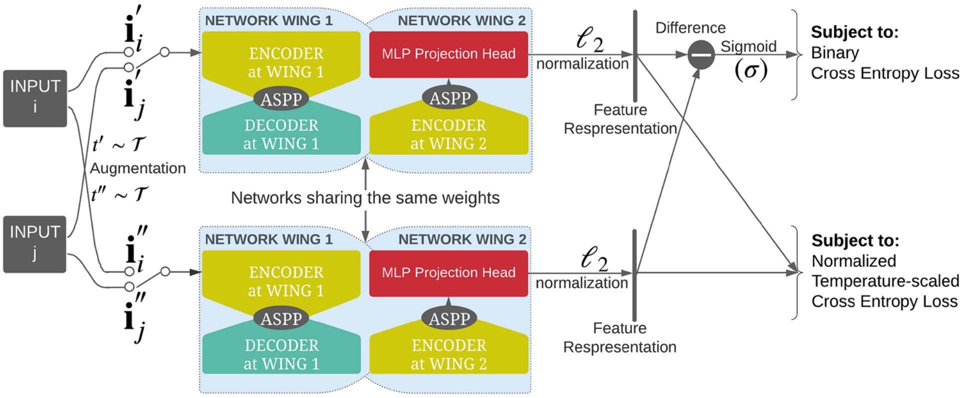

Fig. 10 The architecture for self-supervised learning. Note that the augmented views are originated from either same or different inputs by random switching

where β and ω are tuning parameters, $\mathcal{L}_{CL}$ and $\mathcal{L}_{BCE}$ are contrastive and binary cross entropy losses, respectively. Here, the contrastive loss for a positive pair is defined as:

$$\mathcal{L}_{CL} = -log \frac{e^{sim(\mathbf{x}_i', \mathbf{x}_i'')/\tau}}{\sum_{j=1}^{N_\mathcal{B}} \left(e^{sim(\mathbf{x}_i', \mathbf{x}_j')/\tau} + e^{sim(\mathbf{x}_i', \mathbf{x}_j'')/\tau} \right) \cdot y} \tag{10}$$

where $sim(\mathbf{u}, \mathbf{v}) = {}^{\mathbf{u}^\top \mathbf{v}}/_{\|\mathbf{u}\|\|\mathbf{v}\|}$ denotes the cosine similarity, and the binary cross entropy loss is defined as:

$$\mathcal{L}_{BCE} = -y \cdot \log \hat{y} - (1 - y) \cdot \log (1 - \hat{y}) \tag{11}$$

where y and $\hat{y}$ are the pretext label and the predicted labels for output $\mathbf{z}$, respectively.

5 Experimental framework

In this section, we present the dataset of laminar-turbulent flow measurements, experimental protocols of different learning strategies and metrics utilised for validating our approach.

5.1 Thermographic measurement dataset

Our dataset comprises 23.468 non-labelled and 356 labelled samples where each sample is 512 × 512 × 1 dimensional IR image collected with the thermographic measurement specifications already described in Sect. 3. However, for the efficient use of the computing sources, images have been resized to 128 × 128 × 1 in our experiments.

As shown in Fig. 11, some samples contain scars, shadows, salt & pepper noises and contrast burst regions, demonstrating that realistic laminar-turbulent flow observation scenarios are subject to high noise. Besides, a laminar flow area may occur brighter or darker as compared to the regions in a turbulent flow. Due to some effect (e.g. shadowing the sun) it is even possible that, in one part of the image, the laminar flow area appears darker, and in another part, it appears brighter than the turbulent flow area.

5.2 Network training and validation protocol

We have conducted the experiments in the following order:

- *Supervised learning:* The ButterflyNet has been trained on the labelled samples of laminar-turbulent flow measurements in order to compare its performance against the benchmark segmentation architectures including U-Net (Ronneberger et al. 2015), DoubleU-Net (Jha et al. 2020), and Attention U-Net (Schlemper et al. 2019).
- *Self-supervised learning:* The ButterflyNet and the best performing benchmark architecture have been taken as base networks to train the self-supervised framework on the non-labelled samples.
- *Supervised fine-tuning:* The model weights of ButterflyNet and the best performing benchmark architecture have been initialised with the base network weights obtained from the regarding self-supervised models. Later, the ButterflyNet and the benchmark have been fine-tuned with a supervised learning strategy on the labelled data.

When conducting all of the experiments, the datasets have been randomly split into three different partitions: 80% for training, 10% for validation and 10% for testing. Furthermore, after shuffling the samples across the partitions, each experiment has been repeated multiple times in order to report mean and variance values in the obtained results.

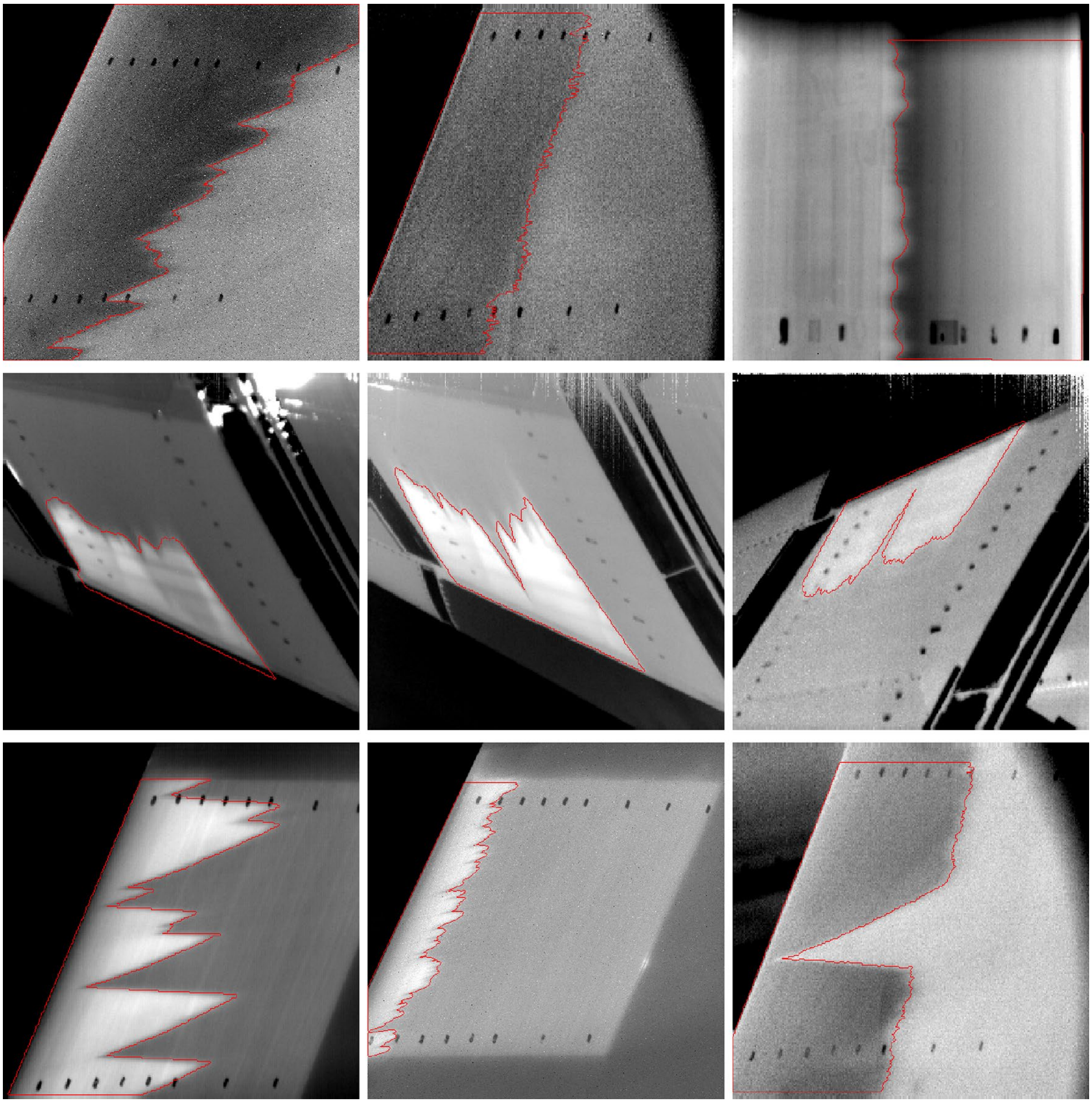

Fig. 11 Thermographic measurement examples from wind tunnel and flight test experiments: i. top and bottom row: wind tunnel ii. center row: vertical stabiliser from AFLoNext Project (AFLoNext 2018a). Note that the red flow-separation lines were semi-automatically drawn as ground-truths by an internal software of our institution. In principle, the software took some pixel samples selected by human experts for each flow region as input, and it accordingly drew laminar flow boundary after statistical analysis on the selected pixels. Finally, if mislocalisation happened in the separation lines, human experts corrected them in an iterative way

5.2.1 Data augmentation

It has been utilised due to the following reasons: (i) in case of supervised learning or fine-tuning, augmentation has been applied to mitigate the labelled data insufficiency and overfitting issue, (i) whereas, in self-supervised learning, SimCLR variants need series of spatial, geometric and appearance transformations in augmentation policy to learn visual representations efficiently (Chen et al. 2020). That is why, in our experiments, image rotation, horizontal and vertical flipping and shifting, image cropping and resizing, blurring, colour distortion and coarse dropout have been applied to generate various augmented views of an input image.

5.2.2 Network initialisation and optimisation

For the network architectures considered in this study, adaptive moment estimation (*ADAM*) (Kingma and Ba 2014) with a batch size of 64 has been used for training optimisation. During the ADAM optimisation, the learning rate has been initialised in the range [0.00001 – 0.01] and divided by 10 after reaching a plateau. The maximum number of training epochs has been set to 200 with the validation loss monitored to determine when to stop the learning processes. *Glorot Uniform* (Glorot and Bengio 2010) has been preferred in the weight initialisation of the networks except for supervised fine-tuning. While training the ButterflyNet, the initial size of the convolution filter has been determined as $F_0 \in \{16, 24, 32\}$, and the total number of the decoder and encoder layers has been set to $N_L \in \{3, 4, 5\}$. In supervised learning and fine-tuning, the mean-absolute error has been preferred as a loss function. In self-supervised learning, the tuning parameters of the contrastive and binary cross-entropy losses have been sought in the range of β, $\omega \in [0 - 1]$, and the temperature parameter has been set to $\tau \in \{0.001, 0.01, 0.1, 1\}$.

5.2.3 Hardware and software specifications

Experiments have been conducted on a system equipped with 128Gb RAM, NVIDIA^{TM} Tesla V100 GPU, running under UBUNTU 18.04 LTE and using the KERAS framework with TENSORFLOW version 2.3.1 (Abadi et al. 2016).

Besides, the following python libraries have been exploited: NUMPY, SCIPY, PANDAS and SCIKIT-LEARN for numerical processing, ALBUMENTATION for data augmentation, SEABORN and MATPLOTLIB for data visualisation. For providing the reproducible machine learning pipeline for the interested audiance, those software dependencies have been included in a DOCKER container (Anderson 2015), as can be seen in the project source code[1].

5.3 Evaluation metrics

The networks examined in our study have been evaluated on the basis of Pixel Accuracy (*PA*), Sørensen-Dice coefficient (*SDC*), Intersection over Union (*IoU*), Precision and Recall, such that:

$$PA = \frac{\sum_{i=0}^{K} p_{ii}}{\sum_{i=0}^{K} \sum_{j=0}^{K} p_{ij}} \tag{12}$$

where p_{ij} is the number of pixel of class i predicted as belonging to class j and K is the number of foreground classes.

$$IoU = \frac{|A \cap B|}{|A \cup B|} \tag{13}$$

$$SDC = 2\frac{|A \cap B|}{|A| + |B|} \tag{14}$$

where A and B denote the prediction and ground-truth segmentation maps ranging between $[0 - 1]$. Note that, in binary segmentation, *SDC* is equal to F_1-score which is the harmonic mean of precision and recall:

$$Precision = \frac{TP}{TP + FP} \tag{15}$$

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

where *TP*, *FP* and *FN* refer to the true positive, false positive and false negative fractions in segmentation maps.

6 Results and discussion

In this section, we present the laminar-turbulent flow segmentation results achieved via the examined networks. In Sect. 6.1, optimum hyper-parameters have been sought for the ButterflyNet. In Sect. 6.2, benchmark architectures have been compared with the ButterflyNet, and in Sect. 6.3, the effect of self-supervised learning for the performance improvement has been demonstrated.

6.1 Examining butterflyNet with various parametric scenarios

First of all, the best performing encoder and decoder parameters of the ButterflyNet have been searched by comparing different initial filter sizes (F_0) and total number of block layers (N_L) in supervised learning scenario. As summarised in Table 1, as the number of layers or filter size increases, a trend towards improvement in segmentation performance can be noticed. In terms of *IoU* and *SDC*, the highest performance can be achieved when the initial filter size and the number of layers are $F_0 = 24$ and $N_L = 5$, respectively. However, it has come with over 40 million network parameters, meaning more complexity in training and inference. That is why the ButterflyNet with $F_0 = 32$ and $N_L = 4$ has been rather chosen for the further experiments and the benchmark comparison due to its significant gain in total number of network parameters in return of negligible performance drop in segmentation.

Additionally, since the ButterflyNet has comprised two cascaded networks, $WING_1$ and $WING_2$, quantitative comparison among the outputs of them has been done to examine if the cascading had a positive impact in flow segmentation. As given in Table 2, $WING_2$ is outperforming $WING_1$, proving that feeding the element-wise multiplication of the input and the output of $WING_1$ into $WING_2$, as described in Eq. 1, has increased the flow segmentation performance.

[1] https://github.com/ridvansalihkuzu/butterflynet

Table 1 Searching encoder and decoder parameters in ButterflyNet

Network parameters		Total Number of Network Parameters	Average evaluation metrics (%)				
F_0	N_L		PA	*IoU*	*Prec*	*Recall*	*SDC*
16	3	5, 457, 219	96.73	88.11	91.05	96.48	93.68
24	3	8, 745, 599	96.64	88.17	91.25	96.33	93.71
32	3	12, 476, 539	96.12	87.79	91.67	95.40	93.49
16	4	8, 785, 286	96.96	88.40	91.73	96.13	93.88
24	4	15, 817, 442	96.61	88.73	91.72	96.40	93.99
32	4	24, 637, 694	96.99	89.21	91.94	96.81	94.31
16	5	20, 743, 689	96.50	88.24	90.78	96.96	93.77
24	5	41, 843, 621	97.32	89.40	91.96	96.70	94.40
32	5	70, 113, 793	96.86	89.02	90.88	97.80	94.20

Besides, the qualitative analysis in Fig. 12 verifies that masking the input with the output of $WING_1$ and later feeding it into $WING_2$ could eliminate some unwanted artefacts when automatically localising the separation boundary of laminar flow from turbulent flow.

6.2 Comparing butterflyNet with benchmark networks

In order to demonstrate the effectiveness of our proposed flow segmentation architecture, the quantitative comparison among the benchmark architectures have been conducted after running the tests 20 times for each architecture. As summarised in Table 3, ButterflyNet has excelled all the segmentation architectures in all evaluation metrics, and the second best architecture has become U-Net 3+ (Huang et al. 2020). Besides, as shown in Fig. 13, the ButterflyNet is computationally less expensive than 4 of the benchmarks with total number of 15.11 Giga-Flops, and more memory efficient than 3 of the benchmarks with 24.64 million parameters.

Moreover, the kernel density estimation (KDE) drawn by the segmentation scores acquired from each benchmark architecture is illustrated in Fig. 14. It is worth mentioning that KDE of ButterflyNet has the lowest variance as compared to other architectures, which means consistency of predictions is higher in our proposed architecture. Moreover, Attention U-Net (Schlemper et al. 2019) has also relatively low variance. On the other hand, U-Net (Ronneberger et al. 2015) and U2-Net (Qin et al. 2020) are the worse performing segmentation architectures in our experiments (Fig. 15).

6.3 Self-supervised learning and supervised fine-tuning

As already demonstrated in Eq. 8, the pretext task in our self-supervised learning strategy is the binary decision about if the inputs are augmented from the same view or not. Thus, as an initial step, the self-supervised network weights have been generated by training the model with different combinations of loss-tuning parameters, β for tuning contrastive loss and ω for tuning cross-entropy loss, as described in Eq. 9.

When the self-supervised model (SSM) weights were ready, the MLP Projection Head has been cut out, and the remaining parts of the model have been utilised to initialise the ButterflyNet weights except for the decoder part at $WING_2$ where random initialisation has been done. After that, the ButterflyNet has been fine-tuned with the labelled data for laminar-turbulent flow segmentation.

If $WING_2$ results reported in Table 2 are considered as the baseline performance of the ButterflyNet, all the fine-tuned supervised models initialised with SSM weights have resulted in better performance in flow segmentation except the model initialised with *SSM-Butterfly-01* as summarised in Table 4. In other words, the existence of contrastive loss ($\beta > 0$) in SimCLR variants of self-supervised learning is indispensable for outperforming the baseline.

In the model *SSM-Butterfly-05*, binary cross-entropy loss has been ignored ($\omega = 0$) in order to have only contrastive loss as proposed in the original SimCLR architecture (Chen et al. 2020). Even though such network initialisation with the original SimCLR strategy is still outperforming as shown in the last line of Table 4, the best flow segmentation performance has been accomplished after the network initialisation with *SSM-Butterfly-02*

Table 2 Quantitative comparison of $WING_1$ and $WING_2$ in ButterflyNet

Output taken at	Average evaluation metrics (%)				
	PA	IoU	Precision	Recall	SDC
$WING_1$	96.7 ± 0.8	87.7 ± 2.1	91.7 ± 1.9	95.3 ± 1.5	93.5 ± 1.5
$WING_2$	97.0 ± 0.5	89.2 ± 1.6	91.9 ± 1.4	96.8 ± 0.8	94.3 ± 1.0

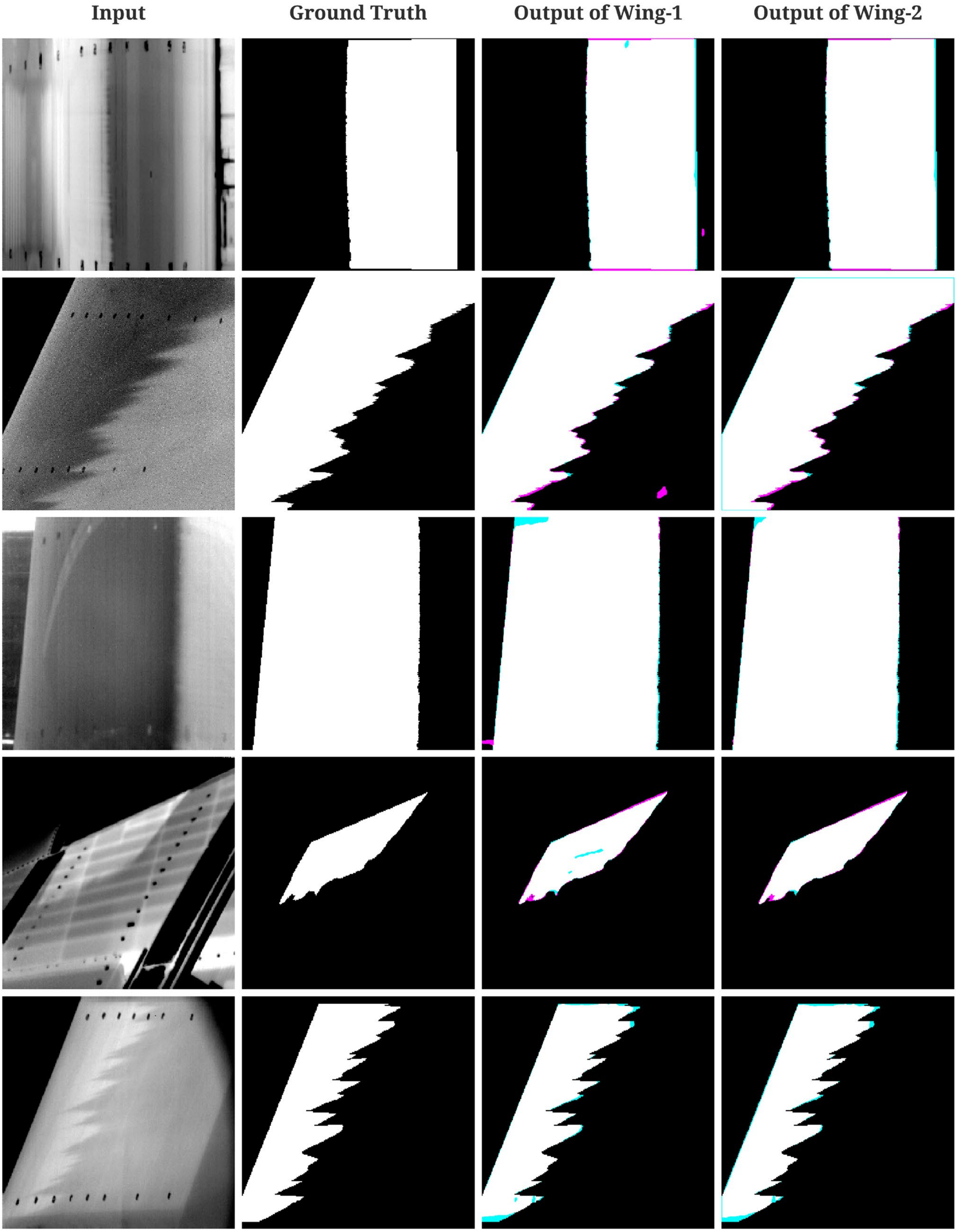

Fig. 12 Qualitative comparison of $WING_1$ and $WING_2$ in terms of flow segmentation. Note that $WING_2$ has generated less artefacts than $WING_1$ where the false positives (FP) are magenta and false negatives (FN) are cyan

Table 3 Quantitative comparison among the benchmark architectures

Benchmark architectures	Average evaluation metrics (%)				
	PA	IoU	Precision	Recall	SDC
U-Net (Ronneberger et al. 2015)	94.9 ± 1.5	83.0 ± 3.9	87.3 ± 3.7	94.5 ± 3.8	90.6 ± 2.4
U2-Net (Qin et al. 2020)	94.8 ± 1.2	81.9 ± 3.8	85.6 ± 3.3	95.0 ± 2.0	90.0 ± 2.3
R2 U-Net (Alom et al. 2019)	96.2 ± 1.3	86.6 ± 3.5	89.7 ± 2.8	96.1 ± 1.7	92.8 ± 2.0
U-Net 3+ (Huang et al. 2020)	96.4 ± 0.8	87.0 ± 3.2	90.7 ± 2.8	95.5 ± 2.3	93.0 ± 1.9
Trans U-Net (Chen et al. 2021)	95.8 ± 1.0	85.4 ± 4.5	89.0 ± 3.0	95.5 ± 1.8	92.1 ± 2.0
DoubleU-Net (Jha et al. 2020)	95.9 ± 1.1	86.0 ± 3.1	89.8 ± 2.4	95.4 ± 2.0	92.5 ± 1.8
Attention U-Net (Schlemper et al. 2019)	96.1 ± 0.9	86.1 ± 2.1	89.5 ± 1.9	95.9 ± 1.5	92.6 ± 1.2
ButterflyNet	97.0 ± 0.5	89.2 ± 1.6	91.9 ± 1.4	96.8 ± 0.8	94.3 ± 1.0

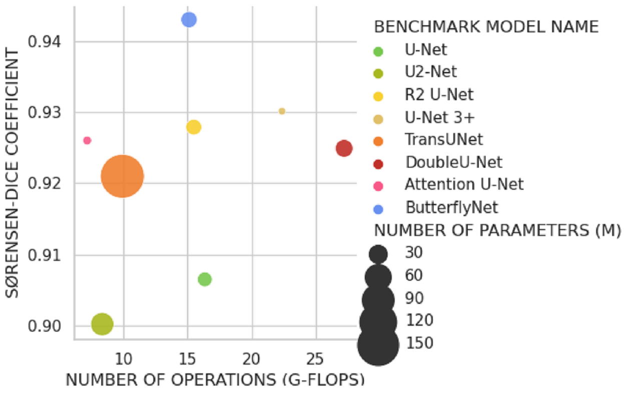

Fig. 13 Sórensen-Dice coefficient vs. computational complexity where G-FLOPs stands for the number floating-point operations required for a single forward pass, and the size of each ball corresponds to the model complexity

where the contrastive and binary cross-entropy losses have been summed with different weights, $\beta = 0.1$, $\omega = 0.9$, respectively.

When the supervised fine-tuning has been conducted for another benchmark network, U-Net 3+ Huang et al. (2020), the similar patterns occurred as summarised in Table 5, which means that flow segmentation performs best when $\beta = 0.1$ and $\omega = 0.9$ for self-supervised learning losses. It is also worth mentioning that the *SSM-Butterfly-02* and *SSM-UNet3+-02* which are subject to those same β and ω value have led to the minimum loss in the pretext task learning, proving that the combination of contrastive and binary cross entropy losses in SimCLR provides better self-supervision than original SimCLR.

Another important impact of self-supervised learning is the reduction in the network variance. For instance, while supervised learning of ButterflyNet has resulted in a

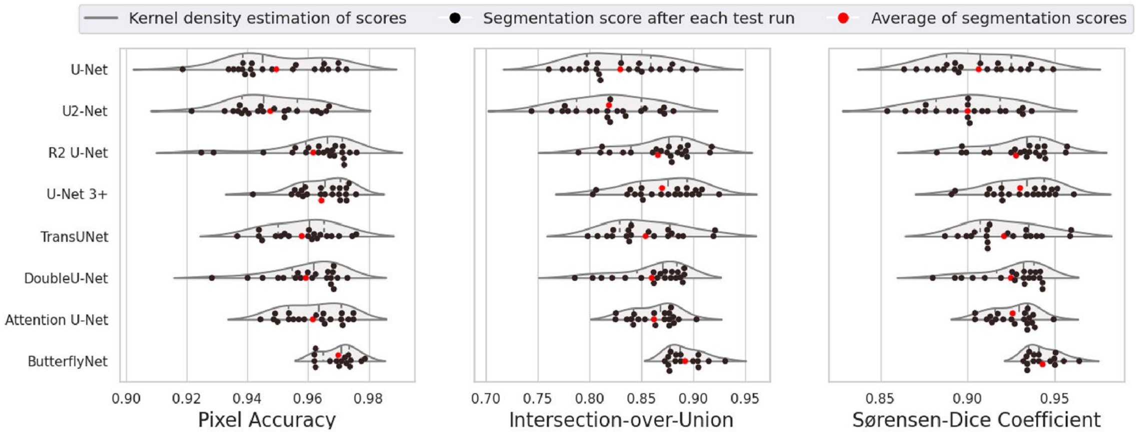

Fig. 14 Kernel density estimations drawn by the segmentation scores taken from each test run in terms of the evaluation metrics PA, IoU and SDC. The comparison demonstrates that ButterflyNet has the highest average segmentation performance and the lowest variance as compared to the other architectures

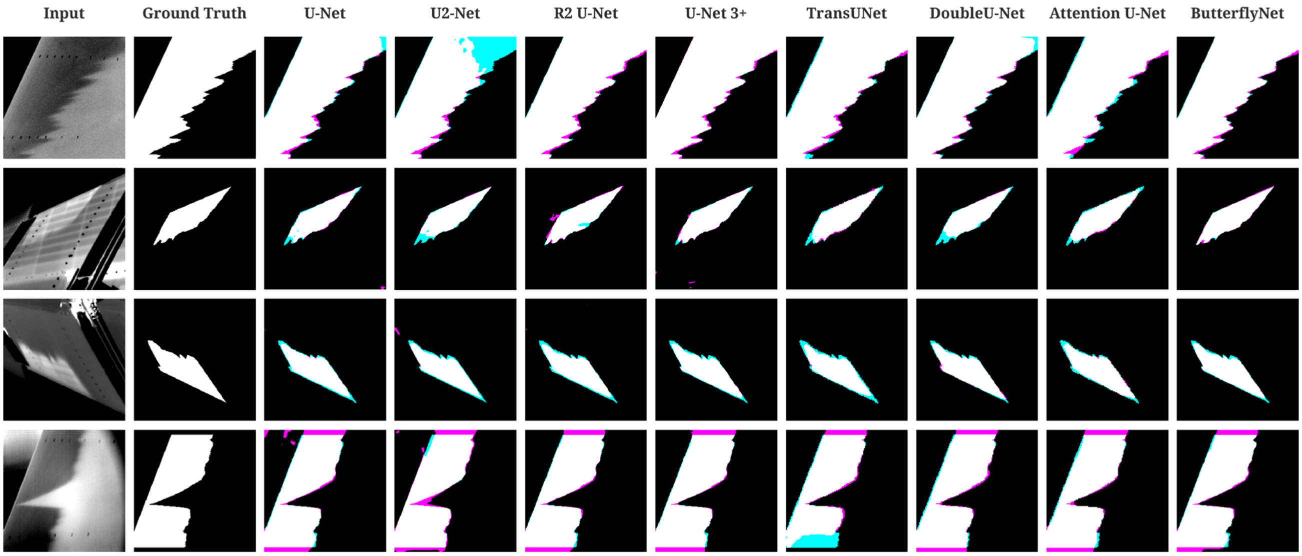

Fig. 15 Qualitative comparison of benchmark architectures. In the ground truth, the laminar flow regions are denoted as white (positive condition), while the rest is black (negative condition). Similarly, in the comparison of architectures, *TP*s are white, *TN*s are black, *FP*s are magenta and *FN*s are cyan

Table 4 Self-supervised learning of ButterflyNet on the pretext task with different combinations of loss tuning parameters (Left) and the corresponding flow segmentation performances after supervised fine-tuning with labelled data on those models (Right)

Self-supervised learning					Supervised Fine-tuning on a given Self-supervised Model				
Self-supervised Model ID	Tuning		Loss for Pretext Task	Accuracy for Pretext Task	Average Evaluation Metrics (%)				
	β	ω			PA	IoU	Precision	Recall	SDC
SSM-Butterfly-01	0.0	1.0	0.3076	0.9419	95.84 ± 0.63	87.46 ± 2.06	90.52 ± 1.93	96.41 ± 0.90	93.36 ± 1.16
SSM-Butterfly-02	0.1	0.9	**0.1988**	0.9843	**97.34** ± 0.53	**91.17** ± 1.39	**93.43** ± 1.15	**97.47** ± 0.68	**95.41** ± 0.77
SSM-Butterfly-03	0.5	0.5	0.3442	0.9921	97.00 ± 0.36	90.11 ± 1.34	93.26 ± 1.31	96.42 ± 0.62	94.80 ± 0.78
SSM-Butterfly-04	0.9	0.1	0.4769	0.9905	97.09 ± 0.45	89.70 ± 1.71	92.68 ± 1.63	96.62 ± 0.93	94.60 ± 0.96
SSM-Butterfly-05	1.0	0.0	0.3936	0.9998	97.06 ± 0.39	89.73 ± 1.56	92.36 ± 1.61	96.98 ± 0.89	94.60 ± 1.15
Baseline (random init. without self-supervised weights)					97.00 ± 0.54	89.21 ± 1.65	91.94 ± 1.41	96.81 ± 0.77	94.31 ± 1.01

The best performing loss and metrics are shown in bold

standard deviation of ±1.65 in IoU and ±1.01 in SDC, the network initialisation with *SSM-Butterfly-02* and supervised fine-tuning of ButterflyNet has suppressed those values to ±1.39 for IoU metric and ±0.77 for SDC metric as demonstrated in Table 4. Similarly, using *SSM-UNet3+-02* for network initialisation has reduced the standard deviation of IoU from ±3.23 to ±1.97 and of SDC from ±1.87 to ±1.13 for U-Net 3+-based flow segmentation as summarised in Table 5. Such reduction implies that self-supervised learning improves not only the segmentation performance but also the consistency in different predictions by reducing the risk of overfitting caused by the presence of a limited number of ground-truth data in the supervised training.

7 Conclusions

In this paper, we handled the automatic separation of different flow regions over flight-wings and stabilisers using the thermographic flow observations and applying deep learning techniques, because detection of flow distribution has crucial importance for optimising the wing & stabiliser geometry and improving the flight efficiency. Since the laminar-turbulent flow measurements are usually exposed to high noise and variance across different thermographic observations, the existing efforts for the automation of flow localisation in the literature had lack of reproducibility, or they only achieved semi-automation with a strong

Table 5 Self-supervised learning of U-Net 3+ on the pretext task with different combinations of loss tuning parameters (Left) and the corresponding flow segmentation performances after supervised fine-tuning with labelled data on those models (Right)

Self-Supervised Learning					Supervised Fine-tuning on a given Self-supervised Model				
Self-supervised Model ID	Tuning		Loss for Pretext Task	Accuracy for Pretext Task	Average evaluation metrics (%)				
	β	ω			PA	IoU	Precision	Recall	SDC
SSM-UNet3+-01	0.0	1.0	0.3168	0.8979	95.32 ± 0.93	86.36 ± 2.62	92.18 ± 2.74	93.01 ± 1.77	92.60 ± 1.99
SSM-UNet3+-02	0.1	0.9	**0.1969**	0.9887	**96.58** ± 0.84	**89.49** ± 1.97	**92.34** ± 1.84	**96.65** ± 1.54	**94.48** ± 1.13
SSM-UNet3+-03	0.5	0.5	0.4263	0.9904	96.60 ± 0.83	88.83 ± 1.84	91.57 ± 1.71	96.79 ± 0.76	94.10 ± 1.01
SSM-UNet3+-04	0.9	0.1	0.4901	0.9931	96.01 ± 0.79	87.40 ± 2.52	89.18 ± 2.51	97.77 ± 2.04	93.19 ± 1.47
SSM-UNet3+-05	1.0	0.0	0.3990	0.9957	96.39 ± 1.05	88.16 ± 3.20	89.92 ± 3.06	97.71 ± 2.11	93.65 ± 1.82
Baseline (random init. without self-supervised weights)					96.44 ± 0.82	87.00 ± 3.23	90.69 ± 2.76	95.54 ± 2.31	93.01 ± 1.87

The best performing loss and metrics are shown in bold

dependency on human experts for guidance and correction. To overcome such difficulties, we introduced a novel encoder-decoder architecture, namely ButterflyNet, for the automatic segmentation of flow regions. In order to compensate for the lack of manually labelled data caused by the time-consuming nature of analysis on IR thermography samples, we customised a self-supervised strategy and, in this way, we benefited from diverse sets of raw thermographic observations to improve the robustness of flow segmentation. The proposed approach achieved 97.34% pixel accuracy and 91.17% intersection-over-union in the automatic separation of laminar flow regions from the remaining regions.

The future study could extend this approach to detect other flow phenomena (e.g. Tollmien- Schlichting or Cross-Flow-Transition) as well as to separate automatically laminar and turbulent flow regions with coordinate points after re-labelling the data accordingly, such as by including the reference coordinate markers into the training process.

Acknowledgements This work has been supported by the Helmholtz Artificial Intelligence Cooperation Unit under Project ID HAICU-HLST 3023535. The aircraft test activities received funding from the European Community's Seventh Framework Programme FP7/2007-2013, under grant agreement no. 604013, AFLoNext project. The authors would like to thank Zeynep Oba, Darshan Thummar and Martin Kruse for their great support and feedback.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Abadi M, Barham P, Chen J, et al (2016) Tensorflow: a system for large-scale machine learning. In: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), pp 265–283

AFLoNext (2018a) 2nd generation active wing. http://www.aflonext.eu/

AFLoNext (2018b) Active Flow- Loads Noise control on next generation wing. https://cordis.europa.eu/project/id/604013/reporting

Alom MZ, Yakopcic C, Hasan M et al (2019) Recurrent residual U-net for medical image segmentation. J Med Imag 6(1):014,006

Anderson C (2015) Docker [software engineering]. IEEE Softw 32(3):102-c3

Badrinarayanan V, Kendall A, Cipolla R (2017) SegNet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39(12):2481–2495

Bakunowicz J, Szewczyk M (2015) Flow and structure deformation research of a composite glider in flight conditions

Barth HP (2021) Beeinflussung des laminar-turbulenten Grenzschichtumschlags durch kontrollierte Anregung stationärer Querströmungsinstabilitäten. PhD thesis, Universität Göttingen

Bégou G, Deniau H, Vermeersch O et al (2017) Database approach for laminar-turbulent transition prediction: navier-stokes compatible reformulation. AIAA J 55(11):3648–3660

Cao H, Wang Y, Chen J, et al (2021) Swin-Unet: unet-like pure transformer for medical image segmentation. arXiv preprint arXiv:2105.05537

Chen J, Lu Y, Yu Q, et al (2021) TransUNnet: Transformers make strong encoders for medical image segmentation. arXiv preprint arXiv:2102.04306

Chen LC, Papandreou G, Kokkinos I et al (2017) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFS. IEEE Trans Pattern Anal Mach Intell 40(4):834–848

Chen T, Kornblith S, Norouzi M, et al (2020) A simple framework for contrastive learning of visual representations. In: *International Conference on Machine Learning*, PMLR, pp 1597–1607

Chen X, Williams BM, Vallabhaneni SR, et al (2019) Learning active contour models for medical image segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp 11,632–11,640

Crawford BK, Duncan GT, West DE et al (2015) Robust, automated processing of IR thermography for quantitative boundary-layer transition measurements. Exp Fluids 56(7):1–11

De Luca L, Carlomagno G, Buresti G (1990) Boundary layer diagnostics by means of an infrared scanning radiometer. Exp Fluids 9(3):121–128

Dollinger C, Balaresque N, Sorg M (1992) Thermographic Boundary Layer Visualisation of Wind Turbine Rotorblades in Operation. J Aircr 29(2):161–171

Dollinger C, Balaresque N, Sorg M et al (2018) IR thermographic visualization of flow separation in applications with low thermal contrast. Infrared Phys Technol 88:254–264

Emmons HW (1951) The laminar-turbulent transition in a boundary layer-part I. J Aeronaut Sci 18(7):490–498

Frederick MA, Banks DW, Garzon G et al (2015) Flight tests of a supersonic natural laminar flow airfoil. Meas Sci Technol 26(6):064,003

Fu J, Liu J, Tian H, et al (2019) Dual attention network for scene segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 3146–3154

Gardner AD, Wolf CC, Heineck JT et al (2020) Helicopter rotor boundary layer transition measurement in forward flight using an infrared camera. J Am Helicopter Soc 65(1):2–14

Gartenberg E, Wright RE (1994) Boundary-layer transition detection with infrared imaging emphasizing cryogenic applications. AIAA J 32(9):1875–1882

Gleichauf D, Dollinger C, Balaresque N et al (2020) Thermographic flow visualization by means of non-negative matrix factorization. Int J Heat Fluid Flow 82(108):528

Gleichauf D, Oehme F, Sorg M et al (2021) Laminar-turbulent transition localization in thermographic flow visualization by means of principal component analysis. Appl Sci 11(12):5471

Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, pp 249–256

Goel V, Weng J, Poupart P (2018) Unsupervised video object segmentation for deep reinforcement learning. Adv Neural Inf Process Syst 31:5683–5694

Grawunder M, Reß R, Breitsamter C (2016) Thermographic transition detection for low-speed wind-tunnel experiments. AIAA J 54(6):2012–2016

Gur S, Wolf L, Golgher L, et al (2019) Unsupervised microvascular image segmentation using an active contours mimicking neural network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 10,722–10,731

He K, Gkioxari G, Dollár P, et al (2017) Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2961–2969

Hinton GE, Sabour S, Frosst N (2018) Matrix capsules with EM routing. In: International conference on learning representations

Huang H, Lin L, Tong R et al (2020) UNet 3+: a full-scale connected unet for medical image segmentation. ICASSP 2020–2020 IEEE international conference on acoustics. Speech and Signal Processing (ICASSP), IEEE, pp 1055–1059

Hung WC, Tsai YH, Liou YT, et al (2019) Adversarial learning for semi-supervised semantic segmentation. In: 29th British Machine Vision Conference, BMVC 2018

Isensee F, Petersen J, Klein A, et al (2018) nnU-Net: Self-adapting Framework for U-Net-based medical image segmentation. arXiv preprint arXiv:1809.10486

Jha D, Riegler MA, Johansen D, et al (2020) DoubleU-net: A deep convolutional neural network for medical image segmentation. In: 2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS), IEEE, pp 558–564

Joseph LA, Borgoltz A, Devenport W (2016) Infrared thermography for detection of laminar-turbulent transition in low-speed wind tunnel testing. Exp Fluids 57(5):77

Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980

Kirillov A, He K, Girshick R, et al (2019) Panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 9404–9413

Koch S, Mühlmann P, Lefebvre-Albaret F, et al (2020) BLADE Flight Test Instrumentation for Transition Detection

Kruse M, Munoz F, Radespiel R, et al (2018) Transition prediction results for sickle wing and NLF (1)-0416 test cases. In: 2018 AIAA Aerospace Sciences Meeting, p 0537

Li Y, Chen X, Zhu Z, et al (2019) Attention-guided unified network for panoptic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 7026–7035

Lin TY, Dollár P, Girshick R, et al (2017) Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2117–2125

Liu Z, Lin Y, Cao Y, et al (2021) Swin transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 10,012–10,022

Noh H, Hong S, Han B (2015) Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp 1520–1528

Punn NS, Agarwal S (2022) Modality specific U-Net variants for biomedical image segmentation: a survey. Artificial Intelligence Review pp 1–45

Qin X, Zhang Z, Huang C et al (2020) U2-Net: going deeper with nested U-structure for salient object detection. Pattern Recogn 106(107):404

Quast A (2006) Detection of transition by infrared image techniques. Tech Soar 30(1–2):33–38

Quast AW (1987) Detection of transition by infrared image technique. In: ICIASF'87-12th International Congress on Instrumentation in Aerospace Simulation Facilities, pp 125–134

Raffel M, Merz CB (2014) Differential infrared thermography for unsteady boundary-layer transition measurements. AIAA J 52(9):2090–2093

Ren S, He K, Girshick R et al (2016) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149

Ren Y, Yu Y, Guan H (2020) DA-CapsUNet: a dual-attention capsule u-net for road extraction from remote sensing imagery. Remote Sens 12(18):2866

Richter K, Schülein E (2014) Boundary layer transition measurements on hovering helicopter rotors by infrared thermography. Exp Fluids 55(7):1–13

Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image cCmputing and Computer-assisted Intervention, Springer, pp 234–241

Schlemper J, Oktay O, Schaap M et al (2019) Attention gated networks: learning to leverage salient regions in medical images. Med Image Anal 53:197–207

Schrauf G (2005) Status and perspectives of laminar flow. Aeronaut J 109(1102):639–644

Schrauf GH, von Geyr H (2021) Simplified hybrid laminar flow control for the A320 Fin. Part 2: Evaluation with the eN-method. In: AIAA Scitech 2021 Forum, p 1305

Seitz A (2007) Freiflug-experimente zum übergang laminar-turbulent in einer tragflügelgrenzschicht. PhD thesis, TU Braunschweig

Simon B, Filius A, Tropea C et al (2016) IR thermography for dynamic detection of laminar-turbulent transition. Exp Fluids 57(5):93

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556

Souly N, Spampinato C, Shah M (2017) Semi supervised semantic segmentation using generative adversarial network. In: Proceedings of the IEEE International Conference on Computer Vision, pp 5688–5696

Tong G, Li Y, Chen H et al (2018) Improved U-NET network for pulmonary nodules segmentation. Optik 174:460–469

Traphan D, Meinlschmidt P, Schlüter F, et al (2015) High-speed measurements of different laminar-turbulent transition phenomena on rotor blades by means of infrared thermography and stereoscopic PIV. In: 10th Pacific Symposium on Flow Visualization and Image Processing

Wang H, Xie S, Lin L et al (2022) Mixed transformer u-net for medical image segmentation. ICASSP 2022–2022 IEEE international conference on acoustics. Speech and Signal Processing (ICASSP), IEEE, pp 2390–2394

Wang Y, Zhang J, Kan M, et al (2020) Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp 12,275–12,284

Wolf CC, Mertens C, Gardner AD et al (2019) Optimization of differential infrared thermography for unsteady boundary layer transition measurement. Exp Fluids 60(1):19

Xia H, Ma M, Li H et al (2022) MC-Net: multi-scale context-attention network for medical CT image segmentation. Appl Intell 52(2):1508–1519

Zhang L, Liu A, Xiao J, et al (2021) Dual encoder fusion u-net (defunet) for cross-manufacturer chest x-ray segmentation. In: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, pp 9333–9339

Zhao H, Shi J, Qi X, et al (2017) Pyramid scene parsing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 2881–2890