# Learning Fluid Flow Visualizations from In-flight Images with Tufts

Jongseok Lee[1,*], W.F.J. Olsman[2,*] and Rudolph Triebel[1]

***Abstract*—To better understand fluid flows around aerial systems, strips of wire or rope, widely known as tufts, are often used to visualize the local flow direction. This paper presents a computer vision system that automatically extracts the shape of tufts from images, which have been collected during real flights of a helicopter and an unmanned aerial vehicle (UAV). As images from these aerial systems present challenges to both the model-based computer vision and the end-to-end supervised deep learning techniques, we propose a semantic segmentation pipeline that consists of three uncertainty-based modules namely, (a) active learning for object detection, (b) label propagation for object classification, and (c) weakly supervised instance segmentation. Overall, these probabilistic approaches facilitate the learning process without requiring any manual annotations of semantic segmentation masks. Empirically, we motivate our design choices through comparative assessments and provide real-world demonstrations of the proposed concept, for the first time to our knowledge. The project website can be accessed via the link: https://sites.google.com/view/tuftrecognition/.**

***Index Terms*—Aerial Systems: Applications; Computer Vision for Automation; Object Detection, Segmentation and Categorization; Probability and Statistical Methods; Aerodynamics.**

## I. Introduction

OVER the last decade, the performance of computer vision techniques improved sharply, leading to new application areas of robotics and automation. In particular, advances in deep learning introduced several frameworks capable of visualizing complex 3D flow phenomena [1]. Such advances in aerodynamics are relevant for aerial robotics, which requires fundamental understanding of the underlying physics behind the flow phenomena. The efficient design of novel systems [2], safe operations of UAVs [3], and reducing noise in rotary wing systems [4], are a few examples that illustrate the subject's relevance in the current aerial systems research.

To advance our understanding of fluid flows around complex aerial systems in real flights, this work focuses on visualizing their local fluid flow topology via the application of tufts – one of the oldest and simplest experimental methods to visualize the flow on a surface. Tufts and their variants like flow cones are small pieces of wire or rope that are attached to a surface, aligning themselves with the local flow velocity [5]. Typically, the orientation and shape of the tufts are captured with a camera and the images are analyzed to obtain insight into the local flow topology on the surface [6], [7]. Examples are notably found in many works of aerodynamicists [6], [8], [7].

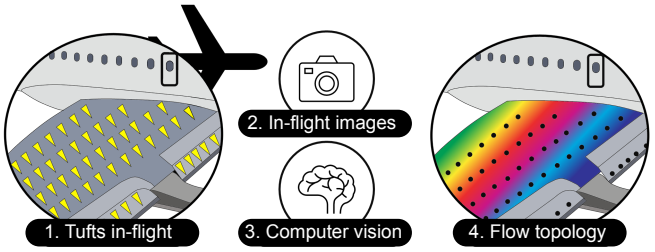


Fig. 1. Tufts are placed on an in-flight vehicle such as aircraft. Visual sensors capture the video sequences, and semantic segmentation of tufts is performed with learning algorithms for visualizing the local flow topology.

In this paper, we propose a learning system to advance the applicability of tufts for in-flight fluid flow visualizations. Given in-flight images with tufts, our system automatically generates semantic segmentation masks of individual tufts for flow visualizations (Fig. 1). In contrast to a manual analysis by aerodynamicists [5], the proposed system enables automation. Such automation has the added benefit of flexibility, reproducibility, lack of human bias, and scalability to the vast number of images and tufts per image. Moreover, this concept is not restricted to controlled environments, where model-based computer vision techniques such as template matching, homography, and application of static masks may be sufficient [7], [8]. The proposed system thereby scales the applications of tuft methods to challenging environments outside the laboratories, where large changes in lighting conditions, perspective, and background scenes are inevitable.

The development of such a system is motivated by a large amount of challenging data we collected for concrete applications of flow visualization. Our data consists of (a) in-flight images from the tail surface of a helicopter, which we collected using another manned helicopter flying in close formation, and (b) the images from a UAV flying in the stratosphere, i.e., the flight sequence back-and-forth from ground to approximately 20 km altitude. Importantly, these images are challenging for model-based computer vision techniques due to large variations between the different images. Moreover, annotated training data is not readily available for semantic segmentation. The problem also involves classifying small objects, i.e., tufts, which have the same appearance, but different labels (Fig. 2). Therefore, the applicability of end-to-end supervised deep

Manuscript received: November 10, 2022; Revised February 21, 2023; Accepted April 4, 2023 .

This paper was recommended for publication by Editor Hyungpil Moon upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the Helmholtz AI under grant agreement ID ZT-I-PF-5-1. Special thanks to Konstantin Kondak, Zhang Kai, Omar Hedeya, Maximilian Durner, Jianxiang Feng and Nari Song for their great support.

[1] Institute of Robotics and Mechatronics, German Aerospace Center (DLR), 82234 Weßling, Germany. jongseok.lee@dlr.de

[2] Institute of Aerodynamics and Flow Technology, German Aerospace Center (DLR), 38108 Braunschweig, Germany. jurrien.olsman@dlr.de

*Co-first authors.

Digital Object Identifier (DOI): see top of this page.

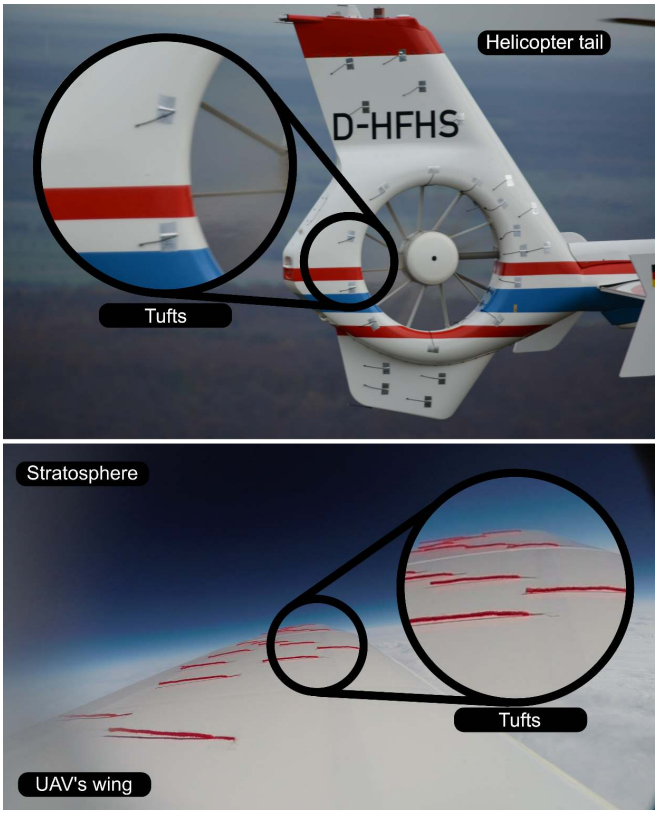


Fig. 2. Top: helicopter tail rotor (the Fenestron duct). Bottom: wing area of a UAV, captured approximately 20 km altitude. For example, some tufts (wires or ropes) are zoomed in. Different class labels are given to each individual tuft. The goal is semantic segmentation of these small and repetitive objects with similar appearance, but different class labels.

learning techniques, like Mask-RCNN, may also be limited.

To this end, we propose a semantic segmentation pipeline that addresses the aforementioned challenges. The key idea is to divide the problem of semantic segmentation into three steps, namely (a) active learning-based object detection using uncertainty sampling, (b) uncertainty-driven label propagation for object classification, and (c) weakly supervised instance segmentation with so-called uncertainty maps. In the paper, we describe in detail, how these probabilistic approaches allow the training of the overall pipeline without requiring any annotations of semantic segmentation masks. Instead, we reduce the annotation efforts to the labeling of fewer bounding boxes, and only a single image where all the class labels are specified. Empirically, several comparative assessments are presented within these three steps, which motivates our design choices. Lastly, quantitative and qualitative assessments of the proposed concept are provided as final demonstrations. We note that our focus is on examining the feasibility of computer vision techniques for the proposed application concept, while aerodynamic results will be presented in a future publication.

In summary, the contributions of this paper are (a) a novel application concept of learning flow visualizations from in-flight images with tufts, (b) a semantic segmentation pipeline, based on probabilistic approaches, for addressing the practical challenges of our scenario, (c) the collection and sharing of in-flight images from the manned helicopter and stratospheric UAV flights, and (d) several experimental results to validate each component of the system, including quantitative and qualitative characterizations of the overall performance.

## II. Related Work

**In-flight Flow Visualization Techniques** Due to the transparency of fluids, their flow patterns are invisible to humans. Thus, several experimentation methods have been devised to visually acquire the flow patterns. In wind tunnel facilities, which replicate the interaction between air and flight models by blowing air using large tubes, certain mediums like smoke, oil flows, particles, etc., have been combined with optical measurement techniques to visualize the air flows. Likewise, such technologies are being applied to aerial systems in-flight, as a way to obtain data from real flights. Here, flow cones and tufts, oil flows, liquid crystals, sublimating chemicals, and smoke are often used [5]. Amongst these methods, for simplicity and low cost, this work automatizes tufts methods.

**Automatic Tuft Recognition for Flow Visualization** Over the past decades, the applications of tufts were mostly on qualitative flow visualizations where the images are manually analyzed by aerodynamicists [5]. Recently, however, quantitative analysis via image processing methods are also being examined. Vey *et al.* [7] and Wieser *et al.* [6] manually specified the position of each tuft and its geometric orientation, and used mean angles to perform line integration convolution. A rule-based system is developed, where given the masks and anchors of the surface containing tufts, threshold-based foreground extraction, and color-based identification schemes are used [9]. Steinfurth *et al.* [8] assumed the location of each tuft to be known, and applied the Prewitt method for shape extraction. These works provide strong evidence that gaining quantitative insights from tufts are a viable option. Our main novelty is a learning-based solution, which relaxes the assumptions of rule-based systems. With this, we demonstrate the real-world applicability at the scale of real manned helicopter flights and stratospheric missions of a UAV.

**Reducing the Annotation Efforts** End-to-end semantic segmentation models heavily rely on large amounts of annotated training data, which are often not readily available. In such cases, the generation of synthetic data is a compelling option, where annotations are readily available. However, tufts are deformable and hence, the supports for such objects are limited along with the well-known problem of sim2real gap [10]. Other alternatives are semi-supervised models [11], which utilize only a small number of annotations, while weakly supervised models [12] use some weaker form of labels. In our approach, we heavily build upon the advances in these two domains. The proposed three-step pipeline exploits active learning for bounding box detection as foreground extractor of tufts. Here, a small number of single-class bounding boxes are relatively easier to annotate. Then, label propagation is devised by combining classical image matching, uncertainty estimates [13], and the idea of key-frames for multi-class semantics. The final step involves instance segmentation, where we employ weakly supervised learning models [14]. In this way, we demonstrate how probabilistic approaches help in real-world applications with limited annotated data.

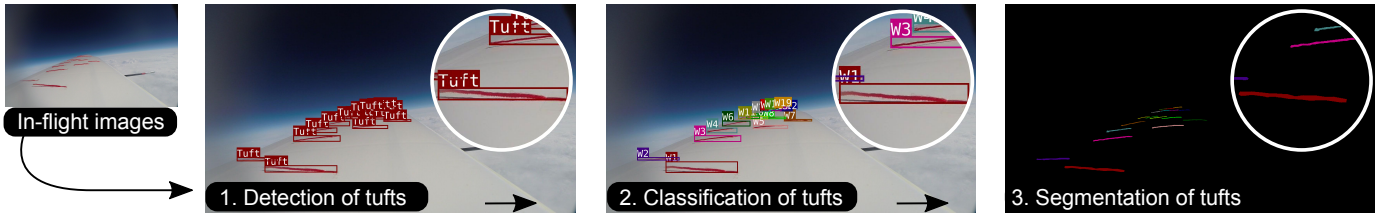


Fig. 3. An overview of our pipeline. Instead of supervised learning, using deep models, we devise a semantic segmentation pipeline with (a) detection of tufts for foreground extraction, (b) classification for semantic information, and then (c) instance segmentation. We note that deep learning-based object detection and segmentation methods can also classify different objects. However, in our real-world application scenario, training data is not readily available, and labeling annotation masks is very costly. Thus, the proposed pipeline is to reduce the annotation efforts down to labeling of fewer bounding boxes, and only a single image where all the class labels of each tuft are specified, i.e., no annotations of semantic segmentation masks are required.

## III. Data, Task, and Challenges

In this section, we provide brief information about data collection, task definition, and associated challenges.

**Data Generation** We performed flight experiments with EC135-ACT/FHS helicopter equipped with 56-81 tufts, while the Bo105 helicopter was used as a camera platform. The resolution of captured images was 6000x4000 pixels. The goal of this first scenario is to analyze the complex in-flight aerodynamic behavior of the anti-torque device known as Fenestron or fan-in-fin [4]. As a second scenario, we gathered the data from a stratospheric UAV [15] with 19 tufts on its wing. These images were captured with a GoPro during the approximately 145 minutes long flight to the stratosphere, i.e., the atmosphere at about 20 km altitude. This dataset is to show the generality of our method. Stratospheric flight is also an area where the flow phenomena cannot be completely reproduced in wind-tunnels [16]. We refer to the project website for more details on data collection from test flights.

**Task Definition** From flight testing, we typically acquire thousands of images under different conditions, from which, useful aerodynamic data can be extracted. Such information can be obtained automatically by a processing methodology that recognizes the exact shape of the tuft in pixels. Moreover, in order to examine the temporal behavior of each individual tuft, the same tuft in many images is to be classified. In other words, we are interested in the flow topology of an aerial system, and so, each individual tuft must be monitored separately over time. Therefore, from a computer vision perspective, the problem involves semantic segmentation as described in Fig. 2. We note that after processing the images, aerodynamic results can be presented as streamlines on the surface [6] or polar histograms for individual tufts [7]. Here, streamlines are the velocity vector fields of airflow while the polar histograms of the airflow directions are for quantitative data. Therefore, fulfilling the herein-defined computer vision task allows fluid flow visualization as well as quantitative characterizations.

**Challenges with In-flight Data** Observing the collected data from real flights, we find that first, the rule-based systems [7], [6], [9], [8] cannot be directly applied, because we cannot assume known masks and anchors due to large variations in perspectives, relative positions, and illuminations between images. Clearly, if the sizes and locations of foreground masks change between the images, we cannot assume static masks and anchors. These large variations in perspective and position are caused by different relative positions of the following helicopter, to enable a view inside the Fenestron duct. Positional shifts also occur due to unintended movements of the following helicopter by turbulence and gusts. While supervised learning techniques are current golden standards for semantic segmentation, generating large amounts of manual annotations is not feasible, e.g., one could imagine annotating semantic segmentation masks for thousands of images as in Fig. 2. Hence, the direct applicability of end-to-end supervised deep learning techniques is limited. Lastly, the problem involves classifications of objects (tufts) with similar appearance, but different labels depending on their relative location. To address this, we conceive that appropriately combining model-based techniques may be a solution rather than learning such patterns from data only.

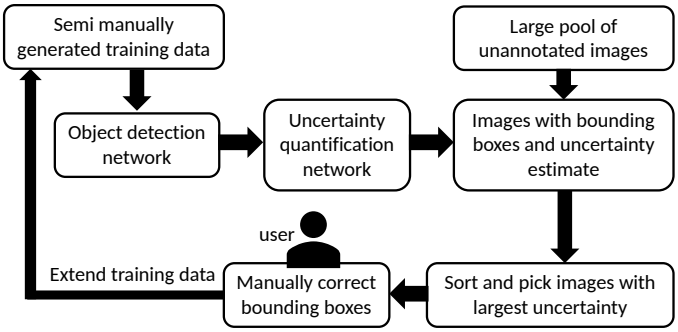


Fig. 4. The proposed pool-based active learning with coarsely annotated pool data. The initial coarse annotations are generated manually with the aid of feature-based image matching.

On the other hand, we assume that the data is acquired either from videos or similar image capturing. Furthermore, the analysis is intended for offline and therefore, the runtime or efficiency is not an important criterion as far as our pipeline is computationally tractable. On the contrary, the system-level requirements are (1) to maximize the accuracy while (2) reducing manual efforts in producing annotations. Given these primitives, the next section presents our solution to the aforementioned practical challenges from real flights.

## IV. The Proposed Learning Pipeline

As depicted in Fig. 3, we propose to break this challenging problem into three feasible sub-problems. We introduce our solutions to each individual problem next.

### A. Tuft Detection

In this step, the objective is to locate the presence of objects with a bounding box, and further classify the located object

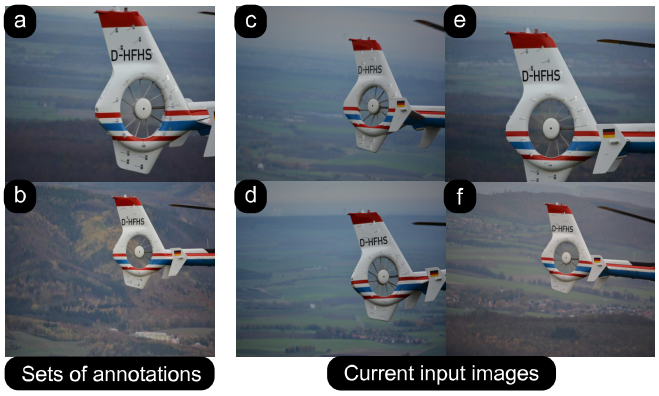


Fig. 5. Given an annotated image as a source image, we propagate the labels to the target image via image matching. Due to various perspective changes, having multiple source images with annotations can increase the accuracy of image matching. For example, image a would match image e easier than image b. Likewise, image b can easily match image f.

in an image. This task is typically performed by a neural network trained in a supervised manner. Hence, bounding box annotations are required for training. Although generating such annotations are more affordable than labeling for semantic segmentation, our goal is to reduce the manual human efforts as much as possible.

To this end, we propose a pool-based active learning framework with coarsely annotated pool data. An overview is provided in Fig. 4. In the first stage, we manually annotate a single image and propagate this annotation to other images via image matching. Like this, we automatically generate a large pool of coarsely annotated images. For some images, this will work, for many it will not. In any case, it only requires manual correction of a user instead of generating annotations from the stretch. With this initial training data for supervised learning, an object detector can be trained. Here, we simplify the detection task by leaving out the classification task, i.e. we treat different tufts as a single class "Tuft". This avoids the problem of detecting objects with the same appearance but different class labels and increases the instances of supervised data points. Moreover, annotating bounding boxes for a single class is less time-consuming.

The next stage then involves an active learning loop. Here, our algorithm queries a user for new annotations from a pool of data. Often, the subset of data is queried/selected with measures of uncertainty [17], i.e. the most uncertain data to neural networks are prioritized [10]. With the newly obtained data, existing train data are extended, and the network is retrained. Repeating the loop, active learning automatically selects the most informative data for the network to learn from. The proposed pipeline uses coarsely annotated pool data (Fig. 4). As such coarse annotations are error-prone, our pipeline involves a user for corrections, only when selected by the active learning algorithm. In this way, we reduce human supervision more. What further motivates active learning is the proposed simplification of object detection into a single class. As such, active learning is well suited as we can avoid under-performance in multi-label set ups [10].

Concretely, two crucial components are uncertainty estimation for neural networks and a selection criterion for label query. The former is a framework, which is based on Bayesian formulation [18], [13]. Given any trainable parameters of neural networks $\boldsymbol{\theta}$ and training data $\mathcal{D}$, these frameworks estimate the weight posteriors $p(\boldsymbol{\theta}|\mathcal{D})$. Then, the prediction uncertainty $p(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D})$ can be quantified:

$$p(\boldsymbol{y}^*|\boldsymbol{x}^*, \mathcal{D}) = \int p(\boldsymbol{y}^*|\boldsymbol{x}^*, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{D}) d\boldsymbol{\theta}, \tag{1}$$

for a output $\boldsymbol{y}^*$ from a new input $\boldsymbol{x}^*$. If applied to object detectors, we obtain a probabilistic object detector (POD) [19] that delivers the calibrated probabilities of class labels and the covariance matrices of the associated bounding box locations. This information is then used in generating the label queries. For this, we utilize the information scores for both classification $\mathcal{U}_{j,cls}$ and regression $\mathcal{U}_{j,reg}$ tasks respectively for all object instances $j$ in an image [10]:

$$\mathcal{U}_{j,cls} = \sum_{i=1}^{|\mathcal{C}|} \mathcal{H}\left(p(c_i|\boldsymbol{x}^*, \mathcal{D})\right) \;\&\; \mathcal{U}_{j,reg} = \mathcal{H}\left(p(\mathbf{b}|\boldsymbol{x}^*, \mathcal{D})\right),$$

which rely on the Shannon Entropy $\mathcal{H}(\cdot)$ by assuming categorical distributions over the classes $c_j$ and Gaussian distributions for the bounding boxes $\mathbf{b}$ (a set of two-pixel coordinates describing a box). These scores can then be aggregated per image to select the most uncertain data.

### B. Tuft Classification

Having obtained the bounding boxes of all tufts with a single class "Tuft", the goal is to classify them into their unique identification labels, like "Tuft W1", "Tuft W2", etc.

To achieve this, we propose an algorithm with uncertainty-driven label propagation. We provide the entire description of the pipeline in Algorithm 1. Intuitively, given an annotated image that contains all the class labels, we can propagate the labels by means of matching this annotated image to the current test image. Due to the temporal nature of our data (like videos), we avoid learning the classification of objects with a similar appearance. Yet, we have to deal with certain challenges such as image matching can fail under severe perspective changes, illumination, etc. (see Fig. 5). Our algorithm is designed to address such challenges, while still reducing the annotation efforts down to one single image.

Specifically, in Algorithm 1, the inputs are current test image $\boldsymbol{I}_T$ and their corresponding outputs of probabilistic object detection $p(\boldsymbol{y}^* \mid \boldsymbol{x}^*, \mathcal{D}), \boldsymbol{b}$, and the outputs are class labels for each bounding box $\{c_j\}_{j=1}^{L}$. Additionally, we use the so-called key-frames, i.e., a set of available images with annotations $\{\boldsymbol{I}_{S,i}\}_{i=1}^{K}$. For this, we initially annotate one single image: $K = 1$, and update until a specific desired number of annotated images is reached. As to reduce the manual annotations by humans, the key-frame update will be performed automatically. The algorithm achieves this by a criterion that assesses the reliability of automatically generated annotations. We stress that annotations with multi-class labels are more expensive than single-class labels with bounding boxes only. Finally, we take the total L number of tufts as another input since in our application scenario, the number of tufts on an aerodynamic vehicle is known.

**Algorithm 1:** Tuft Classification Algorithm with Uncertainty Driven Label Propagation

**input :** $\boldsymbol{I}_T$: the current test image from a video stream; $p(\boldsymbol{y}^* \mid \boldsymbol{x}^*, \mathcal{D}), \boldsymbol{b}$: outputs of probabilistic object detection; $\boldsymbol{x}^* = \boldsymbol{I}_T$ and $\boldsymbol{b} \leftarrow \mathbb{E}[p(\boldsymbol{y}^* \mid \boldsymbol{x}^*, \mathcal{D})]$ $\{\boldsymbol{I}_{S,i}\}_{i=1}^{K}$: The key-frames as a set of source images with annotations; $L$: The total number of tufts.

**output:** $\{c_j\}_{j=1}^{L}$: The class labels for each bounding box; $\{\boldsymbol{I}_{T,i}\}_{i=1}^{K+1}$: New key-frames after evaluating results on the current image $\boldsymbol{I}_T$.

```
begin
    /* Key-frame based Image matching                                          */
    for all the K images in the key-frames do
        T_i, C_i ← image_matching(I_{S,i}, I_T) ∀i ;   // Image matching; Results in transformations T_i and costs C_i
    end
    T ← arg min({C_i}_{i=1}^K) ;                          // Select the result with the least cost
    {c_j}_{j=1}^L ← label_propagation(b, T) ;          // Label all L bounding boxes using Hungarian algorithm
    /* Is the results reliable? Multi-criteria decisions (MCD) are based on (1) If all L tufts are
       detected, (2) If the confidence is high, and (3) If the matching costs are low.   */
    RS ← MCD(p(y* | x*, D), C_i, L) ;                     // Evaluate the Reliability Score (RS)
    if RS is True then
        {I_{T,i}}_{i=1}^{K+1} ← update_keyframe({I_{S,i}}_{i=1}^K, I_T, {c_j}_{j=1}^L, b) ;   // Update if reliable more than a threshold
end
```

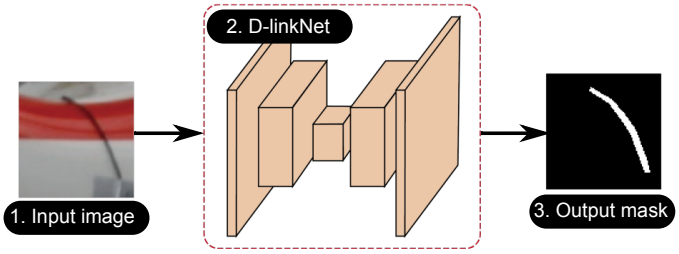


Fig. 6. Tuft segmentation. We predict the segmentation masks using D-LinkNet. The network is trained in a weakly supervised manner.

For the main body of Algorithm 1, we first match the current image with images from the key-frames, which contain annotations of class labels. This results in multiple transformations and their costs, one pair each per source image. We pick a single source image $\boldsymbol{I}_{S,i}$ with the lowest matching cost or error. Then, using this, we propagate the labels by keeping the available boxes from the input, while obtaining the multi-class labels by examining the overlapping areas and nearest points from the matching results. We use a Hungarian algorithm to assign each tuft one and exactly one label. This results in the first output of the algorithm: $\{c_j\}_{j=1}^{L}$. As the final step, we project the obtained results using three criteria, namely the number of detections, uncertainty of the bounding box predictions, and the error of image matching. These three criteria capture the reliability of the obtained results, and the key-frames are updated only when the results are deemed more reliable than a pre-specified threshold. Intuitively, the obtained results can be re-used as the source image for matching with the next images, only if reliable or accurate via quantified uncertainty estimates.

In this way, the idea of keyframes mitigates the failures of image matching, as the keyframes consist of multiple source images from different conditions. By further combining a decision-making criterion, we can automatically generate source images within the key-frames. This reduces any manual annotations of costly multi-class labels.

### C. Tuft Segmentation

Given the locations of each tuft and their class labels, we now perform instance segmentation. So, within the cropped images from each bounding box, we group the pixels that belong to the object, separating the objects of interest from the background. This differs from applying the segmentation method on the whole image directly.

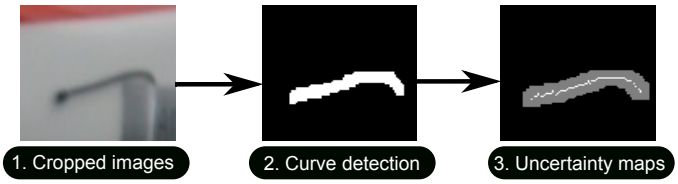


Fig. 7. Generation of annotations for tuft segmentation. A classical curve detection is combined with uncertainty masks (gray). The loss does not use any pixels within the uncertainty maps.

Our pipeline involves a network that predicts a segmentation mask of tufts given a cropped input image (Fig. 6). To train this network, we first employ a line extraction via a curve line detector [20], which generates a coarse result. Then, D-linknet [14] is learned from the coarse segmentation as a weaker form of annotation. In this way, we can perform instance segmentation without annotations of segmentation masks. Concretely, first, a curve detection [20], is adopted because it extracts curvilinear structure by utilizing an explicit model for line and their surroundings, as opposed to a simple model of only a line. Then, we train D-LinkNet, which has been originally developed to extract long and thin objects in the context of remote sensing. The architecture adopts the widely used encoder-decoder structure with dilated convolutions to connect the pre-trained encoder to the decoder, while the intermediate layers use a series of dilated convolutions in order to deal with tiny objects in an image and maintain the spatial details at each scale. The network is trained using the scribble-based weakly supervised loss function [21], i.e., the partial binary cross entropy (PBCE) loss, with what we call the auxiliary loss (AL):

$$L(Y, S) = \text{PBCE}(Y, S) + \text{AL}(Y, S), \tag{2}$$

where Y is the curve mask from the curve detector and S is the prediction mask of the CNN. The PBCE is defined as,

$$\text{PBCE}(Y, S) = -\sum_{x \in \omega} Y_x \log(S_x) + (1 - Y_x)\log(1 - S_x),$$

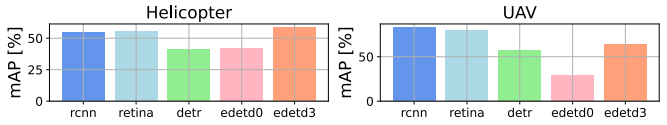


Fig. 8. The results of object detection for five different methods. Higher the better for mAP metric.

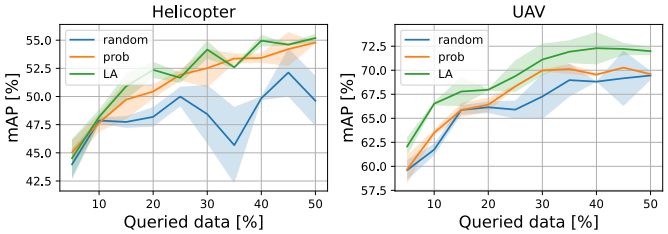


Fig. 9. The results of active learning for different sampling strategies. Higher the better for the curves.

where $x$ stands for a pixel in $\omega$ - the regimes that are known, i.e. the pixels that belong to either the background or the object. This is provided by the curve detector, but we also apply a buffer along the detected line and set it as an unknown regime (see Fig. 7). As the PBCE loss is defined in the regimes that are known as foreground and background, the network learns to generate these regimes by minimizing the loss. Intuitively, the network can learn from examples where the curve detection performs well, in order to separate between the foreground and background of tufts, and such patterns of lines persist within the unknown regimes. Finally, an additional loss function, AL, is used for improvements, which we ablate within our experiments in Section V.

## V. Results

Now, our quantitative results are presented for each individual step. Final performance is then examined. The implementation details can be found on our project website.

**Results on Tuft Detection** We examine if certain object detectors are more suited for the given data. For this, we manually annotate 1000 images each for both the helicopter and the UAV data, and randomly choose 500 data points for training and the other 500 images for testing. For the evaluation, the commonly used mAP is used. The selected models are RCNN, RetinaNet, efficientDet-v0 (edetd0), efficientDet-v3 (edetd3) [23] and DETR [24], which are all widely applied detectors. All the implementations are based on open-source code from Pytorch and Detectron2 and are trained with the ResNet backbone. We use the batch size of two and a learning rate of 0.001 and 0.01 are chosen for the helicopter and the UAV data respectively. The results of model comparisons are shown in Fig. 8. We find that edetd3 yields the highest mAP for the helicopter data, while RCNN achieved the highest mAP for the UAV data. While the results of each model differ depending on the data, we find that DETR and edetd0 under-performs as they are not designed for small objects. In our scenario, we find that detectors for small objects with adaptable anchors are desirable.

For active learning, we use the same setup as the model comparison experiments, e.g., the metric, annotations, etc. Yet, the splits of 0.2, 0.7, and 0.1 ratios are chosen for the test set, the pool set, and the validation set respectively. Total ten active learning loops are used to query up to 50 percent of the total pool data. Three random seeds are repeated for different initialization of neural network training. For the baselines, we use three different strategies. These are the randomized selection from the pool set (random), and uncertainty-based sampling with the state-of-the-art uncertainty quantification methods, namely Monte-Carlo dropout (prob [18]) and Laplace Approximation (LA [13]). These baselines are to examine the influence of uncertainty estimates. Moreover, as we simplified the task to detect only one-single class, uncertainty estimates are one of the influential variables to examine. The results of active learning are reported in Fig. 9. Here, the queried data are depicted in percentage to the total amount of data. The error bars are depicted in shades. Overall, we examine that random selection is outperformed by other methods, while LA results in superior performance. This motivates the design choice of our method. Moreover, the results show that about 50% of overall data would result in 95% of the total performance. This is due to the redundancy in the data and motivates the use of active learning for reducing annotations.

**Results on Tuft Classification** For evaluating the tuft classification task using uncertainty-driven label propagation, we use the previously annotated 1000 images each for both the helicopter and the UAV data. Here, only one multi-class annotated image is used as a source, and other key-frames are chosen with our decision-making criteria [1]. Then, all the others are used for evaluation. The metric of mAP is chosen, which captures both the classification accuracy and the bounding box refinement. The number of keyframe images is ablated from two to twelve in order to show that additions of keyframes enable more accurate label propagation. To evaluate the selection of keyframes, we compare our probabilistic approaches to a random selection, which forms a baseline to show that careful selection of the keyframes can increase the accuracy of the label propagation.

Results are depicted in Fig. 11 for both the helicopter and the UAV data. First, we find that as the number of keyframes increases, the accuracy also increases. This comes with increased computational costs when compared to the use of one single reference image. However, such costs can be justified since the system requires not to be real-time. Second, we observe that random selection of the keyframes can lead to a decrease in the mAP metric because the keyframes are selected from the output of the probabilistic object detector. Inaccurate annotations as sources for image matching can deteriorate the performance (also characterized by a high error bar in a 'single' source image). On the contrary, our uncertainty-driven mechanism improves the accuracy by selecting only the bounding boxes that 'the detector is confident about'. 'LA' again outperformed 'prob' in this case. Overall, the results justify our design choices for reducing the number of required annotations in tuft classification.

[1]We note that more key-frames can be chosen from the available sets of annotations, at the cost of more manual efforts. But, our focus is to reduce such efforts for improving the applicability of our concept.

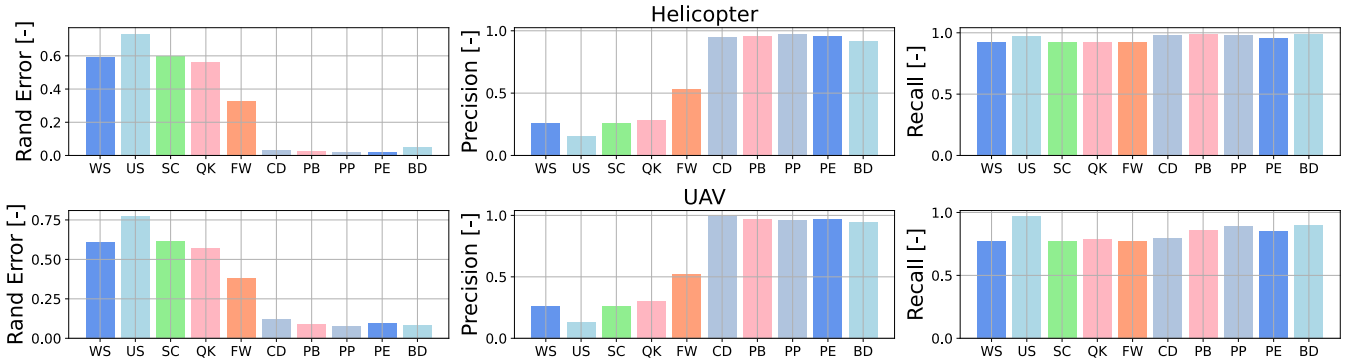


Fig. 10. *Rand* error [22], precision and recall are reported for the ten considered methods. The lower the better for the *rand* error, the higher the better for the precision and recall. Favorable results are observed for our approaches (CD, PB, PP, PE, and BD) over four alternatives.

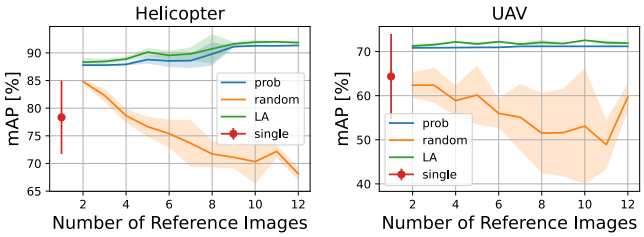


Fig. 11. The results of tuft classification using the proposed label propagation. Higher the better the curves.

TABLE I
SEGMENTATION EVALUATION RESULTS. THE LOWER THE BETTER FOR POINT DISTANCE. HIGHER THE BETTER FOR OTHER METRICS.

| | **Completeness** | **Correctness** | **Quality** | **Point distance** |
|---|---|---|---|---|
| | | *Helicopter In-Flight Images* | | |
| CD | 0.890±0.002 | 0.832±0.002 | 0.761±0.003 | 0.208±0.002 |
| BD | **0.970±0.001** | 0.881±0.009 | 0.828±0.007 | 0.113±0.002 |
| PB | 0.963±0.001 | 0.889±0.003 | 0.840±0.002 | **0.085±0.001** |
| PE | 0.930±0.004 | 0.881±0.002 | 0.840±0.001 | 0.176±0.002 |
| PP | 0.957±0.003 | **0.889±0.004** | **0.841±0.005** | **0.085±0.002** |
| | | *Stratospheric UAV In-Flight Images* | | |
| CD | 0.262±0.001 | 0.810±0.002 | 0.619±0.003 | 1.348±0.002 |
| BD | **0.846±0.005** | 0.893±0.004 | **0.775±0.001** | 0.185±0.002 |
| PB | 0.801±0.003 | 0.898±0.010 | 0.760±0.004 | 0.132±0.003 |
| PE | 0.799±0.004 | 0.894±0.001 | 0.756±0.004 | 0.137±0.001 |
| PP | 0.809±0.005 | **0.902±0.010** | 0.762±0.007 | **0.127±0.002** |

**Results on Tuft Segmentation** For tuft segmentation, we evaluate our approach against several baselines (from classical image processing methods to deep learning) and perform ablations on loss functions.

To do so, we annotate 1000 segmentation masks for evaluation only. In the training steps, we use the curve detection [20] on 1000 tuft patches. Batch size of eight was used with a learning rate of 0.0002. For the baselines, Felzenszwalb [25] (FW), Quick [26] (QK), Watershed [27] (WS), Slic [28] (SC), unsupervised [29] (US) and Curve Detection [20] (CD) are chosen. These are the baselines that do not require segmentation masks, and open-source implementations exist for generic segmentation tasks. Along with precision and recall, adapted random error is used as an evaluation metric. Regarding the ablations on AL, we include PBCE loss only (PB), with a point distance (PP), with the edges (PE), and binary cross entropy with dice loss (BD). For the metric, instead of the pixel-wise intersect of union, we adapt a center-line quality metric, which is (a) completeness – a measure of the percentage of true prediction to all the labels, (b) correctness – a measure of the percentage of true prediction to all other predictions, and (c) quality – a measure on the percentage of true prediction to all predictions and the labels. Moreover, we also use point distance in pixels as another metric to evaluate how well the algorithms are able to predict the start and the end point of the tufts. We explain these measures on the project website.

The results are in Fig. 10. We observe that for the *rand* error [22], the considered baselines perform poorly when compared to the ablation models. The same is observed for the precision while the recall is only slightly lower than the ablation models. The best model in terms of the *rand* error and the precision metrics is PP, which is the proposed method. In the recall metric, PB outperformed PP. The ablation results are reported in Tab. I. Again, we observe significant improvement in all the metrics when compared to CD. From the experiments, we however find that no single loss can be uniformly superior when projected to four success criteria. One clear observation is that the proposed learning method improves over the simple curve detector, thereby validating the proposed approach.

**Results on Final Performance** Lastly, we evaluate the final performance. Two supervised instance segmentation networks, namely Mask RCNN and Cascaded-mask RCNN, are used as baselines. Implementations were based on OpenMMLab toolbox. For training, the default hyperparameters of OpenMMLab are used, except that, the number of epochs had to be increased to 300, and image dimensions were not resized in order to deal with small objects. Both for training and evaluation, we annotated 50 images each per dataset. We note that these annotations are costly, i.e., approximately 2200 tufts had to be annotated with semantic segmentation masks. Five training-test splits of ratio 60-40 were used.

The results are reported in Tab II, where we observe that the supervised instance segmentation networks perform poorly. We attribute this to the lack of available training data. When no abundant training data is available, the results indicate that directly employing the end-to-end supervised deep learning techniques may pose challenges. On the other hand, with probabilistic approaches, we show how semantic segmentation can be performed without requiring any manual

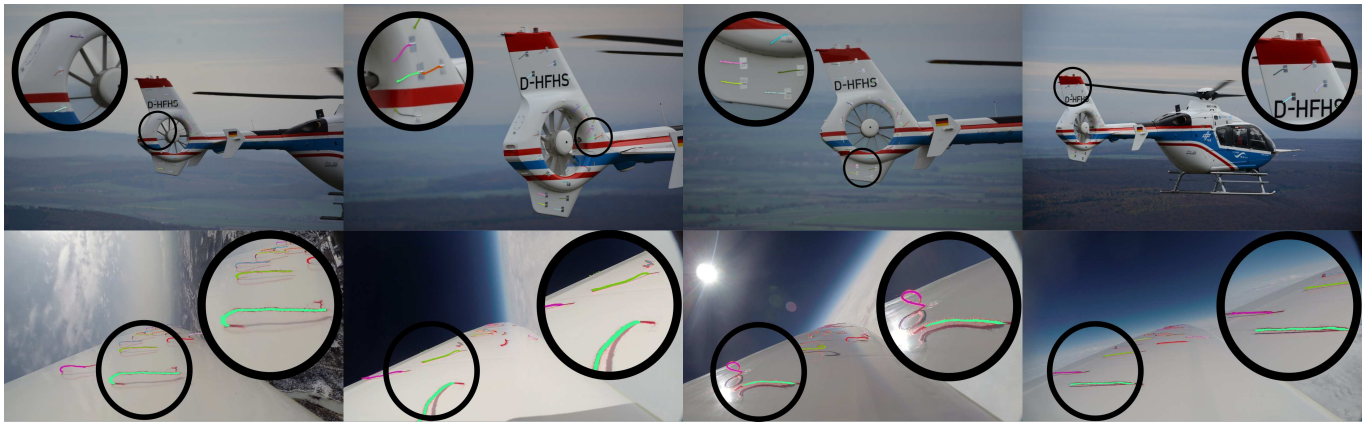


Fig. 12. Final results are illustrated for the helicopter (top) and the UAV (bottom). The relevant portion of the images is zoomed in. Colored overlays indicate the segmentation masks per class. Our method can perform the semantic segmentation tasks under severe perspective changes (as in helicopter data), foreground changes of tufts, and lighting conditions. Model-based technique alone may fail as shown in Fig. 10.

TABLE II
QUANTITATIVE RESULTS IN MAP. HIGHER THE BETTER. END-TO-END NETWORKS TRAINED USING SMALL AMOUNTS OF DATA ONLY.

| | **Mask RCNN** | **Cascaded RCNN** | **Ours** |
|---|---|---|---|
| Helicopter | 23.422±2.1822 | 11.198±2.4028 | **60.729±2.1894** |
| UAV | 34.032±1.2945 | 18.254±0.5844 | **58.343±2.4532** |

annotations of semantic segmentation masks. For qualitative results, Fig. 12 and the accompanying video demonstrate the overall performance, where we visually show that many of the tufts can be segmented with the correct semantics.

## VI. CONCLUSION

This paper presented a learning system to provide automatic evaluations of in-flight images with tuft for flow visualization. For the first time to our knowledge, we developed an automatic tuft recognition system for flow visualization of aerial systems during real test flights. To achieve this, we performed data gathering using two real application scenarios, namely a full-sized helicopter and a UAV flying in the stratosphere. Using probabilistic approaches, we show how the annotation efforts can be reduced significantly. Experimental results demonstrate how the devised approaches can address the identified challenges.

## REFERENCES

[1] C. Liu *et al*., "Deep learning approaches in flow visualization," *Advances in Aerodynamics*, vol. 4, no. 1, pp. 1–14, 2022.
[2] C. Ruiz *et al*, "Aerodynamic reduced-order volterra model of an ornithopter under high-amplitude flapping," *Aerospace Science and Technology*, vol. 121, p. 107331, 2022.
[3] F. Achermann *et al*, "Learning to predict the wind for safe aerial vehicle planning," in *ICRA*, 2019, pp. 2311–2317.
[4] W. F. J. Olsman, "Experimental investigation of fenestron noise," *Journal of the American Helicopter Society*, vol. 67, 2022.
[5] D. F. Fisher *et al*, "Flow visualization techniques for flight research," in *AGARD Symposium of the Flight Mechanics Panel on Flight Test Techniques*, no. NAS 1.15: 100455, 1988.
[6] D. Wieser *et al*, "Surface flow visualization on a full-scale passenger car with quantitative tuft image processing," in *SAE 2016 World Congress and Exhibition*. SAE International, apr 2016.
[7] S. Vey *et al*, "Extracting quantitative data from tuft flow visualizations on utility scale wind turbines," *Journal of Physics: Conference Series*, vol. 524, p. 012011, jun 2014.
[8] B. Steinfurth *et al*, "Tuft deflection velocimetry: a simple method to extract quantitative flow field information," *Experiments in Fluids*, vol. 146, no. 61, 2020.
[9] L. Chen *et al*, "Flow visualization and transient behavior analysis of luminescent mini-tufts after a backward-facing step," *Flow Measurement and Instrumentation*, vol. 71, p. 101657, 2020.
[10] J. Feng *et al*, "Bayesian active learning for sim-to-real robotic perception," *IROS*, 2022.
[11] S. Mittal *et al*, "Semi-supervised semantic segmentation with high-and low-level consistency," *IEEE PAMI*, vol. 43, no. 4, pp. 1369–1379, 2019.
[12] J. Zhang *et al*, "Weakly-supervised salient object detection via scribble annotations," in *CVPR*, 2020, pp. 12 546–12 555.
[13] J. Lee *et al*, "Estimating model uncertainty of neural networks in sparse information form," in *ICML*. PMLR, 2020, pp. 5702–5713.
[14] L. Zhou *et al*, "D-linknet: Linknet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction," in *CVPR Workshops*, 2018, pp. 182–186.
[15] S. Wlach *et al*, "Dlr hableg–high altitude balloon launched experimental glider," in *22nd Symposium on European Rocket and Balloon Programmes and Research*, no. SP-730. ESA, 2015, pp. 385–392.
[16] J. Lee *et al*, "Towards autonomous stratospheric flight: A generic global system identification framework for fixed-wing platforms," in *IROS*. IEEE, 2018, pp. 6233–6240.
[17] J. Gawlikowski *et al*, "A survey of uncertainty in deep neural networks," *arXiv preprint arXiv:2107.03342*, 2021.
[18] Y. Gal *et al*, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *ICML*. PMLR, 2016, pp. 1050–1059.
[19] H. Ali *et al*, "Bayesod: A bayesian approach for uncertainty estimation in deep object detectors," in *ICRA*. IEEE, 2020, pp. 87–93.
[20] C. Steger, "An unbiased detector of curvilinear structures," *IEEE PAMI*, vol. 20, no. 2, pp. 113–125, 1998.
[21] Y. Wei *et al*, "Scribble-based weakly supervised deep learning for road surface extraction from remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2021.
[22] I. Arganda-Carreras *et al.*, "Crowdsourcing the creation of image segmentation algorithms for connectomics," *Frontiers in neuroanatomy*, vol. 9, p. 142, 2015.
[23] M. Tan *et al*, "Efficientdet: Scalable and efficient object detection," in *CVPR*, 2020, pp. 10 781–10 790.
[24] C. Nicolas *et al*, "End-to-end object detection with transformers," in *ECCV*. Springer, 2020, pp. 213–229.
[25] P. F. Felzenszwalb *et al*, "Efficient graph-based image segmentation," *IJCV*, vol. 59, no. 2, pp. 167–181, 2004.
[26] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *ECCV*. Springer, 2008, pp. 705–718.
[27] P. Neubert *et al*, "Compact watershed and preemptive slic: On improving trade-offs of superpixel segmentation algorithms," in *ICPR*. IEEE, 2014, pp. 996–1001.
[28] A. Radhakrishna et al, "Slic superpixels," Tech. Rep., 2010.
[29] A. Kanezaki, "Unsupervised image segmentation by backpropagation," in *ICASSP*. IEEE, 2018, pp. 1543–1547.