# Face Verification Algorithms for UAV Applications: An Empirical Comparative Analysis

Julio Diez-Tomillo, Jose M. Alcaraz-Calero, and Qi Wang

***Abstract*—Unmanned Aerial Vehicles (UAVs) are revolutionising diverse computer vision use case domains, from public safety surveillance to Search and Rescue (SAR), and other emergency management and disaster relief operations. The growing need for accurate face verification algorithms has prompted an exploration of synergies between UAVs and face verification. This promises cost-effective, wide-area, non-intrusive person verification. Real-world human-centric use cases such as a "Drone Guard Angel" for vulnerable people can contribute to public safety management and offload significant police resources. These scenarios demand efficient face verification to distinguish correctly the end users for authentication, authorisation and customised services. This paper investigates the suitability of existing solutions, and analyses five state-of-the-art candidate face verification algorithms. Informed by the advantages and disadvantages of existing solutions, the paper proposes an extended dataset and a refined face verification pipeline. Subsequently, it conducts empirical evaluation of these algorithms using the proposed pipeline and dataset in terms of inference times and the distribution of the similarity indexes. Furthermore, this paper provides essential guidance for algorithm selection and deployment in UAV-based applications. Two candidate algorithms, ArcFace and FaceNet512, have emerged as the top performers. The choice between them will depend on the specific use case requirements.**

***Index Terms*—Face verification, UAV, Drone, Similarity index, Inference speed.**

## I. Introduction

Unmanned Aerial Vehicles (UAVs) have gained momentum in recent years as a novel and transformative technology, finding applications across diverse domains, from surveillance and security applications [1], [2] to Search and Rescue (SAR) operations [3]. Moreover, the demand for effective and accurate face verification algorithms has become increasingly pronounced. These algorithms have become widely used in different applications but they have several limitations like restricted coverage or the inability to track individuals, as it is mostly performed from fixed camera systems [4].

The combination of both technologies - UAVs and face verification - can achieve a breakthrough in the capacity to verify people at a low cost, covering large areas fast and from a high distance, thereby preserving the safety of the individual without being an intrusive method.

A number of use cases have explored this combination. For example, the Drone Guard Angel in the EU Horizon 2020 project ARCADIAN-IoT [5]. It is a public safety management service: an UAV will go to the position of a user who has requested the service, the user's face will be verified, and then the UAV will accompany the user ensuring that the user arrives at the destination safely. Furthermore, yet another use case can be surveillance network applications in the EU Horizon 2020 project 5G-INDUCE [6], where the UAV pilot identity will be verified before the start of the operation.

The emergence of these applications has underscored the need for highly accurate and fast face verification algorithms capable of functioning effectively with UAV video feeds. Some challenges arise while dealing with images from UAVs. For instance, the long distances from UAVs to human faces result in low-pixel resolution, making it more difficult to verify the users [7]. Moreover, UAVs must deal with adverse environmental factors such as low lighting conditions during night-time operations. Moreover, there can be rapid changes in the pose and positioning of a person or sudden movement of the UAV. Notably, there is a lack of practical solutions in the literature regarding face verification from UAVs. This research work is based on [8], which has provided a preliminary study regarding face verification algorithms. This paper expands that research by conducting a more comprehensive analysis of state-of-the-art algorithms.

In this research paper, five state-of-the-art face verification algorithms are compared, in terms of accuracy, inference and building time and the size of the weights. The analysis is performed using an extended dataset with videos at six fixed distances: 2, 5, 7, 10, 15 and 20 meters. In order to perform the experiments, an enhanced face verification pipeline has been designed and implemented to support different similarity index calculations. Furthermore, an up-to-date literature review on face verification algorithms is conducted. In summary, this study's principal contributions are as follows:

- Design and implementation of an enhanced face verification pipeline to conduct the experiments using different similarity index calculations.
- Expansion of the UAV recorded dataset (UAV-UWS) to add two additional distances to have a more comprehensive dataset.
- An empirical analysis of the inference time in a face verification pipeline using five state-of-the-art algorithms.

Manuscript received November 2, 2023; revised December 18. Date of publication January 12, 2024. Date of current version January 12, 2024.

Authors are with the School of Computing, Engineering and Physical Sciences (CEPS), University of the West of Scotland (UWS), Paisley, United Kingdom (e-mails: {julio.diez-tomillo, jose.alcaraz-calero, qi.wang}@uws.ac.uk).

This paper was presented in part at the International conference on Software, Telecommunications and Computer Networks (SoftCOM) 2023.

Digital Object Identifier (DOI): 10.24138/jcomss-2023-0165

- An empirical evaluation of five state-of-the-art face verification algorithms, yielding practical insights into their suitability for application in UAV-based scenarios.

The structure of this paper is outlined as follows. Section II provides an overview of state-of-the-art face verification algorithms. Section III presents the dataset expansion, the design of the pipeline, and the implementation. The testbed, experimental results and findings are presented and discussed in section IV. Finally, section V concludes the paper.

## II. Related Work

In recent years, the utilisation of UAVs has experienced a remarkable surge across numerous applications. One of the most important advancements attributed to drones is their ability to perform operations that were before extremely expensive at a considerably reduced cost. One of the use cases is border surveillance, where UAVs can cover extensive territories easily at a low cost [9]. Another notable application is Search and Rescue (SAR) operations, which used to require the deployment of helicopters at a high economic cost. By using UAVs the operations can be performed easier and explore larger areas in less time by employing multiple UAVs. Furthermore, the incorporation of object detection algorithms using optical or thermal cameras in the UAVs has significantly increased the effectiveness of these operations, resulting in a higher number of successful rescues [10], [11].

Several research papers have explored the utilisation of drones in face verification applications, often employing simple yet fast algorithms, like LBPH (Local binary patterns histograms) [12]. These algorithms are usually lightweight models that are suitable for embedded systems like a UAV companion computer. However, the situation transforms when a high-speed connection can be established between a Ground control station (GCS) like QGroundControl [13] and the UAV. In this context, high-performance algorithms with large models can be executed in a high-spec computer where the GCS or other software is running and the UAV will only have to send video [14].

Furthermore, prior research papers have analysed the influence of factors such as distance and height on the accuracy of face recognition algorithms using UAVs [15]. In [16] face recognition and distance estimation are performed from UAV-captured data using a Siamese network. Different architectures can be used for face recognition on UAVs as analysed in [17], that compares ResNet-50 and SENet.

It is important to clarify the distinction between two key terms: face recognition and face verification [18]. The fundamental difference is that face recognition needs a database because it aims to tell which person a face belongs to. On the other hand, face verification just compares two faces and decides whether the face belongs to the same person or to a different one. In this research paper, only face verification is going to be analysed as two faces will be compared: one is the reference or identity face, and the other is the face of an unidentified person.

Table I shows an in-depth comparison among different face verification algorithms available in the literature. It has been expanded from our previous preliminary work [8] to add more parameters and up-to-date algorithms. The verification performance and the different parameters of each algorithm have been obtained from its official paper. In the table, different parameters are compared, such as the input and output size of the neural network, the number of images used for training and the size of the weights file. This last parameter is useful in case the algorithm wants to be embedded in a companion computer in the UAV, as large models are not going to be able to be embedded. Moreover, the verification accuracy is compared using the most common verification datasets for face recognition and verification: Labeled Faces in the Wild (LFW) [19] and YouTube Faces (YTF) [20]. The evaluation in the LFW is conducted using 6000 face pairs while in the YTF 5000 face pairs are used.

The first row of the table shows the performance that human beings have on the LFW dataset. The accuracy obtained is 97.53% [21] and it will be our reference for the comparison of the accuracy of the face verification algorithms. Two algorithms do not surpass this accuracy: Deep-Face [22] with 97.35% and OpenFace [23] with 92.92%.

The algorithms that achieve the highest accuracy in the LFW dataset are ArcFace [27], AdaFace [32] and CurricularFace [34] with 99.83%, 99.82% and 99.8% respectively. Regarding the YTF dataset, ArcFace is the one which achieves the highest accuracy. It is important to mention that not all algorithms have been evaluated on the YTF dataset. For instance, we have no information for AdaFace, CurricularFace or FaceNet512 [29] algorithms.

Moreover, not all the algorithms report the output size of the neural network, which will be the size of the obtained feature vector. The algorithm with the biggest length is Deep-Face with 4096 while the smallest are FaceNet [28] and OpenFace with 128. Regarding the size of the weights file, the biggest ones are AdaFace, VGG-Face [24] and Deep-Face making them large models. The smallest are DeepID2 [26], OpenFace and Dlib [30], making them light models, and easily embedded.

SphereFace [31] is an algorithm that has not been trained with many images (only 0.5 million) but achieves high accuracy on the LFW dataset (99.42%). CosFace [25] achieves high accuracy on the LFW and YTF datasets. However, it has not been selected due to the lack of information in the available literature. DAM-R [33] is a novel algorithm that achieves high accuracy on the LFW dataset but underperforms in the YTF dataset compared with the other algorithms.

Finally, FaceNet and FaceNet512 algorithms were trained with the highest number of images (200 million), followed by AdaFace with 15.1 million. The difference between both FaceNet algorithms is that FaceNet512 [29] is an extended version of FaceNet [28] that has a 512 vector as output instead of 128, increasing the verification performance as can be seen in the table.

In our previous work [8], three algorithms were chosen to perform an analysis on UAV-based use cases: ArcFace [27], VGG-Face [24] and FaceNet512 [29]. In this paper, our research has been expanded by incorporating two additional algorithms present in Table I: OpenFace [23] and Dlib [30].

TABLE I
COMPARISON BETWEEN STATE-OF-THE-ART FACE VERIFICATION ALGORITHMS

| Ref | Algorithm | Input Size | Output Size | Verification Accuracy on LFW dataset [19] | Verification Accuracy on YTF dataset [20] | Weights size | Training Images |
|---|---|---|---|---|---|---|---|
| [21] | **Human-beings** | N/A | N/A | 97.53% | NG | N/A | N/A |
| [22] | **Deep-Face** | 152x152x3 | 4096 | 97.35% | 91.40% | 551 MB | 4.4M |
| [23] | **OpenFace** | 96x96x3 | 128 | 92.92% | NG | 14 MB | 0.5M |
| [24] | **VGG-Face** | 224x224x3 | 2622 | 98.95% | 97.30% | 554 MB | 2.6M |
| [25] | **CosFace** | 112x96x3 | NG | 99.73% | 97.60% | 214 MB | 5M |
| [26] | **DeepID2** | 55x47x3 | 160 | 99.53% | 93.20% | 1.6 MB | 0.2M |
| [27] | **ArcFace** | 112x112x3 | 512 | 99.83% | 98.02% | 131 MB | 5.8M |
| [28] | **FaceNet** | 160x160x3 | 128 | 98.87% | 95.12% | 88 MB | 200M |
| [29] | **FaceNet512** | 160x160x3 | 512 | 99.60% | NG | 91 MB | 200M |
| [30] | **Dlib** | 150x150x3 | 128 | 99.38% | NG | 22 MB | 3M |
| [31] | **SphereFace** | 112x96x3 | 512 | 99.42% | 95.00% | 69 MB | 0.5M |
| [32] | **AdaFace** | 112x112x3 | NG | 99.82% | NG | 668 MB | 15.1M |
| [33] | **DAM-R** | 112x112x3 | NG | 99.03% | 95.23% | NG | 1.5M |
| [34] | **CurricularFace** | 112x112x3 | NG | 99.80% | NG | 249 MB | 6.3M |

NG = Not Given; N/A = Not applicable

The inclusion of these two algorithms was driven by various considerations. OpenFace, while not delivering the highest accuracy, distinguishes itself as a lightweight and fast algorithm. Moreover, Dlib has a high accuracy, surpassing human performance, while also being a lightweight model. By introducing these two lightweight algorithms, we aim to facilitate a comprehensive comparative analysis, allowing us to contrast them with algorithms known for their higher accuracy but slower inference speeds.

Furthermore, many metrics can be used to calculate the similarity indexes between two faces, such as Euclidean distance [35], Manhattan distance [36], and cosine distance [37]. This paper, as opposed to the previous work [8], is going to use two different metrics: cosine distance and Euclidean distance. These are two of the most used metrics for distance calculation in the literature.

Cosine distance [37] has two vectors as inputs and the result will be a number that signifies the similarity index between the vectors. It is calculated as shown in Equation 1:

$$CD(\vec{x},\vec{y}) = 1 - \frac{\sum_1^n x_i y_i}{\sqrt{\sum_1^n x_i^2}\sqrt{\sum_1^n y_i^2}} \quad (1)$$

Here, $\vec{x}$ and $\vec{y}$ are the feature vectors of two different faces. The result of the calculation is the similarity index between them, which will be within the range of 0 and 2. A similarity index of 0 indicates that the faces are exactly alike while if the result is 2 means that the two faces are opposite.

Moreover, we have Euclidean distance [35]. This metric also takes two feature vectors as inputs. The results will be the similarity index between them. The Euclidean distance can be calculated as shown in Equation 2:

$$ED(\vec{x},\vec{y}) = \sqrt{\sum_1^n (x_i - y_i)^2} \quad (2)$$

Once again, $\vec{x}$ and $\vec{y}$ represent the feature vectors of two different faces. Greater similarity indexes mean that the faces are less similar whereas small values signify that the faces are more alike.

## III. DESIGN AND IMPLEMENTATION OF THE PROPOSED SYSTEM

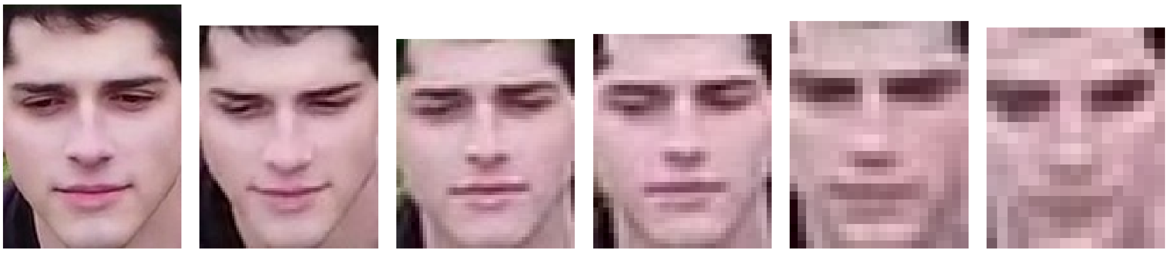
### A. UAV-UWS dataset

In our previous preliminary work [8], a dataset was introduced, known as the UAV-UWS dataset. It was a UAV-recorded dataset at four fixed distances from the volunteers: 5, 7, 10 and 15 metres. This research extends this dataset by incorporating two additional distances: 2 and 20 metres, thereby significantly enlarging the face verification range from 5 to 15 meters to 2 to 20 meters. This addition improves the dataset's overall comprehensiveness.

The 2-metre distance helps evaluate how well the algorithms perform at very close ranges, which is the closest safe distance of operating a UAV in front of volunteers. Furthermore, the addition of the 20-metre distance allows us to assess the performance of the face verification algorithms at a greater distance and identify where their accuracy decreases significantly, making them ineffective. The extension of the UAV-UWS dataset enables a more thorough evaluation of the algorithms to the greatest extent for the UAV platform used.

Therefore, this UAV-UWS dataset has been created and expanded by recordings at six different distances: 2, 5, 7, 10, 15 and 20 metres. These videos were recorded using a DJI Mini 2 UAV with 4K resolution (3840x2160 px) at a frame rate of 30 frames per second (FPS). The dataset includes videos of 20 volunteers representing diverse age groups, genders and ethnicities, resulting in a more inclusive and diverse dataset.

The recording angle was set at 30 degrees to ensure that the volunteers did not need to strain their necks to face the UAV directly. Table II shows the vertical and horizontal distance from the face to the UAV for each of the recording distances in our dataset. Each video has a duration of 30 seconds, during which volunteers were asked to perform a range of head movements, including facing down, up, left and right, making head circles, and staring directly at the UAV. This allows us to have a comprehensive range of the features of each face and analyse how the algorithms perform when volunteers are not facing the UAV directly.



(a) 2 m (b) 5 m (c) 7 m (d) 10 m (e) 15 m (f) 20 m

Fig. 1. Example of a cropped face at the six distances recorded in the dataset

TABLE II
VERTICAL, HORIZONTAL AND DIRECT DISTANCE FROM THE UAV TO THE FACE OF THE VOLUNTEERS WITH A RECORDING ANGLE OF 30 DEGREES.

| Direct distance | Horizontal distance | Vertical distance |
|---|---|---|
| 2 m | 1.7 m | 1.0 m |
| 5 m | 4.3 m | 2.5 m |
| 7 m | 6.0 m | 3.5 m |
| 10 m | 8.7 m | 5.0 m |
| 15 m | 13 m | 7.5 m |
| 20 m | 17 m | 10 m |

TABLE III
AVERAGE FACE SIZE IN PIXELS AT DIFFERENT DISTANCES FROM THE CAMERA OF THE UAV.

| Distance | Size Face |
|---|---|
| 2 m | 125x170 px |
| 5 m | 80x100 px |
| 7 m | 50x60 px |
| 10 m | 40x45 px |
| 15 m | 25x30 px |
| 20 m | 20x25 px |

Furthermore, the dataset also contains a close image of each volunteer taken with a smartphone. This will be the identity face that will be compared with the ones from the UAV videos. Fig 1 shows one example of a cropped face from the UAV-UWS dataset at each distance: 2, 5, 7, 10, 15 and 20 metres. Moreover, Table III shows the relation between the average face size in pixels versus the distance from the camera of the UAV. These values have been obtained using our dataset and therefore images with 4K resolution (3840x2160 px).

The UAV-UWS dataset will not be publicly released due to GDPR (General Data Protection Regulation) considerations as it contains personal information for the volunteers.

### *B. Design of the face verification pipeline*

One of the primary contributions of this study lies in the conceptualisation and realisation of a pipeline for executing face verification and obtaining empirical results, as depicted in Figure 2. The pipeline operates on a pair of input images. The first image is extracted from a video frame captured by a UAV within our proprietary dataset. The second input image corresponds to the facial image of the individual undergoing verification, captured at close proximity using a mobile device. The ultimate outcome of this pipeline is a similarity index quantifying the likeness between the target individual's face and the face within the video frame. The pipeline comprises four distinct stages:

1) Face Detection: This initial stage of our pipeline employs the RetinaFace algorithm, chosen for its superior accuracy in long-distance face detection. Although RetinaFace might not be the fastest face detection algorithm available, its speed is not a significant concern for our research's objectives, which focus solely on face verification algorithms. RetinaFace takes an image as input and yields the coordinates of all detected faces within the image, along with the associated detection accuracy.
2) Preprocessing: This stage is divided into two additional steps. The first one involves cropping the face from the image using the coordinates provided by RetinaFace. Then, the face is resized to match the input dimensions required by the specific face verification algorithm employed. These input dimensions vary depending on the chosen algorithm as shown in Table I. Since the aspect ratio of the original face may differ from the input size, padding is applied to the image. Black pixels are added to the sides of the image to achieve the expected input dimensions. This approach ensures that the face is not distorted during the resizing process.
3) Siamese network [22] [38]: In the third stage of the pipeline a Siamese network is used. It is a frequently adopted configuration in the literature for face verification tasks. It comprises two identical Convolutional Neural Networks (CNN) with identical backbones and weights. When presented with the same input, these CNNs produce identical outputs. Each CNN takes as input a face resized to the required input dimensions. The output of these networks is a feature vector, whose dimensions vary based on the chosen algorithm. This vector represents the distinctive characteristics of the processed face.
4) Similarity index calculation: The final stage of the pipeline involves calculating the similarity index between the two feature vectors obtained from the Siamese network. It is important to note that this stage diverges from our prior conference paper's approach as outlined in [8]. In that research, the sole metric employed for index calculation was the cosine distance. However, in this paper, the flexibility of our approach has been expanded to allow the utilisation of various distance calculation metrics, such as Euclidean distance, Manhattan distance and cosine distance. The optimal distance calculation method varies depending on the face verification algo-

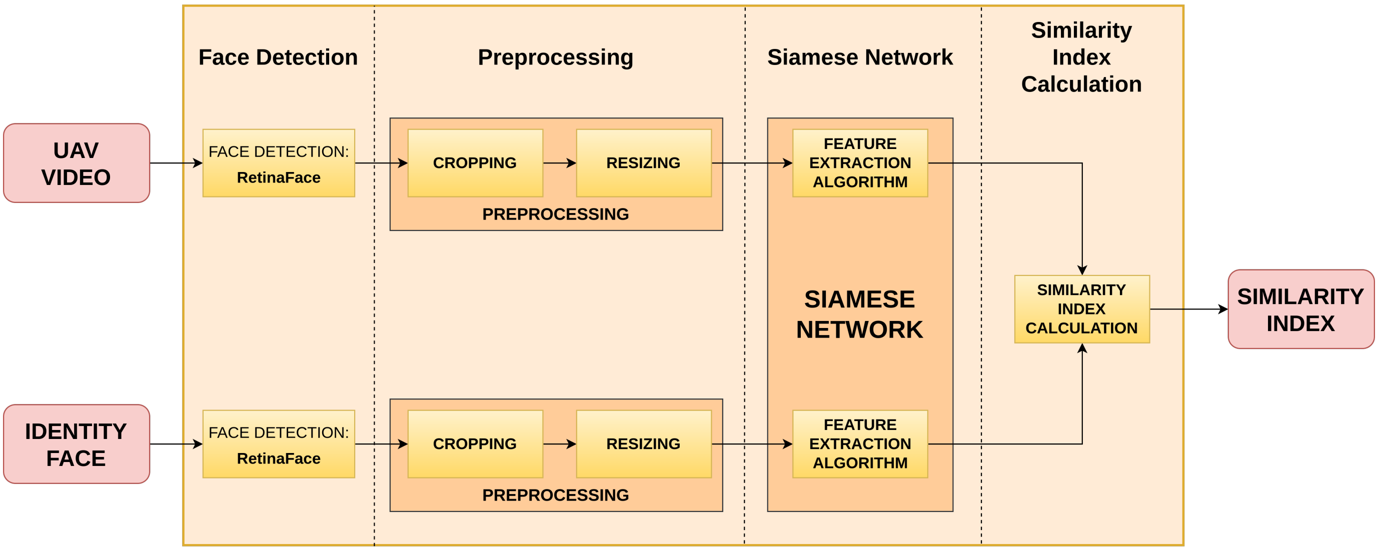



Fig. 2. Enhanced pipeline used to obtain the results from the experiments. It is divided into four stages: face detection, preprocessing, Siamese network, and similarity index calculation.

rithm used.

Other authors have created similar approaches for a face verification pipeline but without further explanation about the preprocessing stage or having a fixed similarity index metric [39]. Our pipeline allows swift changes in the similarity index metrics depending on the algorithm used. For example, we can switch between cosine distance, Euclidean distance, Manhattan distance, etc. Furthermore, our pipeline allows us to introduce other stages in the preprocessing in case further improvements to the images are needed, such as face alignment. Finally, the Siamese network allows faster execution of the pipeline as the identity features only need to be extracted in the first execution, as it will not change.

### C. Implementation

The results presented in this study were obtained using a unified framework, where all five verification algorithms are implemented. The framework is denoted as DeepFace [40] [41] and includes the five face verification algorithms used. It also includes the face detection algorithm: RetinaFace and the two metrics used in this paper: cosine distance and Euclidean distance. Evaluating all the algorithms on a uniform platform ensures reliable and easily comparable results given that they were acquired under identical conditions.

DeepFace uses the Keras library [42] in a Python 3.8 environment. Additionally, the OpenCV Python library was employed for the image preprocessing. The computational environment for the implementation was a computer running Focal Ubuntu version 20.04.3.

The pipeline was meticulously executed employing all the five verification algorithms and evaluated with a common testbed. For the purposes of this comparative analysis, the algorithms were executed on a high-performance GPU. This setup simulates a scenario where the video feed from the UAV is transmitted to a computer for the execution of the pipeline.

## IV. Empirical Results and Discussion

### A. Testbed Description

The experiments were conducted on an NVIDIA GeForce GTX TITAN X with 12GB of onboard memory. These experiments were executed 10 times to obtain the final results. The UAV used to record the dataset is a DJI Mini 2 with a 4K (3840x2160 px) camera and the videos have been recorded at 30 FPS.

It is noted that substantially more experiments have been conducted to produce a significantly larger number of comparative results, in contrast to the results reported in our preliminary work [8].

### B. Comparison of Different Face Verification Algorithms

*1) Build Time and Weights Size:* Table IV shows the model's build time and the size of the weights for each algorithm. As can be seen, FaceNet512 is the one with the highest building time, while VGG-Face has the lowest. Moreover, VGG-Face is the one with the largest weights file. The size of the weights file is important, for example, if the algorithm is going to be embedded in a system without a lot of memory onboard. In that case, the smaller the size of the file, the better. In addition, the build time is also significant as it shows us how long it is going to take the algorithm to initiate the verification process.

TABLE IV
Comparison between Build Time and Weights of each Model

| Algorithm | Build Time | Weights |
|---|---|---|
| ArcFace | 1001 ms | 131 MB |
| OpenFace | 828 ms | 15 MB |
| FaceNet512 | 2130 ms | 91 MB |
| Dlib | 144 ms | 22 MB |
| VGGFace | 758 ms | 554 MB |

*2) Inference Time:* It is defined as the time it takes for one frame to complete a full pipeline execution. The inference time can be divided into four stages, the same as the processing pipeline. This allows for a comparison of the inference time of only the face verification algorithms, rather than the entire pipeline.

1) Face Detection: Using the RetinaFace algorithm in the pipeline, the face detection process takes approximately 165 ms. However, by employing a faster face detection algorithm, the pipeline's inference time can be reduced significantly. It is worth noting that this stage is the slowest of the pipeline.
2) Preprocessing: This time is independent of the algorithms used. This stage of the pipeline only takes around 0.3 ms, making it the fastest of the four stages.
3) Siamese Network: Only the time required for extracting the facial features from a face from the video is considered. This has been done because the features of the identity face are initially extracted at the beginning of the process and remain constant throughout. This eliminates the need for recalculations to speed up the process. The time taken for this stage varies depending on the face verification algorithm used, as shown in Figure 3.
4) Similarity index calculation: The last stage is also independent of the algorithms used. The inference time is approximately the same for both metrics: cosine and Euclidean distance. It takes around 1 ms.

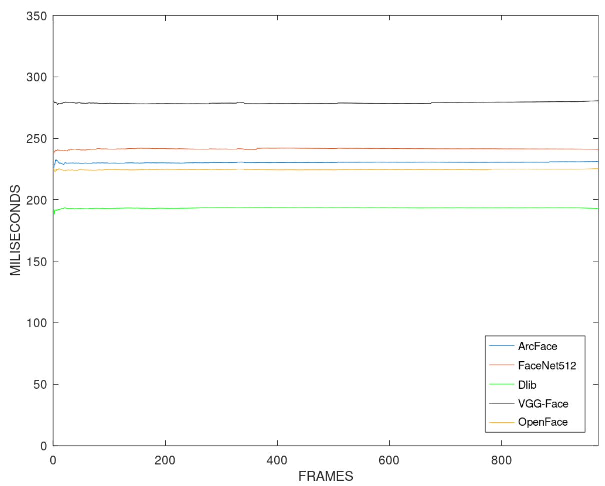



Fig. 3. Cumulative average of the inference time (milliseconds) per frame for each algorithm

All the stages have fixed times for the five face verification algorithms used, except for the Siamese network stage. The cumulative average inference time for the five face verification algorithms is shown in Figure 3, with improved visualisation compared with our preliminary work [8]. As a result, it is possible to see the evolution of the average inference time during a whole video, frame by frame.

The variations in the inference time depicted in Figure 3 are exclusively due to which face verification algorithm is used. VGG-Face is clearly the slowest algorithm, while Dlib is the fastest. FaceNet512, ArcFace and OpenFace have similar inference times, with OpenFace being the fastest of the three.

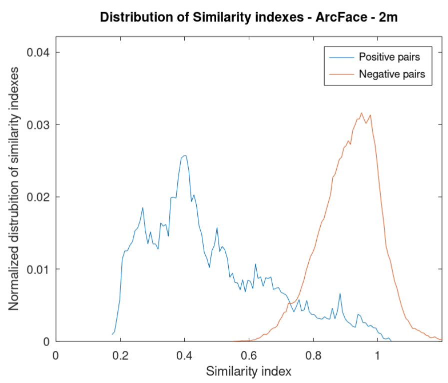
It is worth noting that all the stages excluding the Siamese network have approximately a constant inference time of 167 ms. Therefore, the inference times of only the Siamese network for the face verification algorithms are VGG-Face 114 ms, FaceNet512 74 ms, ArcFace 64 ms, OpenFace 58 ms and Dlib 25 ms, respectively.

*3) Similarity indexes distribution:* The similarity indexes were obtained for each of the five face verification algorithms at the six distances of our UAV-UWS dataset. They were obtained using the proposed pipeline designed and explained in Section III. They are divided into two different types: Positive pairs and negative pairs. The first ones were obtained by having the one that contains the same persons of the identity faces as the video input, i.e., we compare the image of the face of one person with a different image containing the face of the same person. On the other hand, the negative pairs were obtained using a video input that contains different people from the identity face, i.e., we compare the image of the face of one person with the image of another person. The negative pairs similarity indexes should be high, as two different people are compared and their similarity should be low. On the other hand, the positive pair similarity indexes should be low as two faces from the same person are compared.

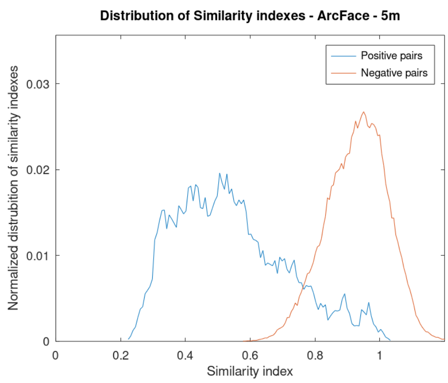
In the graphs, the most important aspect to compare is the overlapping between the two plots. If the positive and negative pairs plots are highly overlapped, it is going to be extremely difficult to define an optimal threshold, therefore the accuracy will be low, with a high number of false positives and negatives. On the other hand, if both plots are further apart, a threshold will be easy to define and the algorithm will have high accuracy.

As can be seen in all the graphs, the further the distance, the positive pairs plots shift more to the right. This occurs because as the UAV moves further from the person, the face is going to have lower resolution, so it is more difficult to appreciate the features of the face and therefore the similarity index is higher.

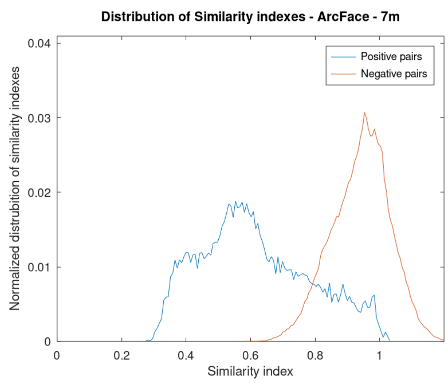
Firstly, let us focus on the ArcFace graphs (Figure 4). At 2 (Figure 4a) and 5 metres (Figure 4b), it can be seen that both plots are well separated; therefore, it can be easy to define an optimal threshold. The specific position of the threshold can vary depending on the use case. If it is a high-security one with no false positives allowed, 0.6 will be an optimal value. If it is a more permissive use case, the threshold could be defined at around 0.75. At 7 metres (Figure 4c), the positive pairs plot starts shifting to the right and starts to overlap with the negative pairs one, yet still, a high accuracy can be obtained. At 10 metres (Figure 4d), a high quantity of positive pairs similarity indexes are completely overlapped with the negative pairs having a peak at around 0.9. The accuracy is not going to be very high, yet still, some good identifications can be obtained. At 15 metres (Figure 4e), both plots are almost completely overlapped except for some values between 0.4 and 0.7, and the accuracy will be very low. Finally, at 20 metres (Figure 4f), both plots are completely overlapped and it will not be possible to differentiate between two faces. Therefore,



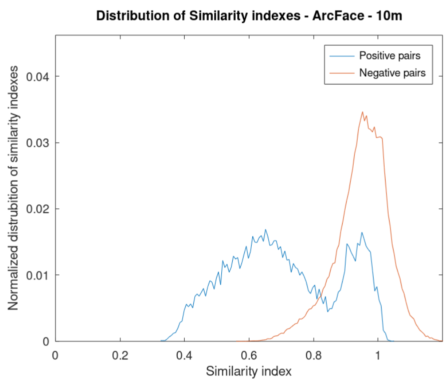

(a) 2 metres



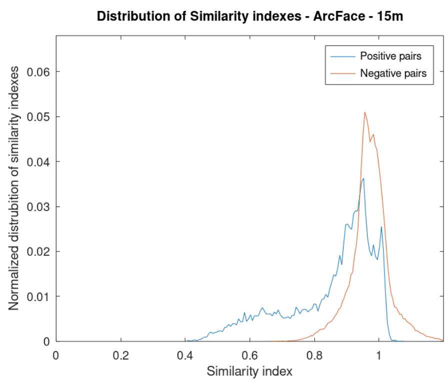

(b) 5 metres



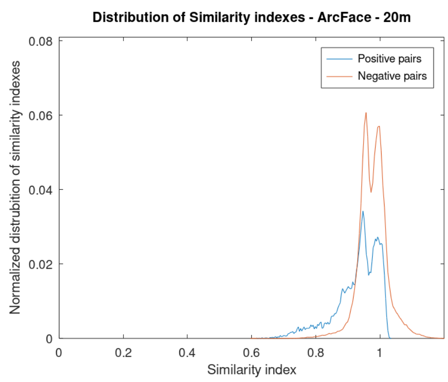

(c) 7 metres




(d) 10 metres

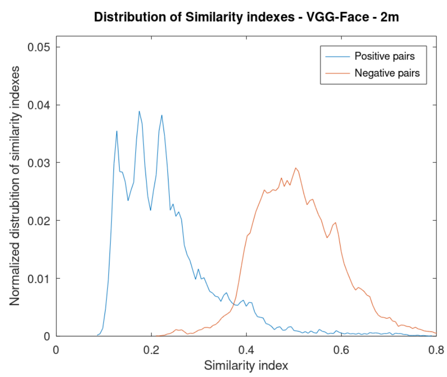



(e) 15 metres

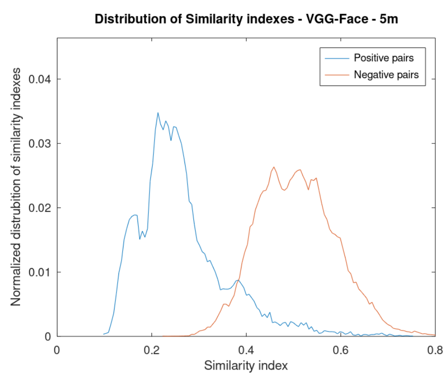



(f) 20 metres

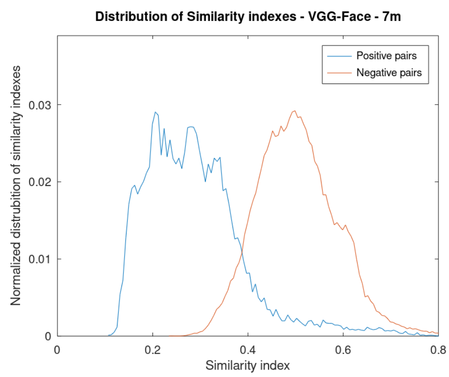
Fig. 4. Similarity indexes of the ArcFace face verification algorithm for 2, 5, 7, 10,15 and 20 metres of distance



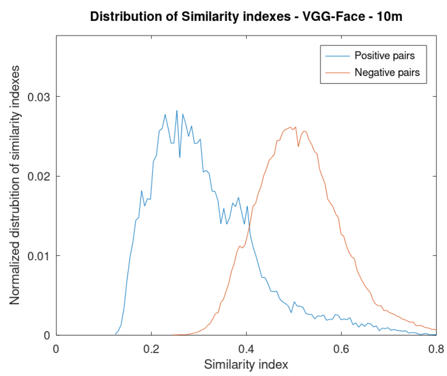

(a) 2 metres



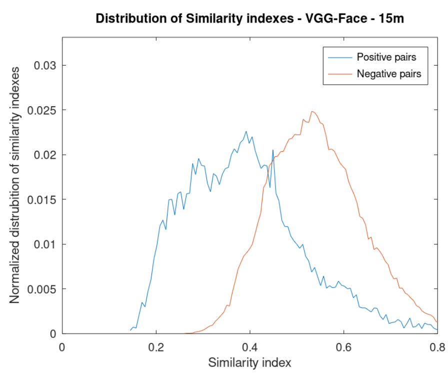

(b) 5 metres



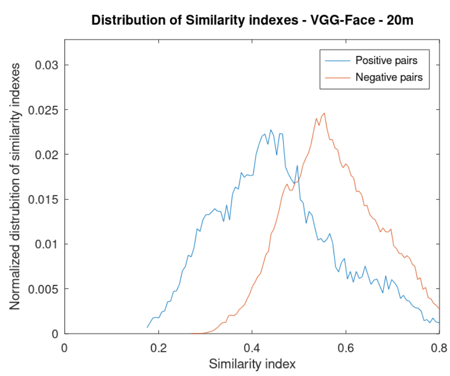

(c) 7 metres




(d) 10 metres

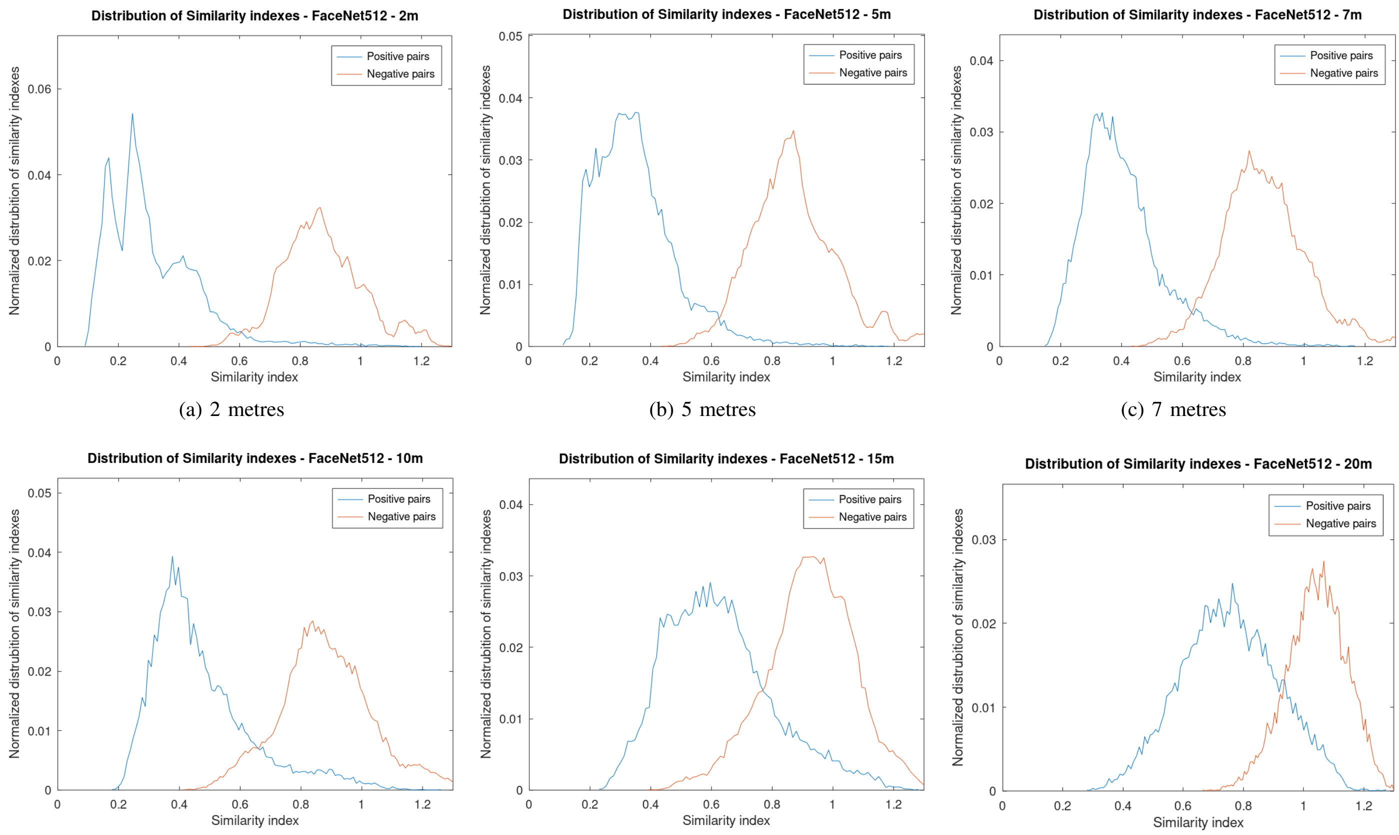



(e) 15 metres




(f) 20 metres

Fig. 5. Similarity indexes of the VGG-Face face verification algorithm for 2, 5, 7, 10, 15 and 20 metres of distance




(a) 2 metres (b) 5 metres (c) 7 metres

(d) 10 metres (e) 15 metres (f) 20 metres

Fig. 6. Similarity indexes of the FaceNet-512 face verification algorithm for 2, 5, 7, 10, 15 and 20 metres of distance

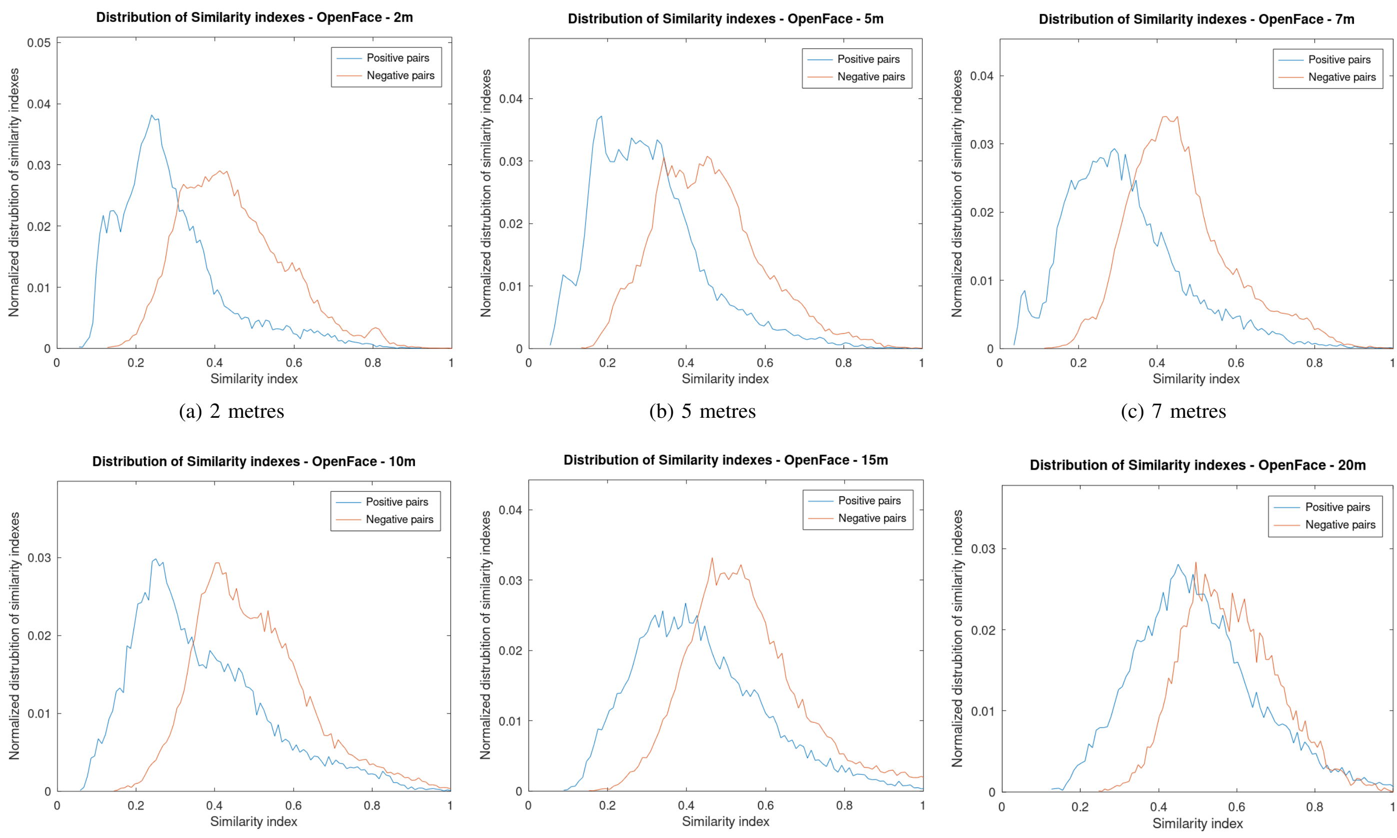
ArcFace will not be able to verify at distances higher than 15 metres.

Secondly, VGG-Face (Figure 5) has good results at the first three distances: 2 (Figure 5a), 5 (Figure 5b) and 7 metres (Figure 5c). Both plots are well separated in the three graphs. The peak of the negative pairs is around 0.5, while the positive pairs is at 0.2. At 10 metres (Figure 5d), the plots start to become more overlapped but the peak of positive pairs is still not overlapped, so a high accuracy can still be obtained. At 15 metres (Figure 5e), both plots are overlapped; however, still a considerable number of positive pairs similarity indexes are out of the overlapping, and thus reasonable accuracy can still be obtained. Finally, although the plots are highly overlapped at 20 metres (Figure 5f), VGG-Face will verify the face at this distance with low accuracy, as some similarity indexes are still not overlapped.

Thirdly, FaceNet512 is the algorithm that has the best performance as seen in Figure 6. All the graphs are well separated so it will be easy to define an optimal threshold and verify correctly most users. At 2 metres (Figure 6a), both plots are almost completely separated so the accuracy will be very high with almost no false negatives or positives. At 5 (Figure 6b), 7 (Figure 6c) and 10 metres (Figure 6d), the peak of the positive pairs similarity index is not overlapped so the accuracy will still be very high. At 15 (Figure 6e) and 20 metres (Figure 6f), the plots are more overlapped, although it is still possible to differentiate them; thus a decent threshold can be defined in order to maintain a good accuracy. FaceNet512 achieves very good results being able to verify at all distances using cosine distance as the metric.

Fourthly, let us scrutinise OpenFace (Figure 7). Even at 2 metres (Figure 7a), both plots are overlapped, although the main peak of the positive pairs is not, and thus it will have reasonable accuracy. At 5 metres (Figure 7b), the curves are highly overlapped, and thus it will be difficult to verify correctly unless the conditions are optimal. At 7 (Figure 7c), 10 (Figure 7d) and 15 metres (Figure 7e), the same occurs respectively. The curves are almost completely overlapped, and hence there is a high probability of having false positives while verifying a person. Moreover, the choice of a threshold will be difficult and false positives will have to be assumed. At 20 metres (Figure 7f), both plots are totally overlapped and it will not be possible to verify anybody correctly. It can be seen that OpenFace, as it is a lightweight model, will only have good accuracy at really close distances, and thus it will not be suitable for UAVs.

Fifthly and finally, Dlib is a fast algorithm as can be seen in Figure 8 although it does not have a good performance. At 2 metres (Figure 8a), most of the positive pairs plot is overlapped. Meanwhile, the main peak is below 0.05 (beginning of negative pairs plot), and thus some good results may be achievable. At 5 (Figure 8b), 7 (Figure 8c) and 10 metres (Figure 8), the positive pairs plot starts to shift to the right, becoming highly overlapped with the negative pairs plot. In




(a) 2 metres (b) 5 metres (c) 7 metres

(d) 10 metres (e) 15 metres (f) 20 metres

Fig. 7. Similarity indexes of the OpenFace face verification algorithm for 2, 5, 7, 10, 15 and 20 metres of distance

this situation, a high number of false positives and negatives will be obtained; therefore, the accuracy is going to be low. At 15 metres (Figure 8e), both graphs are almost completely overlapped and as both start at around 0.05, it is not possible to have true positives without obtaining also false positives. The accuracy at this distance will be very low. At 20 metres (Figure 8f), both plots are completely overlapped; therefore, the accuracy is going to be extremely low. The algorithm at this distance is not useful. Hence, despite Dlib being the fastest algorithm, the accuracy is low; consequently, it will not be useful for UAV use cases.

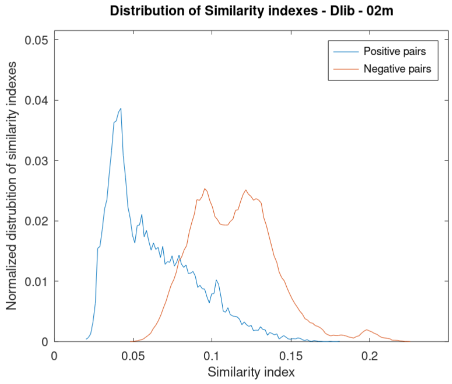
For further comparison, we have obtained the graphs for OpenFace at three different distances but using Euclidean distance as the metric to obtain the similarity indexes. As it can be seen in Figure 9, the graphs are mostly similar to the ones obtained using the cosine distance (Figure 7). OpenFace will perform similarly using cosine distance or Euclidean distance. Therefore, the use of one distance or the other will only vary the accuracy slightly. Hence, the recommendation of one distance calculator will depend on the algorithm and the specific use case. Further analysis will be needed when an algorithm has been selected to choose the optimal distance calculator. This can be obtained using the enhanced pipeline that admits any distance calculator, such as the previously mentioned cosine or Euclidean distance, but also Manhattan or Minkowski distance [43].

### C. Discussion

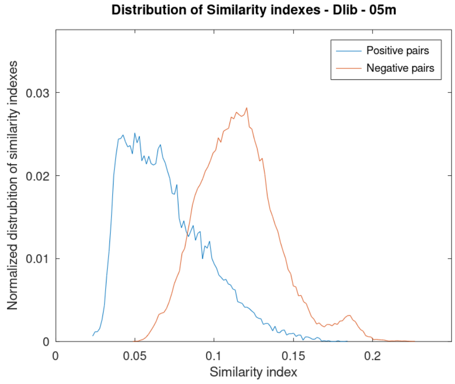
Based on the results obtained, ArcFace achieves good results only up to 10 metres. VGG-Face is going to be a useful algorithm for UAV use cases up to 15 metres, whilst as the distance increases, the accuracy will be reduced significantly. FaceNet512 is the best algorithm of all. It is going to have good accuracy at every distance concerned in this study, making it a highly useful algorithm for face verification from UAVs. Finally, although OpenFace and Dlib are fast algorithms, the accuracy they achieve is very low, and they will only be useful at close distances (less than 2 metres). Therefore, they are not suitable for face verification at long distances. Table V shows a summary of the results obtained for each algorithm. The first column shows how fast each of the algorithms is. The second column shows which is the highest distance where the algorithms can verify correctly with reasonable accuracy. Finally, the last column shows the overall recommendation for the use of the algorithms in UAV-based use cases.

## V. Concluding Remarks

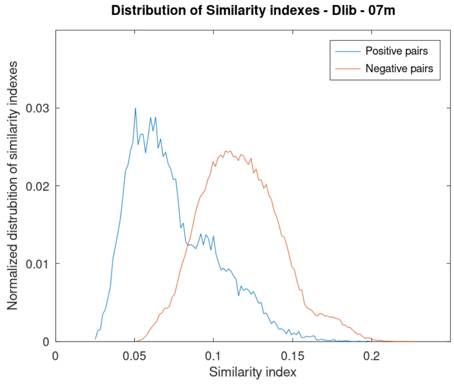
In this paper, we have conducted a comprehensive comparison among five state-of-the-art face verification algorithms with a proposed pipeline for executing these algorithms and obtaining empirical results. In order to conduct a more inclusive analysis, the UAV-UWS dataset has been expanded to add two new distances (2 and 20 metres) to have six in



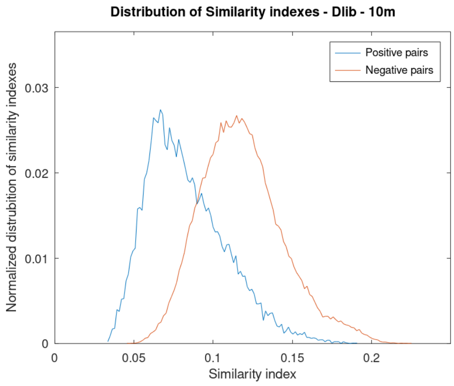

(a) 2 metres



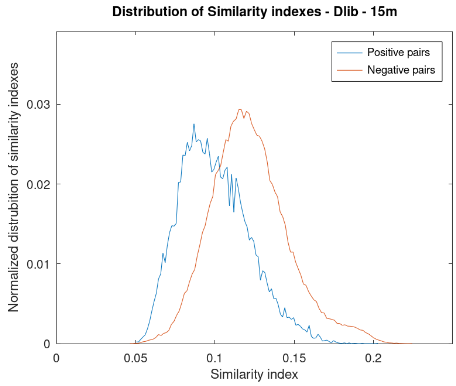

(b) 5 metres



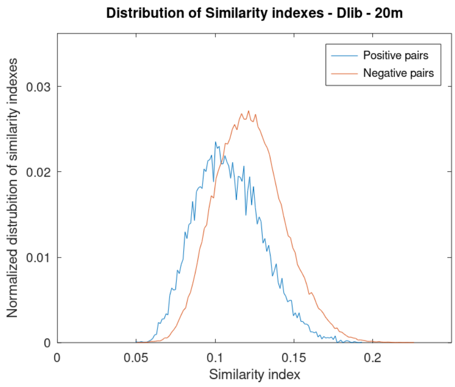

(c) 7 metres




(d) 10 metres

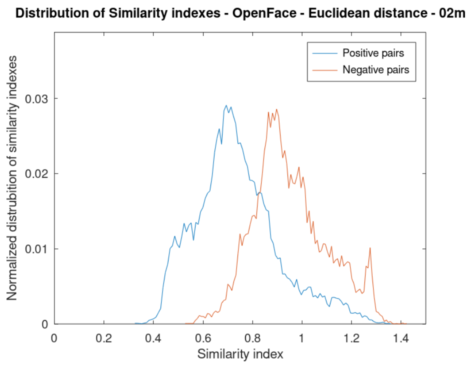



(e) 15 metres

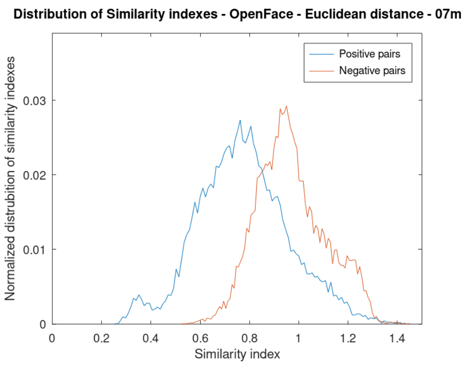



(f) 20 metres

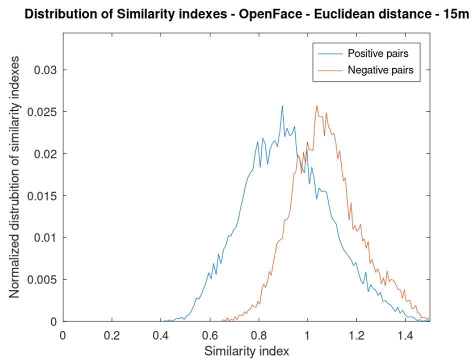
Fig. 8. Similarity indexes of the Dlib face verification algorithm for 2, 5, 7, 10, 15 and 20 metres of distance




(a) 2 metres




(b) 7 metres




(c) 15 metres

Fig. 9. Similarity indexes of the Dlib face verification algorithm for 2, 7 and 15 metres of distance using Euclidean distance as metric

TABLE V
RESULTS COMPARISON FOR EACH FACE VERIFICATION ALGORITHM

| Algorithm | Inference time | Maximum verification distance | Overall recommendation for UAVs |
|---|---|---|---|
| ArcFace | Medium | 10 metres | Recomended |
| OpenFace | Fast | 2 metres | Not recommended |
| FaceNet512 | Slow | 20 metres | Highly recommended |
| Dlib | Very fast | 2 metres | Not recommended |
| VGGFace | Very Slow | 10 metres | Not recommended |

total, compared with the previous preliminary study where four distances (5, 7, 10 and 15 metres) were investigated. The algorithms have been compared based on different metrics such as their inference time, similarity indexes distribution, building time and the size of the weights file.

Our findings indicate that FaceNet512 emerges as the algorithm that will have the best accuracy in UAV use cases, as demonstrated by the results obtained from the experiments. While VGG-Face and ArcFace will also have good accuracy, they perform less optimally at long distances (more than 10 meters). Notably, VGG-Face is the slowest algorithm among the concerned algorithms. OpenFace and Dlib are the fastest algorithms but exhibit reduced accuracy when used with UAVs, and will only have good performance at really close distances (less than 2 meters). As a result, our recommendation for UAV face verification applications leans towards FaceNet512 and ArcFace. FaceNet512 will have better accuracy than ArcFace at long distances although it is a slower algorithm. Therefore, the choice of one algorithm or the other will depend on the requirements of the specific use case.

For future work, one of the previously recommended algorithms will be chosen to perform face verification from UAVs. The pipeline will be enhanced for that specific algorithm, and an appropriate threshold will be defined. The video feed from the UAV will be transmitted via a wireless network, such as 4G or 5G, and visualised alongside the face verification results in a Ground Control Station.

Moreover, in follow-up publications, the accuracy of each of the algorithms can be obtained using different calculations to define the optimal thresholds, such as maximising the accuracy, the F1-score or defining a FAR limit.

## ACKNOWLEDGMENT

This work was in part funded by the EU Horizon 2020 ARCADIAN-IoT project (“Autonomous Trust, Security and Privacy Management Framework for IoT”) under the Grant number H2020-SU-DS-2020/101020259 and the EU Horizon 2020 5G-PPP 5G-INDUCE project (“Open cooperative 5G experimentation platforms for the industrial sector NetApps”) under the Grant number H2020-ICT-2020-2/101016941. The authors would like to thank all the partners in these projects for their support.

## REFERENCES

[1] Z. Zaheer, A. Usmani, E. Khan, and M. A. Qadeer, “Aerial surveillance system using uav,” in *2016 Thirteenth International Conference on Wireless and Optical Communications Networks (WOCN)*, 2016, pp. 1–7.

[2] E. Semsch, M. Jakob, D. Pavlicek, and M. Pechoucek, “Autonomous uav surveillance in complex urban environments,” in *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, vol. 2. IEEE, 2009, pp. 82–85.

[3] I. Martinez-Alpiste, G. Golcarenarenji, Q. Wang, and J. M. Alcaraz-Calero, “Search and rescue operation using uavs: A case study,” *Expert Systems with Applications*, vol. 178, p. 114937, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095741742100378X

[4] R. Ullah, H. Hayat, A. A. Siddiqui, U. A. Siddiqui, J. Khan, F. Ullah, S. Hassan, L. Hasan, W. Albattah, M. Islam *et al.*, “A real-time framework for human face detection and recognition in cctv images,” *Mathematical Problems in Engineering*, vol. 2022, 2022.

[5] EU H2020 ARCADIAN-IoT project, “Autonomous trust, security and privacy management framework for IoT.”

[6] EU H2020 5G-INDUCE project, “Open cooperative 5G experimentation platforms for the industrial sector netapps.”

[7] J. Diez-Tomillo, J. M. Alcaraz-Calero, and Q. Wang, “Dynamic-Distance-Based Thresholding for UAV-Based Face Verification Algorithms,” *Sensors*, vol. 23, no. 24, 2023. [Online]. Available: https://www.mdpi.com/1424-8220/23/24/9909

[8] ——, “Empirical comparison of face verification algorithms from uavs,” in *2023 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 2023, pp. 1–6.

[9] S. Berrahal, J.-H. Kim, S. Rekhis, N. Boudriga, D. Wilkins, and J. Acevedo, “Border surveillance monitoring using quadcopter uav-aided wireless sensor networks,” *Journal of Communications Software and Systems*, vol. 12, no. 1, pp. 67–82, 2016.

[10] I. Martinez-Alpiste, G. Golcarenarenji, Q. Wang, and J. M. A. Calero, “Altitude-adaptive and cost-effective object recognition in an integrated smartphone and uav system,” in *2020 European conference on networks and communications (EuCNC)*. IEEE, 2020, pp. 316–320.

[11] I. Martinez-Alpiste, G. Golcarenarenji, Q. Wang, and J. M. Alcaraz-Calero, “Real-time low-pixel infrared human detection from unmanned aerial vehicles,” in *Proceedings of the 10th ACM symposium on design and analysis of intelligent vehicular networks and applications*, 2020, pp. 9–15.

[12] A. Saha, A. Kumar, and A. K. Sahu, “Face recognition drone,” in *2018 3rd International Conference for Convergence in Technology (I2CT)*, 2018, pp. 1–5.

[13] “Qgroundcontrol.” [Online]. Available: http://qgroundcontrol.com/

[14] R. Jurevičius, N. Goranin, J. Janulevičius, J. Nugaras, I. Suzdalev, and A. Lapusinskij, “Method for real time face recognition application in unmanned aerial vehicles,” *Aviation*, vol. 23, no. 2, pp. 65–70, 2019.

[15] M. Rostami, A. Farajollahi, and H. Parvin, “Deep learning-based face detection and recognition on drones,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–15, 2022.

[16] Y.-H. Pu, P.-S. Chiu, Y.-S. Tsai, M.-T. Liu, Y.-Z. Hsieh, and S.-S. Lin, “Aerial face recognition and absolute distance estimation using drone and deep learning,” *The Journal of Supercomputing*, vol. 78, no. 4, pp. 5285–5305, 2022.

[17] G. Amato, F. Falchi, C. Gennaro, F. V. Massoli, and C. Vairo, “Multi-resolution face recognition with drones,” in *2020 3rd International Conference on Sensors, Signal and Image Processing*, 2020, pp. 13–18.

[18] K. Jonsson, J. Matas, J. Kittler, and Y. Li, “Learning support vectors for face verification and recognition,” in *Proceedings fourth ieee international conference on automatic face and gesture recognition (cat. no. pr00580)*. IEEE, 2000, pp. 208–213.

[19] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, “Labeled faces in the wild: A database for studying face recognition in unconstrained environments,” University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[20] L. Wolf, T. Hassner, and I. Maoz, “Face recognition in unconstrained videos with matched background similarity,” in *CVPR 2011*, 2011, pp. 529–534.

[21] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, “Attribute and simile classifiers for face verification,” in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 365–372.

[22] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, “Deepface: Closing the gap to human-level performance in face verification,” in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1701–1708.

[23] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, “Openface: A general-purpose face recognition library with mobile applications,” CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016.

[24] O. M. Parkhi, A. Vedaldi, and A. Zisserman, “Deep face recognition,” in *Proceedings of the British Machine Vision Conference (BMVC)*, M. W. J. Xianghua Xie and G. K. L. Tam, Eds. BMVA Press, September 2015, pp. 41.1–41.12. [Online]. Available: https://dx.doi.org/10.5244/C.29.41

[25] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, “Cosface: Large margin cosine loss for deep face recognition,” 2018. [Online]. Available: https://arxiv.org/abs/1801.09414

[26] Y. Sun, X. Wang, and X. Tang, “Deep learning face representation by joint identification-verification,” 2014.

[27] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “Arcface: Additive angular margin loss for deep face recognition,” 2019.

[28] F. Schroff, D. Kalenichenko, and J. Philbin, “Facenet: A unified embedding for face recognition and clustering,” in *Proceedings of the IEEE*

*conference on computer vision and pattern recognition*, 2015, pp. 815–823.
[29] D. Sandberg, "Face recognition using tensorflow," 2018. [Online]. Available: https://github.com/davidsandberg/facenet
[30] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, no. 60, pp. 1755–1758, 2009. [Online]. Available: http://jmlr.org/papers/v10/king09a.html
[31] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, and L. Song, "Sphereface: Deep hypersphere embedding for face recognition," 2017. [Online]. Available: https://arxiv.org/abs/1704.08063
[32] M. Kim, A. K. Jain, and X. Liu, "Adaface: Quality adaptive margin for face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.
[33] J. Liu, Y. Wu, Y. Wu, C. Li, X. Hu, D. Liang, and M. Wang, "Dam: discrepancy alignment metric for face recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3814–3823.
[34] Y. Huang, Y. Wang, Y. Tai, X. Liu, P. Shen, S. Li, J. Li, and F. Huang, "Curricularface: adaptive curriculum learning loss for deep face recognition," in *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 5901–5910.
[35] M. Malkauthekar, "Analysis of euclidean distance and manhattan distance measure in face recognition," in *Third International Conference on Computational Intelligence and Information Technology (CIIT 2013)*. IET, 2013, pp. 503–507.
[36] M. Khalil-Hani and L. S. Sung, "A convolutional neural network approach for face verification," in *2014 International Conference on High Performance Computing & Simulation (HPCS)*. IEEE, 2014, pp. 707–714.
[37] H. V. Nguyen and L. Bai, "Cosine similarity metric learning for face verification," in *Asian conference on computer vision*. Springer, 2010, pp. 709–720.
[38] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1. IEEE, 2005, pp. 539–546.
[39] R. Ranjan, A. Bansal, J. Zheng, H. Xu, J. Gleason, B. Lu, A. Nanduri, J.-C. Chen, C. D. Castillo, and R. Chellappa, "A fast and accurate system for face detection, identification, and verification," *IEEE Transactions on Biometrics, Behavior, and Identity Science*, vol. 1, no. 2, pp. 82–96, 2019.
[40] S. I. Serengil and A. Ozpinar, "Lightface: A hybrid deep face recognition framework," in *2020 Innovations in Intelligent Systems and Applications Conference (ASYU)*. IEEE, 2020, pp. 23–27. [Online]. Available: https://doi.org/10.1109/ASYU50717.2020.9259802
[41] ——, "Hyperextended lightface: A facial attribute analysis framework," in *2021 International Conference on Engineering and Emerging Technologies (ICEET)*. IEEE, 2021, pp. 1–4. [Online]. Available: https://doi.org/10.1109/ICEET53442.2021.9659697
[42] F. Chollet *et al.*, "Keras," 2015. [Online]. Available: https://github.com/fchollet/keras
[43] M. A. El-Sayed, "New similarity measure based on minkowski distance and its applications in face recognition," *Advances in Computer Science and Engineering*, vol. 11, no. 1, p. 1, 2013.

**Julio Diez-Tomillo** is a PhD candidate at School of Engineering and Computing (CEPS) at the University of the West of Scotland, where he is actively involved in the research of artificial intelligence applied to UAVs, focusing on face verification and detection algorithms. Nowadays he is also working on the EU Horizon 2020 project ARCADIAN-IoT: Autonomous Trust, Security and Privacy Management Framework for IoT and in the EU Horizon INCODE project: Programming Platform for Intelligent Collaborative Deployments over Heterogeneous Edge-IoT Environments. He has a BSc Honours and a MSc in Telecommunications Engineering at Universidad de Valladolid in Spain.

**Prof. Jose M. Alcaraz-Calero (M)** is a Professor in Networks at School of Engineering and Computing at the UWS. He is an IEEE Senior Member with experience in 5G networks, network slicing, monitoring, automation and management. In the academic side, he has more than 150 publications in SCI-indexed international journals, conferences and books. He has been involved in 20 editorial activities in the most prestigious journal in the field and served as chair in 20 international flagship conferences and contributes as Technical Programme Committee member in more than 100 international conferences. In the industrial side, he has more than 50 patents and intellectual property rights. From the leadership perspective, Jose M. has significant experience as Principal Investigator or as Co-Investigator in more than 20 research projects at local, national and especially at European and international level. He is the Co-Technical Manager for EU Horizon 2020 projects SELFNET and SliceNet.

**Prof Qi Wang (M)** is a Professor in Next-Generation Smart Networks and Services at UWS. He has served as a Board Member of the EU 5G-PPP Technology Board, Member of several 5G-PPP Working Groups, Member of Scotland's Developing AI and AI Enabled Products and Services Working Group, and the Co-Technical Manager for EU Horizon 2020 projects SELFNET and SliceNet. He is Co-Principal Investigator for EU Horizon 2020 projects 6G BRAINS, 5G INDUCE and ARCADIAN-IoT, and EU Horizon Europe projects RIGOUROUS, INCODE and 6G PATH. He has published about 200 papers in 5G, 6G, AI and related areas. He was a Best Paper Award Winner of IEEE IEEE ICCE2012, ICCE 2014, SIGMAP 2014, SOFTNETWORKING 2017 and PETRA 2023. He was a Winner of 2018 Scottish Knowledge Exchange Award, Winner of Scotland Centre for Engineering Education and Development (CeeD) Industries Awards - Innovation Award 2020, and Winner of UK Times Higher Education (THE) Awards 2020 -Knowledge Exchange/Transfer Initiative of the Year Award.