PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Facial feature tracking method using a hybrid model of the Kalman filter and the sliding innovation filter

Connor Wilkinson, Waleed Hilal, S. Andrew Gadsden, John Yawney

Connor W. Wilkinson, Waleed Hilal, S. Andrew Gadsden, John Yawney, "Facial feature tracking method using a hybrid model of the Kalman filter and the sliding innovation filter," Proc. SPIE 12528, Real-Time Image Processing and Deep Learning 2023, 1252805 (13 June 2023); doi: 10.1117/12.2663896

SPIE.

Event: SPIE Defense + Commercial Sensing, 2023, Orlando, Florida, United States

Facial Feature Tracking Method using a Hybrid Model of the Kalman Filter and the Sliding Innovation Filter

Connor W. Wilkinson[ab], Waleed Hilal[a], S. Andrew Gadsden[a], John Yawney[b]
[a]Department of Mechanical Engineering, McMaster University, 1280 Main St W, Hamilton, ON, Canada, L8S 4L8; [b]Adastra Corporation, 175 Commerce Valley Dr W, Thornhill, ON, Canada, L3T 7P6

ABSTRACT

The purpose of this paper is to aid in detecting synthesized video (specifically created through the use of DeepFake) by exploring facial-feature tracking methods. Analyzing individual facial features, should allow for more successful detection of DeepFake videos according to H. Nguyen et al.'s research [22] and A. A. Maksutov's list of commonly use techniques to identify fabricated media [17]. To detect these facial features in images, Computer Vision techniques such as YOLOv3 [24] can be used. Once detected, object-tracking methods should be explored. This paper will compare the accuracy of three existing object-tracking methods: the minimum-distance approach, the Kalman Filter (KF) method, and the Sliding Innovation Filter (SIF) method. Following this comparison, the paper proposes a novel hybrid object-tracking approach, in which the benefits of the KF method and SIF method are combined to provide a time-gap tolerant object-tracking method. Each of the models are tested on their ability to track multiple objects that follow different trajectories and compared against one another to identify the most effective manner of tracking.

Keywords: Convolutional Neural Network (CNN), Deep Learning (DL), Generative Adversarial Network (GAN), Kalman Filter (KF), Sliding Innovation Filter (SIF)

1. INTRODUCTION

In recent years, data manipulation techniques and deep learning algorithms have transformed countless research topics under the umbrella of Artificial Intelligence (AI). One such subcategory has revolutionized several scientific fields over recent years. This is the field of machine learning (ML). MLs primary goal is to produce a system capable of accurately making decisions that would otherwise be made by humans [1]. To achieve this, concepts such as Neural Networks (NN) and Deep Learning (DL) were created. NN is a modelling approach wherein data is passed through multiple different computational layers. The first layer to which the input data is passed is called the 'input layer'. The neural network will eventually produce a desired output in a final layer called an 'output layer'. All layers between the input and output are considered 'hidden layers', with each determining and modifying parameters according to its internal computations. DL is an approach that integrated 'deep neural networks', which are NNs with multiple hidden layers [2]. As with standard NN solutions, DL can be categorized into the following: supervised, semi-supervised, partially supervised, and unsupervised techniques [3]. These techniques have been explored for decades, and with advancements in computing power, countless complex DL models have been developed. Examples include Convolutional Neural Networks (CNNs) and Generative Adversarial Networks (GANs).

1.1 Relevant Machine Learning Models

CNNs are a form of DL approach which gets its name from a unique 'convolutional' layer within the architecture. Drawing from the definition of 'convolution', meaning to use two or more mathematical operations together, these convolutional layers use multiple operations in its architecture such as sigmoid, ReLU or Tanh [4]. This technique was pioneered by LeCun et al. in 1990, attempting to recognize handwritten digits [5]. Although it was not very accurate due to the limited computing power available in 1990, it paved the way for future researchers to expand on its technique. In 2012, Krizhevsky et al. proposed AlexNet, which used CNNs to recognize objects in images. The result gave a much lower error rate than any other object recognition model at the time [6]. As a result, it won ILSVRC 2012, and CNNs became the most used DL technique at the time [7].

Another important ML technique that stemmed from DL architecture, was GANs. GANs use an adversarial approach to train their networks, where two components 'compete' against one another to satisfy their individual purpose [8]. These

Real-Time Image Processing and Deep Learning 2023, edited by Nasser Kehtarnavaz, Mukul V. Shirvaikar, Proc. of SPIE Vol. 12528, 1252805 · © 2023 SPIE
0277-786X · doi: 10.1117/12.2663896

components are the Generator, whose purpose is to generate proper data, and the Discriminator, whose purpose is to detect flaws in the data that the Generator provides. This process stops when the Discriminator cannot distinguish differences between the generated and actual data any longer [9]. This concept can be used in a plethora of fields, with the most prominent being creating [10] and manipulating [11] images.

ML models and techniques are utilized in a vast spectrum of fields including image processing, computer vision, speech recognition, natural language processing (NLP), and many others. The aforementioned techniques (CNNs and GANs) are most commonly used within the field of computer vision, and such methods have given rise to a novel model called DeepFake [12].

1.2 DeepFake Technology

DeepFake technology uses GAN models to produce fake images of an individual's face (called the target), using known features about the chosen individual. The generated images of the target can then be overlayed onto an existing image of another individual (called the source), essentially swapping their faces to result in a generated fake image of the target [13]. Such models are capable of producing hyper-realistic images and videos, that when paired with generated audio, can easily fool human perception [14, 15], as shown in Figure 1. Due to the incredible efficacy of DeepFakes, synthesized videos can cause serious threats to privacy, fraudulence, and the spread of misinformation. A recent report from Deeptrace Labs indicates that, as of 2019, there were 14,678 deepfake videos online and 96% of them are related to pornographic content [16], indicating the relevance and severity of the issue today.

Figure 1. DeepFake example showing the true image of Sean Connery (on the right) and the fabricated image of Burt Reynolds (on the left) [14].

As a result, researchers are constantly attempting to identify methods of DeepFake detection. Per previous research, there are several usual indicators of DeepFake videos including: too smooth skin, colour mismatches between the generated face and the original, temporal flicking around individual facial-features, inconsistent eye blinking rates and artifacts on small moving parts such as eyelashes, hairs, or small skin defects [17]. To assist in this research, open datasets such as FaceForensics++ [18] and Celeb-DF [19] offer vast datasets of true and synthesized images, and open platforms such as the DeepFake-o-meter [20] encourage collaboration on the topic. Using similar ML tools as those mentioned previously, detection techniques are constantly being introduced, such as DeepfakeStack [15] and XceptionNet [21].

Referring to the list of usual indicators, many of the most common methods to identify DeepFake videos relate to intricate details about unique facial-features (eyes, eyebrows, lips, etc). As a result, specifically identifying anomalies with individual facial-features is a highly effective approach. For instance, research done by H. Nguyen et al. outlines the success in focusing only on the subject's eyebrows [22].

1.3 Facial-Feature Recognition Techniques

To analyze individual facial-features, the images must be processed to identify their respective locations. Object-detection models use ML techniques, such as CNN, to locate specific objects in images. These models recognize consistent shapes and patterns within an image, based on learned images of recognizable objects. One of the most successful object-detection models used today is You Only Look Once (YOLO) by pjreddie [24]. In April 2018, YOLO creator Josh Redmon released YOLOv3, which improved its mAP metric from previous iterations of the model to 57.9%, which is at least as good as

other top object-tracking methods such as R-CNN and Focal Loss, while being significantly faster [25]. The existence of powerful object-detection methods such as YOLOv3 (and the releases of YOLOv4 and YOLOv5 in 2020) [26, 27] showcases the apparent feasibility of facial-feature recognition. An example of YOLOv3 detecting people's faces in an image is shown in Figure 2.

Figure 2. Results from YOLOv3 when recognizing individual faces in a crowd [28].

Unfortunately, the task of facial-feature recognition using object-detection techniques is non-trivial since the face is highly deformable [29] and there is a plethora of possible face types and shapes. Additionally, there are complications regarding the subject's privacy and security concerns. A study by L. Steinacker et al. shows the results from a cross-national public survey on the acceptance of the use of facial recognition technology in public affairs [23].

As shown in Tables 1 and 2, approximately 35% of individuals in the polled countries somewhat or strongly oppose the use of facial recognition in their countries, while approximately 20% of citizens in China and the UK and roughly 40% of citizens in Germany and the US oppose the idea of government surveillance. Though not the majority, these numbers show that a significant portion of the population in major countries have negative feelings towards surveillance and facial recognition being used on the public. Affirming the effect of the public's hesitation towards facial recognition technology, Meta (formerly Facebook) stopped using facial recognition altogether as of 2021 [30].

These object-detection models can be extended to track objects in videos, by detecting them in sequential frames. For real-time object-tracking, detection speed is essential, making YOLO one of the best options. The most commonly used method for linking the objects from frame-to-frame is using the minimum-distance approach, where an object in Frame B is considered to be the same as an object in Frame A if, out of all detected objects in Frame B, it is the closest that object in Frame A. This approach poses several issues when tracking multiple objects in a single video if there is a substantial time-delta between frames or if there are overlapping objects.

Table 1. Results of survey question: "Do you generally support or oppose the use of surveillance by your government in your country?" [23].

RESPONSE	CHINA	GERMANY	UK	US
STRONGLY OPPOSE	4%	15%	9%	12%
SOMEWHAT OPPOSE	12%	19%	15%	19%
NEITHER OPPOSE OR ACCEPT	32%	27%	31%	32%
SOMEWHAT ACCEPT	34%	32%	31%	26%
STRONGLY ACCEPT	18%	8%	13%	11%

Table 2. Results of survey question: “Do you accept or oppose the use of FRT in public?” [23].

RESPONSE	CHINA	GERMANY	UK	US
STRONGLY OPPOSE	4%	18%	14%	16%
SOMEWHAT OPPOSE	18%	21%	19%	22%
NEITHER OPPOSE OR ACCEPT	28%	24%	27%	27%
SOMEWHAT ACCEPT	39%	29%	28%	24%
STRONGLY ACCEPT	11%	8%	11%	11%

Another generally applicable complication with object-tracking stems from the data reliability for video-retrieval Internet of Things (IoT) systems. Due to the large size of image data, systems involving the retrieval of images often require high storage capacity. As a result, cloud computing systems are commonly regarded as the most effective storage method for image retrieval systems, due to its ability to manage large amounts of data [31]. One of the major downsides to cloud storage systems, is that to save costs, cloud storage often trades increased space for less data reliability [32]. As a result, it is probable that certain video frames will be lost when processing large quantities of image data. Developing object-tracking models that can account for such frame-drops, is an important problem.

To help address some of the aforementioned problems, such as complications with overlapping objects or substantial time differences between sequential images, more robust tracking models should be explored. One such approach is using estimation theory.

1.4 Estimation Theory

Estimation theory is used to analyze system state dynamics, in a system with measurement noise, to infer information on the system states [33]. The most popular method of Estimation Theory is the Kalman Filter (KF) [34]. For a linear system with Gaussian white noise and known system dynamics, the KF is the optimal filtering algorithm to track each of the system’s states [35]. As a result, KF has many tangible uses, including radar target tracking [35], music tempo and beat-pulse tracking [36] and even tracking object within videos [37]. Since the KF is a linear estimation technique, and real-world systems have non-linear components, several alternate non-linear forms have been introduced, such as the extended KF (EKF), unscented Kalman Filter (UKF) and the Sliding Innovation Filter (SIF) [33]. The SIF is a highly robust method, with similar accuracy to the normal KF, to estimate system states in a system where there is high modelling uncertainty [38]. As a result, in a situation where the system dynamics are not fully known, the SIF provides an accurate estimation method. Since system dynamics are seldom known when tracking objects in video, this method will likely give a higher accuracy than the KF, EKF or UKF alone.

Due to the assumed inefficacy of the minimum distance object-tracking method when attempting to track multiple overlapping objects in video, several other researchers have explored the tracking capabilities of the KF when paired with an object-detection tool such as YOLO [39, 40]. Though some results are promising, these researchers do not consider the newer SIF method for tracking, and do not consider unexpected time-jumps, which, as discussed, are quite common in video retrieval processes, especially when streaming mass amounts of image data.

The rest of this paper is divided into four sections. Section 2 considers each of the discussed existing tracking solutions: minimum-distance tracking, KF tracking, and SIF tracking. Section 3 tests each of these methods in simulated tracking experiments, and the results are shared. Section 4 outlines the proposed solution of the hybrid object-tracking method, where the KF and SIF models are combined to provide a time-gap tolerant solution. Section 5 sees the proposed solution run against the previously mentioned tests, and the results will be reviewed and discussed. In Section 6, future considerations will be outlined, and finally, in Section 7, the paper’s findings are reviewed.

2. EXISTING MODEL DEFINITIONS

As explained, object tracking methods function on video data by analyzing individual frames as separate images. These images are often captured at a set rate called the videos 'frame-rate'. When analyzing the individual images, metadata including the time that the image was captured will accompany the image. Then, the first step of object-tracking is to identify each of the objects within the image. This can be done using object detection models, such as YOLOv3. These detection models will be able to detect a specific type of object in the image, for instance a person's nose or eye. Unfortunately, since these object detection models are commonly trained on a wide variety of images, with training data containing variations of the object in question, they can seldom distinguish between sub-types of that object. For instance, in an image containing two people, an object detection algorithm can identify the locations of their respective noses, but cannot identify which nose belongs to which person. To solve this issue, object-tracking aims to analyze sequential images to relate objects between images.

2.1 Minimum-Distance Method

The first discussed method was the minimum-distance method, wherein the Euclidean distance (1) between the points identified in an image (called Image 2) at time t and each of the points identified in the image (called Image 1) at time t-1. The point identified in Image 2 that has the smallest Euclidean distance with some object A, is considered to also be object A.

The following equation defines the Euclidean distance:

$$d(q,p) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2} \tag{1}$$

where q,p are two points in Euclidean space, q_i, p_i are the i^{th} coordinates of q and p respectively, where $1 \leq i \leq n$, and n is the dimension of the Euclidean space.

Since this model is used on images, which produces data with two dimensions (x,y), the Euclidean distance equation (1) can be simplified to:

$$d(q,p) = \sqrt{(q_x - p_x)^2 + \left(q_y - p_y\right)^2} \tag{2}$$

where q,p are two points in Euclidean space, q_x, p_x are the x coordinates of q and p respectively, q_y, p_y are the y coordinates of q and p respectively.

2.2 Kalman Filter (KF) Method

The second method of tracking objects discussed in this paper is the KF method. Here the state dynamics of the system are estimated, which gives an estimated position of the object while accounting for the system noise.
Since this model is used on a discrete number of images, a discrete model can be used. As a result, an image will be taken at a discrete time, corresponding to some integer k. The following image will correspond to the integer $k+1$. The time between images k and $k+1$ is referred to as Δt.

The whole filtering process is composed of a prediction and update stage. The referenced prediction stage predicts the state of object m in image k, $\mathbf{x}_m(k)$, and the covariance matrix of object m in image k, $\mathbf{P}_m(k)$, knowing information the state in image $k-1$, $\mathbf{x}_m(k-1)$, and its covariance matrix in image $k-1$, $\mathbf{P}_m(k-1)$. The update stage updates both predicted values (the state and the covariance matrix) depending on the measured value $\boldsymbol{z}(k)$.

For the purposes of this paper, the true state vector considers an objects position, x, and velocity, v.

$$\mathbf{x}_m = [x_m, v_m]^T \tag{3}$$

where $\mathbf{x}_m$ is the state of object m, x_m is the position of object m and v_m is the velocity of object m.

Each image produces data with two dimensions, so both , x_m and , v_m will be 2-dimensional vectors, resulting in $\mathbf{x}_m$ being a 4-dimensional vector.

The prediction stage equations are defined by:

$$\mathbf{x}_m(k) = A \cdot \mathbf{x}_m(k-1) + B \cdot \mathbf{u}_m(k-1) \tag{4}$$

$$\mathbf{P}_m(k) = A \cdot \mathbf{P}_m(k-1) \cdot A + Q \tag{5}$$

where $\mathbf{x}_m(k)$ is state of object m in the image k, $\mathbf{x}_m(k-1)$ is state of object m in the image $k-1$, A is the state transition matrix, B is the control input matrix, $\mathbf{u}_m(k-1)$ is the control input for image $k-1$, $\mathbf{P}_m(k)$ is covariance matrix of object m in the mage k, $\mathbf{P}_m(k-1)$ is covariance matrix of object m in the image $k-1$, A is state transition matrix, and $\boldsymbol{Q}$ is the process noise covariance.

The control input in this scenario is the acceleration of each of the object. As this value is not known, due to the unpredictability of the movement of objects in video, it's initial value is assumed to be zero. As a result, the state prediction (4) can be rewritten:

$$\mathbf{x}_m(k) = A \cdot \mathbf{x}_m(k-1) \tag{6}$$

The state space matrix, A, can be derived from kinematic equations, and the process noise covariance, $\boldsymbol{Q}$, was determined through experimental procedure to be:

$$Q = \begin{bmatrix} 0.25 \\ 0.25 \\ 0 \\ 0 \end{bmatrix} \tag{7}$$

The update process is defined by the following equations:

$$\mathbf{x}_m(k) = \mathbf{x}_m(k) + \mathbf{K} \cdot \left(\boldsymbol{z}_m(k) - \boldsymbol{C} \cdot \mathbf{x}_m(k)\right) \tag{8}$$

$$\mathbf{P}_m(k) = J \cdot \mathbf{P}_m(k) \cdot J^T + \mathbf{K} \cdot \boldsymbol{R} \cdot \mathbf{K}^T \tag{9}$$

$$J = (I - (\boldsymbol{K} \cdot \boldsymbol{C})) \tag{10}$$

where, $\mathbf{x}_m(k)$ is state of object m in the image k, $\mathbf{K}$ is the Kalman gain, $\mathbf{z}_m(k)$ is the measurement of object m in the image k, $\boldsymbol{C}$ is the measurement mapping matrix, $\mathbf{P}_m(k)$ is state of object m in the image k, $\boldsymbol{R}$ is the measurement noise covariance and I is the identity matrix.

The Kalman gain, $\mathbf{K}$, is given by:

$$\mathbf{K} = \frac{\mathbf{P}_m(k) \cdot \mathbf{C}^T}{\boldsymbol{C} \cdot \mathbf{P}_m(k) \cdot \mathbf{C}^T + \boldsymbol{R}} \tag{11}$$

where, $\boldsymbol{P}_m(k)$ is state of object m in the image k, $\boldsymbol{C}$ is the measurement mapping matrix and $\boldsymbol{R}$ is the measurement noise covariance.

Since the object detection algorithms only gave information about the object's positions, the measurement mapping matrix, $\boldsymbol{C}$, was defined as:

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{12}$$

Similar to $\boldsymbol{Q}$, the measurement noise covariance, $\boldsymbol{R}$, was experimentally determined to be:

$$\boldsymbol{R} = \begin{bmatrix} 10^{-4} & 0 \\ 0 & 10^{-5} \end{bmatrix} \tag{13}$$

To track objects in video data, the KF can be used to predict the estimated location of the object in the upcoming video frame, given its location in the past video frame. Since the KF considers the velocity of the objects, as well as accounting for noisy measurements from the object detection, this prediction should give an accurate estimate of the object's location in the subsequent image. For each object detected in the first frame, there will be a corresponding KF model, each initialized with their corresponding object's position. Each KF model will have an associated label, which will allow each

object to be tracked. Once the unlabeled objects are detected in the following frame, the label for each object will correspond to the closest KF model's predicted location, found by determining the Euclidean distance (2) between the predicted location and the unlabeled object. In other words, consider the situation with two objects and two KF models, A and B. If the KF with label A has a predicted value that was closer to an unlabeled object in the new image than that of the KF with the label B, the new object would be labelled A. Once labelled, the KF will be updated with the labelled object's new measured position, using the update equations (8, 9). If more objects are detected in later images than there are KF models, the labelling process will occur, and all unlabeled objects will be assigned a new KF model, initialized using its corresponding objects position. Conversely, if there are more models than objects found, the model's state will be projected for each image using the KF filter's prediction equation (6) until the it labels a new object, or until its position exceeds the boundaries of the image.

2.3 Sliding Innovation Filter (SIF) Method

The final tested existing tracking method is using the SIF. This filtering process is identical to the KF method in its prediction stage, but the update stage has been modified. Specifically, the Kalman gain (11) has been changed to the SIF gain (14) [38].

$$\boldsymbol{K} = \boldsymbol{C}^{+}\overline{sat}\left(\left|\frac{innov}{\delta}\right|\right) \tag{14}$$

$$innov = \mathbf{z}_m(k) - C \cdot \mathbf{x}_m(k) \tag{15}$$

where $\boldsymbol{C}$ is the measurement matrix, $\boldsymbol{C}^{+}$ is the pseudoinverse of the measurement matrix, $\overline{sat}$ refers to the diagonal of the saturation term, sat is a saturation value (yielding a value between -1 and 1), $innov$ is the innovation value, calculated using (14), δ is the sliding boundary layer width, $\boldsymbol{z}_m(k)$ is the measured position of object m in image k, and $\boldsymbol{x}_m(k)$ is the predicted position of object m in image k.

The variable feature in calculation of SIF gain is the sliding boundary layer term, δ. This variable is defined based on the amount of uncertainty in the estimation process [38]. Thus, for high uncertainties, the boundary layer term should have a correspondingly high value.

Through trial and error, and due to the highly uncertain movement of the motion of an individual in a video, the sliding boundary layer term was chosen to be [0.8,1.2] for the initial trials.

The tracking set-up is identical to that of the KF method, wherein an SIF model is initialized for each object found in the first image. For any inconsistencies between the number of models and objects, a similar technique to that used for the KF model is used. Following this, the same procedure as the KF model is also used, where the prediction made by the SIF model will determine the label of the objects in each image and the models will be updated with the position of the labelled object.

3. EXISTING MODEL DEFINITIONS

Since this paper means to test the efficacy of the tracking methods, rather than the object detection methods, a simulation was created wherein multiple cartesian points followed multiple custom paths. Each of these points represent a recognized object by an object detection model such as YOLOv3. This permitted testing for multiple scenarios, without having to rely on object trajectories in actual video footage. Each trajectory consisted of 100 points, which represented individual frames in video footage. Randomly generated noise was added to each of the trajectories, to simulate inconsistencies in the object-detection. Three scenarios were used to test the efficacy of the existing solutions:

A. A pair of linear and non-linear paths that have non-intersecting objects. The object along Path 0 runs right to left, where the object along Path 1 runs left to right. The trajectories used are:

$$\text{Path 0:} \qquad y = 10\sin\left(\frac{x}{10}\right) \tag{16}$$

$$\text{Path 1:} \qquad y = -x + 50 \tag{17}$$

B. A pair of linear paths that have intersecting objects. The objects on both Path 0 and Path 1 are travelling left to right. The trajectories used are:

$$\text{Path 0:} \qquad y = x - 50 \tag{18}$$

$$\text{Path 1:} \qquad y = -x + 50 \tag{19}$$

C. A pair of linear paths that have intersecting objects. The objects on both Path 0 and Path 1 are travelling left to right. The trajectories used are:

$$\text{Path 0:} \qquad y = 10 \sin\left(\frac{x}{10}\right) \tag{20}$$

$$\text{Path 1:} \qquad y = -x + 50 \tag{21}$$

For the purpose of this paper, these scenarios will be referenced as Scenario A, Scenario B and Scenario C respectively.

Beginning with Scenario A, since the objects did not intersect, and remained relatively far away from

one another along their routes, it was expected that each of the tracking methods would label the objects properly. Figures 3 and 4 show the trajectories of Scenario A with the labels of each tracking method at different times along the object's path. The non-linear sin-wave trajectory was chosen due to its smooth nature, which mimics a human's anticipated motions, since they seldom rapidly change location between images captured at 32 FPS or higher, but often follow non-linear trajectories. Table 3 outlines the label prediction accuracy for each of these tracking methods using a customized accuracy metric. The accuracy metric initializes each of the objects with a label and measures the percentage of frames in the video where the predicted label equals the initialized label. The simulation was run 100 times to acquire the data shown in Table 3.

Table 3. Outlines the label prediction accuracy for each of the existing methods on Scenario A through 100.

RESPONSE	**Minimum Distance Method**	**KF Method**	**SIF Method**
Label Prediction Accuracy	100%	100%	100%

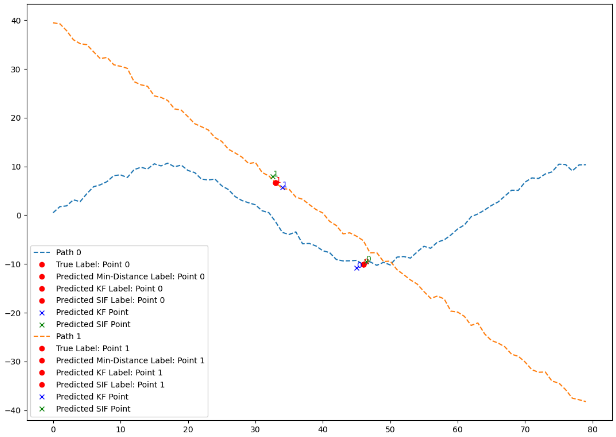

Figure 3. Scenario A shown with the objects beginning their trajectories. Labels shown in the Figure's Legend.

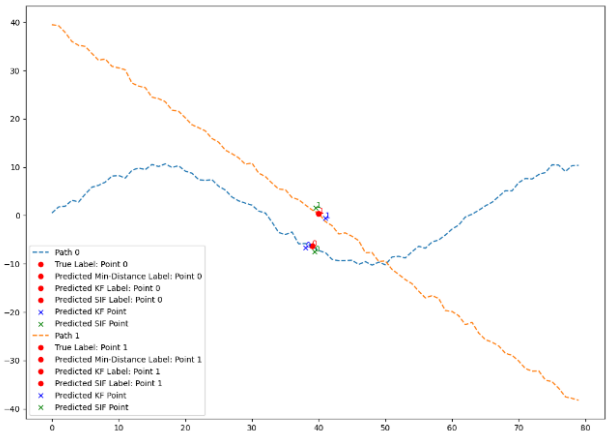

Figure 4. Scenario A shown with the objects halfway through their trajectories. Labels shown in the Figure's Legend.

As predicted, each of the tracking methods saw no complications when tracking objects that did not intersect.

Scenario B saw the objects intersect. Here, it was predicted that the filtering methods would work more effectively than the minimum-distance method, since when the paths intersect, it is likely that there will be at least one frame where the first object's position in the previous frame is closer to the second object's new position, and their labels would reverse. Since the filter tracking models account for the object's velocity, even if they incorrectly label the objects at the time of intersection, they should correct themselves afterwards. Figures 5 and 6 show the trajectories of Scenario B before and after the intersection. Table 4 outlines the label prediction accuracy for each of the tracking methods. The simulation was run 100 times to acquire the data shown in Table 4.

Table 4. Outlines the label prediction accuracy for each of the existing methods on Scenario B through 100 runs.

RESPONSE	Minimum Distance Method	KF Method	SIF Method
Label Prediction Accuracy	72.1%	89.6%	68.2%

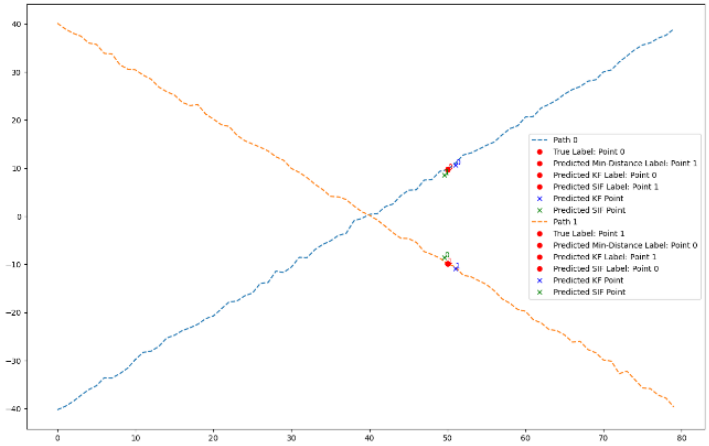

Figure 5. Scenario B shown with the objects prior to their intersection. Labels shown in the Figure's Legend.

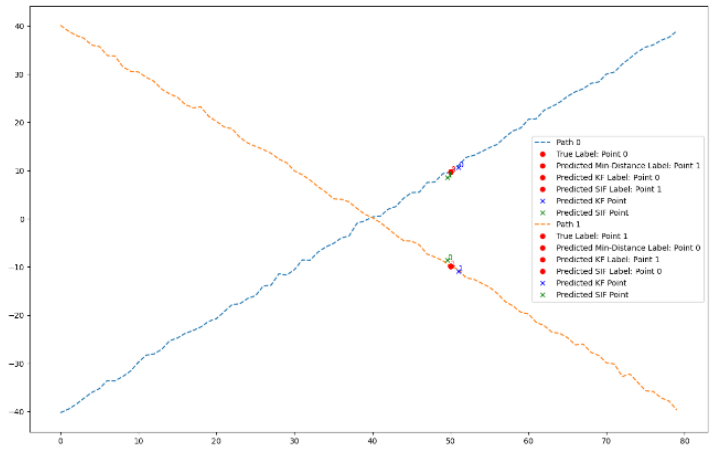

Figure 6. Scenario B shown with the objects after their intersection. Labels shown in the Figure's Legend.

Aligning with the prediction, the KF method outperformed the minimum distance method. Conversely, the SIF method did not outperform either of the methods. After further analysis, this is found to be because of the extreme robustness of the SIF method, and its ability to accommodate to changing directions. As shown in A. Gadsden et al.'s paper [38], given an abrupt change in an objects position, the SIF method will accommodate to its new position and trajectory much better than the KF method. In regard to object tracking, this hinders its ability to follow a specific point after two points intersect, since when a new point with a new trajectory is introduced, if the incorrect object is labelled at the point of intersection, the SIF method will accommodate to the new object trajectory extremely well. This means that its predicted trajectory will change, and it will likely begin to follow the incorrect object. On the other hand, the KF method is not as robust, so when a new object with a new trajectory is introduced, even if the KF method incorrectly predicts a label for some few frames at the point of intersection, it will not track it 'well' enough to completely alter its original trajectory. As a result, it will more often persist on its original trajectory, and correct its labelling once the objects distance from one another.

Scenario C uses the same trajectories as that of Scenario A but sends the objects in the same directions along their respective paths. As a result, they intersect. These trajectories are shown in Figures 7 and 8, with the label prediction accuracies shown in Table 5.

Table 5. Outlines the label prediction accuracy for each of the existing methods on Scenario C through 100 runs.

RESPONSE	**Minimum Distance Method**	**KF Method**	**SIF Method**
Label Prediction Accuracy	73.5%	88.2%	69.9%

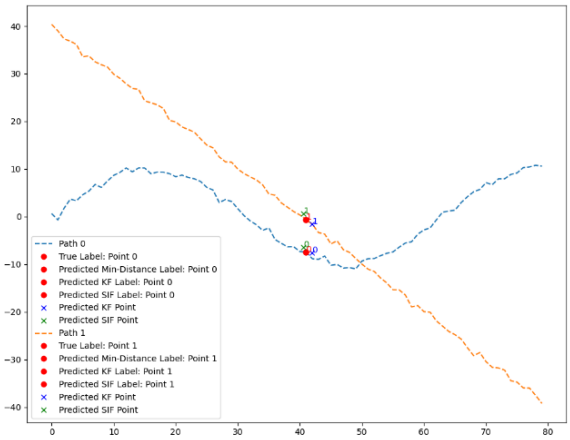

Figure 7. Scenario C shown with the objects prior to their intersection. Labels shown in the Figure's Legend.

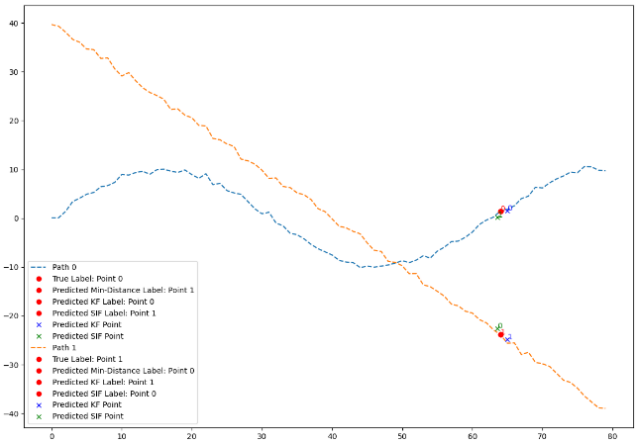

Figure 8. Scenario C shown with the objects following their intersection. Labels shown in the Figure's Legend.

The Scenario showed similar results to that of Scenario B, for similar reasons, even with the non-linearity included.

As previously discussed, one of the common flaws with IoT image retrieval systems is data retention. Due to expected data loss, it is extremely important to test the results of tracking systems with time-gaps. To test this, the scenarios were tested again, but gaps in the data of random magnitude and random location were added in each of the object's trajectories. Per run, a single time-gap was added, with a maximum allowed time-gap magnitude of 10 missing data points.

Since the results from Scenario's B and C were so close, it was assumed that the non-linear path did not heavily affect the testing method. As a result, only Scenarios A and B were tested with randomized time-gaps.

An example of one of the random time-jumps runs for Scenario's A and B is shown in Figures 9, 10, 11 and 12. The accuracy results for each scenario are shown in Table 6.

Table 6. Outlines the label prediction accuracy for each of the existing methods on Scenarios A and C with random time-gaps through 100 runs.

RESPONSE	Minimum Distance Method	KF Method	SIF Method
Label Prediction Accuracy for Scenario A with Time Gap	51.4%	51.5%	51.5%
Label Prediction Accuracy for Scenario B with Time Gap	50.6%	50.2%	51.3%

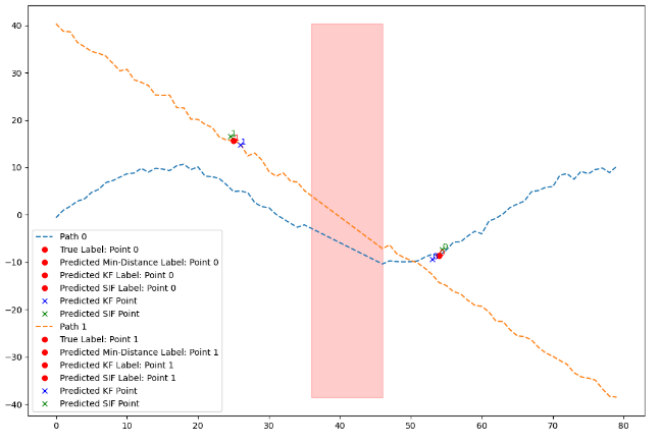

Figure 9. Scenario A shown with random time-gap in the middle of the run. Objects in this image are yet to encounter the time-gap.

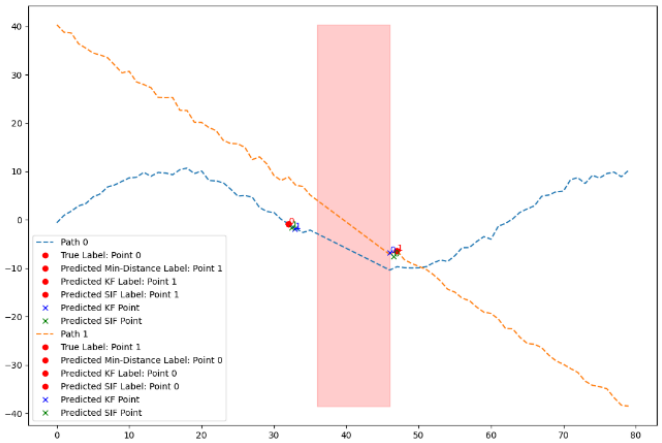

Figure 10. Scenario A shown with random time-gap in the middle of the run. Objects in this image have encountered the time-gap.

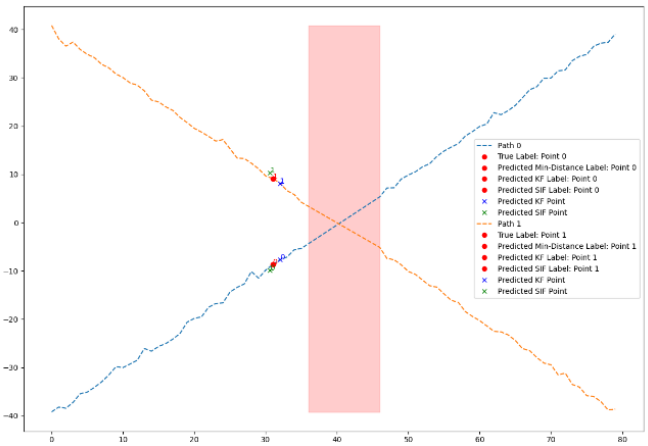

Figure 11. Scenario B shown with random time-gap in the middle of the run. Objects in this image are yet to encounter the time-gap.

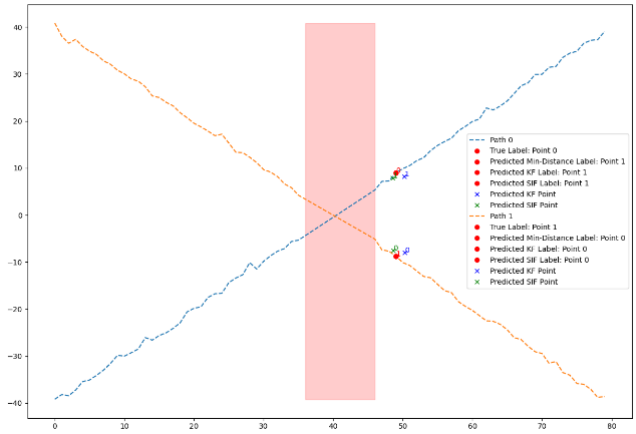

Figure 12. Scenario B shown with random time-gap in the middle of the run. Objects in this image have encountered the time-gap.

As shown, each of the results are significantly worse than their corresponding scores without the time-gap. Due to the randomized placements of the time gaps, it is expected that the average starting placement for the time-gaps is approximately halfway through the object's trajectory. Therefore, an accuracy measure of approximately 50% indicates that these tracking methods are wholly inaccurate once they encounter the time-gap. Further thought corroborates this theory, since the position, and predicted position, of the object in the frame directly prior to the time-gap, will be closer to the other object directly following the time-gap, resulting in the switching of labels.

4. PROPOSED HYBRID KF-SIF SOLUTION

To account for this inaccuracy, a novel approach was created wherein the benefits of the KF and SIF methods were utilized to permit tracking with time-gaps in the image data. As with the KF and SIF methods, a hybrid KF-SIF model is initialized for each object found in the first image. For any inconsistencies between the number of models and objects, a similar technique to that used for the KF model and SIF model is used.

As explained, within the image metadata are timestamps for when each image was captured. Using this, along with the known frame rate of the camera, one can derive the number of missed frames (22).

$$mf = \left\lfloor \left(\frac{\Delta t}{fr} \right) - 1 \right\rfloor \tag{22}$$

where mf is the integer number of missed frames, Δt is the time difference between the timestamps of two sequential images, and fr is the camera's frame-rate.

With this information, the hybrid solution will utilize the KF tracking model when the $mf=0$ (ie. there are no missing frames). Otherwise, if $mf \geq 1$, the KF model will be projected exactly mf times using the KF model's prediction equation (6). This method will update the filter's predicted state, $\mathbf{x}_{\mathrm{m}}$, accounting for however many images were lost. Along with updating the state, the covariance matrix, $\boldsymbol{P}_m$, will also be updated using the associated KF update equation (5).
The new projected state will be used as the predicted location of the object in the new image. The minimum distance method, used in each of the filtering tracking methods, is then used to label the closest object. Once labelled, the KF will be updated with the labelled objects new measured position, using the update equations (8, 9).
After labelling the found object following the time-gap, the hybrid KF-SIF model will determine if the object is significantly far away from its previous location. To do this, a metric called the allowed KF distance, $akfd$, is calculated (23).

$$akfd = akfd_p \times \sqrt{im_h^2 + im_w^2} \tag{23}$$

where $akfd_p$ is the defined allowed KF distance percentage, im_h is the image height and im_w is the image width.

The allowed KF distance percentage, $akfd_p$, is the allowable percentage of the image's diagonal length where the points are considered close enough together to not require the SIF recovery method. For this paper, this value was set to 0.05.

This parameter permits the hybrid KF-SIF model to determine which update method to use, following a time-gap. If the Euclidean distance (2) between the object's previous location and its new position is above the allowed KF distance, $akfd$, then the update mechanics of the SIF are utilized. If the SIF update functions are used, the SIF model will be used for a set 'recovery' period, defined by a set of video frames, until the model switches back to using the KF model. The length of the 'recovery' period used in this paper was 5 image frames. The reason for this, is due to the SIF model's ability to handle large and abrupt changes in an object's location. As previously discussed and shown by A. Gadsden et al. [38], given an abrupt change in an objects position, the SIF method will accommodate to its new position and trajectory much better than the KF method.

Since the SIF model is only utilized when there is an abrupt jump between points and it is assumed that the movement in the x-direction would be less variable than the movement in the y-direction, the sliding boundary layer was chosen, through trial and error, to be [2,4].

5. COMPARING THE HYBRID KF-SIF SOLUTION

Since this object only differs from the usual KF method when there is a time-gap, it was only tested on scenarios where time-gaps were introduced. The hybrid KF-SIF method's accuracy with no time-gaps is equal to that of the raw KF model. These tests were on the same randomized time-gap scenarios shown in Figures 9, 10, 11 and 12. The results are shown below in Table 7.

Table 7. Outlines the label prediction accuracy for each of the existing methods and the hybrid KF-SIF method on Scenarios A and C with random time-gaps through 200 runs.

RESPONSE	Minimum Distance Method	KF Method	SIF Method	Hybrid KF-SIF Method
Label Prediction Accuracy for Scenario A with Time Gap	51.4%	51.5%	51.5%	93.23%
Label Prediction Accuracy for Scenario B with Time Gap	50.6%	50.2%	51.3%	94.25%

As seen, the accuracy of this method greatly improves upon the other methods, in situations with a time-gap. One of the reasons why, is due to the time-gap overlapping with the intersection of the two points, which in effect, removes that complexity (as the points will be projects across their intersection). Fortunately, this finding leads to an extension of this work.

6. FUTURE CONSIDERATIONS

6.1 Extensions of the Work

The proposed solution in this paper shows the extreme accuracy of using a KF model and SIF model in tandem, to track objects in images when they unexpectedly jump locations due to a time-gap. A prominent extension of this work can be found when considering this problem in relation to human facial features. Due to a person's facial structure, and the substance around their facial features, when tracking an individual facial feature on two people (for instance their respective noses), it is impossible for them to exactly overlap on an image. If a person is travelling in front or behind of another individual, their facial features will be lost to the camera for several frames. By extending the hybrid KF-SIF solution, the trajectory of any disappearing facial features can be tracked and re-labelled once appearing again.

6.2 Future Work

After exploring the capabilities of these different tracking methods for the purposes of tracking facial features in video footage, several potential future considerations can be outlined.

Firstly, the aforementioned difficulties with public surveillance must be explored. Tracking individuals using Computer Vision techniques may be difficult in the future, with more powerful Computer Vision techniques leading to more public concern. These concerns should be understood carefully before implementing solutions such as these. Fortunately, these tracking methods can be used for purposes other than specifically tracking objects in images.

Another consideration stems from the fact that these solutions, including the proposed hybrid approach, rely only on positional data from the detected objects. Other Computer Vision techniques such as Canny Edge detection, or simple blurring techniques using OpenCV, could help identify the shapes of certain facial features. With this added knowledge, recognizing unique features could increase labelling accuracies. For example, when recognizing the eyes of two individuals, the ratio of the length of one individual's top eyelid line, to the length of their bottom eyelid line, could prove to be unique to the individual. Seeing as this ratio should seldom change, no matter the size of the individual's eye in the frame, it could lead to enhanced recognition.

7. CONCLUSION

This paper has explored the optimal ways of detecting DeepFake videos, among which, identifying facial feature anomalies is prominent. Delving into the methods in which facial features can be identified, and tracked in video, the paper identified commonly used object detection and object tracking techniques. A survey identified YOLO [24] as one of the most effective object detection methods. Then, experimental results showed that each of the explored tracking methods performed almost identically when tracked objects had no interference. When the objects did overlap, the KF tracking method provided the best tracking accuracy compared with the minimum-distance tracking method (which was outperformed due to its simplicity) and the SIF method (which was outperformed due to its robustness). Finally, understanding that video-retrieval IoT solutions often have issues with data dependency due to the massive storage capacity requirements, a novel hybrid KF-SIF approach was proposed, that provided a time-gap tolerant solution for the object tracking problem, that retained the accuracy of the KF method in situations without a time-gap, and greatly improved upon the tracking accuracies of each of the existing tracking methods when a time-gap was introduced.

REFERENCES

[1] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders and K.-R. Müller, "Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications," *Proceedings of the IEEE,* vol. 109, no. 3, pp. 247-278, March 2021.

[2] T. Sarker, S. Noor and A. Uzzal, "Basic Application and Study of Artificial Neural Networks," *SK International Journal of Multidisciplinary Research Hub,* vol. 4, no. 4, pp. 1-12, April 2017.

[3] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. V. Essen, A. A. S. Awwal and a. V. K. Asari, "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches," *arXiv: 1803.01164,* March 2018.

[4] D. Arora, M. Garg and M. Gupta, "Diving deep in Deep Convolutional Neural Network," in *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, Greater Noida, India, 2020.

[5] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel, "Handwritten Digit Recognition with a Back-Propagation Network," *NIPS,* 1989.

[6] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems,* vol. 25, 2012.

[7] J. Fan, W. Xu, Y. Wu and Y. Gong, "Human Tracking Using Convolutional Neural Networks," *IEEE Transactions on Neural Networks,* vol. 21, no. 10, pp. 1610-1623, October 2010.

[8] C. Wang, C. Xu and X. Yao, "Evolutionary Generative Adversarial Networks," *IEEE Transactions on Evolutionary Computation,* vol. 23, no. 6, pp. 921-934, December 2019.

[9] S. Agarwal, N. Girdhar and H. Raghav, "A Novel Neural Model based Framework for Detection of GAN Generated Fake Images," in *2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, Noida, India, 2021.

[10] T. Karras, S. Laine and T. Aila, "A Style-Based Generator Architecture for Generative Adversarial Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2020.

[11] Y. Viazovetskyi, V. Ivashkin and E. Kashin, "StyleGAN2 Distillation for Feed-Forward Image Manipulation," *Computer Vision -- ECCV 2020,* pp. 170-186, 2020.

[12] torzdf, "deepfakes_faceswap," github.com, 14 August 2020. [Online]. Available: https://github.com/deepfakes/faceswap. [Accessed 25 October 2021].

[13] H. A. Khalil and S. A. Maged, "Deepfakes Creation and Detection Using Deep Learning," in *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, Cairo, Egypt, 2021.

[14] Crescendo, "Best Of Deep Fakes Compilation," YouTube, 2020. [Online]. Available: https://www.youtube.com/watch?v=xkqflKC64IM&ab_channel=Crescendo.

[15] M. S. a. S. A. H. Rana, "DeepfakeStack: A Deep Ensemble-based Learning Technique for Deepfake Detection," in *2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*, New York, USA, 2020.

[16] H. Ajder, G. Patrini, F. Cavalli and L. Cullen, "The State of Deepfakes: Landscape, Threats, and Impact," Deeptrace, 2019.

[17] A. A. Maksutov, V. O. Morozov, A. A. Lavrenov and A. S. Smirnov, "Methods of Deepfake Detection Based on Machine Learning," in *2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, St. Petersburg and Moscow, Russia, 2020.

[18] A. Rössler, D. Cozzolino, L. Verdoliva, C. Riess, J. Thies and M. Niessner, "FaceForensics++: Learning to Detect Manipulated Facial Images," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea, 2019.

[19] Y. Li, X. Yang, P. Sun, H. Qi and S. Lyu, "Celeb-DF: A Large-Scale Challenging Dataset for DeepFake Forensics," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, USA, 2020.

[20] Y. Li, C. Zhang, P. Sun, L. Ke, Y. Ju, H. Qi and S. Lyu, "DeepFake-o-meter: An Open Platform for DeepFake Detection," in *2021 IEEE Security and Privacy Workshops (SPW)*, San Francisco, USA, 2021.

[21] Y. S. Malik, N. Sabahat and M. O. Moazzam, "Image Animations on Driving Videos with DeepFakes and Detecting DeepFakes Generated Animations," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, Bahawalpur, Pakistan, 2020.

[22] H. M. Nguyen and R. Derakhshani, "Eyebrow Recognition for Identifying Deepfake Videos," in *2020 International Conference of the Biometrics Special Interest Group (BIOSIG)*, Darmstadt, Germany, 2020.

[23] L. Steinacker, M. Meckel, G. Kostka and D. Borth, "Facial Recognition: A cross-national Survey on Public Acceptance, Privacy, and Discrimination," in *ICML 2020 - Law and Machine Learning Workshop*, Vienna, Austria, 2020.

[24] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, USA, 2016.

[25] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," April 2018. [Online]. Available: https://pjreddie.com/media/files/papers/YOLOv3.pdf. [Accessed 26 October 2021].

[26] A. B. Liao, C.-Y. Wang and H.-Y. Mark, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv:2004.10934*, 2020.

[27] G. Jocher, "Yolov5," 25 June 2020. [Online]. Available: https://github.com/ultralytics/yolov5. [Accessed 25 October 2021].

[28] L. Zhang, M. Shi and Q. Chen, "Crowd Counting via Scale-Adaptive Convolutional Neural Network," in *IEEE Winter Conference on Applications of Computer Vision*, Lake Tahoe, 2018.

[29] E.-J. Ong and R. Bowden, "Robust Facial Feature Tracking Using Shape-Constrained Multiresolution-Selected Linear Predictors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1844-1859, September 2010.

[30] K. Hill and R. Mac, "Facebook, Citing Societal Concerns, Plans to Shut Down Facial Recognition System," *The New York Times*, 2021.

[31] J. Yang, B. Jiang and H. Song, "A distributed image-retrieval method in multi-camera system of smart city based on cloud computing," *Future Generation Computer Systems*, vol. 81, pp. 244-251, 2018.

[32] Y. Zhuang, N. Jiang, Z. Wu, Q. Li, D. K. W. Chiu and H. Hu, "Efficient and robust large medical image retrieval in mobile cloud computing environment," *Information Sciences*, vol. 263, pp. 60-86.

[33] A. S. Lee, S. A. Gadsden and M. Al-Shabi, "An Adaptive Formulation of the Sliding Innovation Filter," *IEEE Signal Processing Letters*, vol. 28, pp. 1295-1299, 2021.

[34] A. S. Gadsden, "Smooth Variable Structure Filtering: Theory and Applications," McMaster University, Hamilton, Canada, 2011.

[35] S. A. Gadsden, H. H. Afshari and S. R. Habibi, "Development of a sliding mode controller and higher-order structure-based estimator," in *2016 IEEE Transportation Electrification Conference and Expo (ITEC)*, Dearborn, USA, 2016.

[36] L. Jianxing, C. Qingliang, B. Defeng, Z. Xu and W. Yanting, "A Radar Target Tracking Method based on Coordinate Transformation KF," in *2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing, China, 2018.

[37] Y. Shiu and C.-C. J. Kuo, "On-line Music Beat Tracking with Kalman Filtering and Probability Data Association (KF-PDA)," in *2008 Digest of Technical Papers - International Conference on Consumer Electronics*, Las Vegas, USA, 2008.

[38] S. A. Gadsden and M. Al-Shabi, "The Sliding Innovation Filter," *IEEE Access*, vol. 8, pp. 96129 - 96138, May 2020.

[39] V. Paidi, H. Fleyeh, J. Hakansson and R. G. Nyberg, "Tracking Vehicle Cruising in an Open Parking Lot Using Deep Learning and Kalman Filter," *Journal of Advanced Transportation*, vol. 2021, 2021.

[40] A. Auguste, W. Kaddah, M. Elbouz, G. Oudinet and A. Alfalou, "Behavioral Analysis and Individual Tracking Based on Kalman Filter: Application in an Urban Environment," *Sensors*, vol. 21, 2021.