Video Face Tracking for IoT Big Data using Improved Swin Transformer based CSA Model

[1]Anbumani K, [2]Cuddapah Anitha, [3]Achuta Rao S V, [4]Praveen Kumar K, [5]Meganathan Ramasamy and [6]Mahaveerakannan R

[1]Department of Electronics and Instrumentation Engineering, Sri Sairam Engineering College, Chennai, India.
[2]Department of Computer Science and Engineering, School of Computing, Mohan Babu University, (Erstwhile Sree Vidyanikethan Engineering College), Andhra Pradesh, India.
[3]Data Science Research Laboratories, Sree Dattha Institute of Engineering & Science, Sheriguda, Telangana, India.
[4]Department of Information Technology, Kakatiya Institute of Technology and Science, Warangal, India.
[5]Department of Computing, De Montfort University Kazakhstan, Al-Farabi Ave, Republic of Kazakhstan.
[6]Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu, India.
[1]anbumani.ei@sairam.edu.in, [2]anithacuddapah@vidyanikethan.edu, [3]sreedatthaachyuth@gmail.com, [4]kpk.it@kitsw.ac.in, [5]rmeganathan@gmail.com, [6]mahaveerakannanr.sse@saveetha.com

Correspondence should be addressed to Mahaveerakannan R : mahaveerakannanr.sse@saveetha.com

Article Info
Journal of Machine and Computing (http://anapub.co.ke/journals/jmc/jmc.html)
Doi: https://doi.org/10.53759/7669/jmc202404029
Received 15 May 2023; Revised from 26 September 2023; Accepted 07 January 2024.
Available online 05 April 2024.

Abstract – Even though Convolutional Neural Networks (CNNs) have greatly improved face-related algorithms, it is still difficult to keep both accuracy and efficiency in real-world applications. The most cutting-edge approaches use deeper networks to improve performance, but the increased computing complexity and number of parameters make them impractical for usage in mobile applications. To tackle these issues, this article presents a model for object detection that combines Deeplabv3+ with Swin transformer, which incorporates GLTB and Swin-Conv-Dspp (SCD). To start with, in order to lessen the impact of the hole phenomena and the loss of fine-grained data, we employ the SCD component, which is capable of efficiently extracting feature information from objects at various sizes. Secondly, in order to properly address the issue of challenging object recognition due to occlusion, the study builds a GLTB with a spatial pyramid pooling shuffle module. This module allows for the extraction of important detail information from the few noticeable pixels of the blocked objects. Crocodile search algorithm (CSA) enhances classification accuracy by properly selecting the model's fine-tuning. On a benchmark dataset known as WFLW, the study experimentally validates the suggested model. Compared to other light models, the experimental findings show that it delivers higher performance with significantly fewer parameters and reduced computing complexity.

Keywords – Convolutional Neural Networks, Crocodile Search Algorithm, Global local Transformer Block, Face Tracking, Spatial Pyramid, Pooling Shuffle Module.

I. INTRODUCTION

The expansion of the Internet of Things (IoT) is a major representation of the information era. One fundamental component of the IoT is video monitoring systems, which have numerous practical applications in areas like intelligent anti-theft systems, intelligent access control systems, and intelligent systems for monitoring the care of the elderly [1]. In particular, intelligent algorithms are first deployed to each edge end (e.g., security cameras) to process the incoming data in real time; subsequently, the back-end (server) receives the processed recognition or analysis results and stores them or even triggers an alarm [2]. By repurposing the transmission data to include analytical results rather than the original data, we can achieve real-time needs while simultaneously reducing data transmission and increasing security [3]. Intelligent algorithms used under this system must meet stringent requirements, such as increased recognition accuracy, faster processing speed, and fewer parameters (due to the restricted memory of the edge device). Given this real-world need, this study will center on studies that investigate video face-recognition techniques that can be used in Internet of Things (IoT) monitoring systems [4].

Facial recognition has already found numerous practical applications in our daily lives, such as in sign-in systems and fugitive tracking systems. Videos and image sets provide more information about the items in them than a single

image can, such as different positions, lighting, and expressions [5]. Therefore, the picture set classification task—a method for studying the classification problem—must be prioritized [6]. By comparing the probing set to the gallery movies, image set classification (also known as set-based video recognition) can determine which labels to give the probe set [7]. By eliminating the need to identify each frame of a video independently, set-based video recognition tasks are able to immediately compute labels for entire films, significantly increasing computation speed compared to single image-based identification tasks [8]. The two main methodologies for set-based video face recognition are video representation and precise distance measurement, since each video contains a large variety of appearance variations [9].

Also, a face detector is usually needed beforehand by most of the current algorithms that estimate head poses and align the face. For that reason, they will not be able to achieve the theoretical speed. To further expedite head posture estimation and face alignment in video-based processing, face tracking eliminates the need for face identification in every frame [10]. The idea of items or identities was the starting point for connecting different gadgets. It is possible to remotely control and monitor these devices from a computer via the Internet [11]. Internet and Things are the two most influential terms in the Internet of Things (IoT). The Internet is a vast network that connects devices and servers. With the help of the Internet, a wide variety of devices are able to communicate with one another and share data [12]. A plethora of transformer versions used to computer vision have recently emerged, with vision transformer (VIT) appearing as a particularly outstanding example. But even with these VIT variations, the computational cost is still somewhat significant [13]. There is an excessive amount of parameters, and there is insufficient integration of local and global context data. Rough edge segmentation and significant segmentation holes caused by shadow occlusion are problems that this does not help solve [14–15].

An article-wide framework for video tracking in IoT networks is suggested in an effort to address the aforementioned issues. The network architecture of Deeplabv3+ and the Swin transformer are combined in the hybrid model. Using spatial pyramid pooling, Deeplabv3+ builds on CNN. To extract features from high-resolution data, the architecture employs a Swin transformer as both an encoder and a decoder. Lastly, the encoder makes use of Swin-Conv-Dspp (SCD) to record information about features across scales and mitigate the detrimental impacts of light-induced high levels of interclass similarity and intraclass disagreement. To further investigate the spatial correlation between global and local features, enhance target edge transformer block (GLTB) module is incorporated prior to each visual upsampling. This module also captures local and global feature information. In order torecover the classification accuracy, CSA fine-tunes the proposed model.

Here is how the remainder of the paper is structured: In **Section 2**, the relevant literature is reviewed; in **Section 3**, the approach that was suggested is examined; in **Section 4**, the results are analyzed; and in Section 5, the conclusion is obtainable.

II. RELATED WORKS

For the purpose of detecting face spoofing, Reddy et al. [16] has provided a variety of ANN architectures, the majority of which heavily employ convolutional layers. We "train" a deep neural network with a large amount of labeled data, and then "teach" it to use that network specifically for an application-specific domain that has few training instances. A proper sequence for this would be "training" followed by "teaching." With this, we can reach our objective. The next step is to combine data from both domains to form training sets that will be used for the network distillation. When there is a dearth of training data for a certain application area in the Internet of Things (IoT), we "train" a deep neural network with a large amount of labeled data and then "teach" it the specifics of that field. If we do this, we can "train" it. The most common technical word to describe this process is "teaching" a deep neural network. There is room for comparison between the two fields. To begin, we must collect data that is particular to spoofing in order to train a discriminative deep neural network on a domain that is application-specific. Multiple experiments have demonstrated that the suggested method works best when coupled with anti-spoofing parameters.

An intelligent mobile surveillance robot using a combination of DL models and conventional algorithms has been shown by Medjdoubi et al., [17], and it is based on the ESP32-CAM microcontroller. While a convolutional neural network (CNN) and two preexisting DL models, ResNet and VGG, are used for feature extraction, the Haar-Cascade (HC) technique is used for face detection. Naive Bayes (NB) and K-nearest neighbors (KNN) are two separate algorithms that make the categorization. By obtaining accuracy rates of 92.00% on the LFW database, 94.00% on the AR database, and 96.00% on the ORL database, respectively, validation studies show that a composite model combining HC, VGG, and KNN is preferable. On top of that, it has a remarkable recognition accuracy rate of 99.00% on a proprietary database, and it responds in real-time, even sending out email alerts. The benefits of this ET monitoring system are low power consumption, mobility, ease of use from afar, and reasonable cost.

In their groundbreaking face recognition system, Ali et al. [18] combine VGGFace for feature extraction, Support Vector Machine (SVM) for efficient classification, and Multi-task Cascaded Convolutional Neural Networks (MTCNN) for accurate face identification. When it comes to monitoring attendance, the technology really shines in real-time tracking many faces in one picture. It is worth mentioning that the "VGGFace" model stands out from the rest, displaying outstanding accuracy and attaining a fantastic F-score of 95% when combined with SVM. The model's success in detecting face identities is attributed to its strong training on huge datasets, as this highlights. The research highlights the

effectiveness of the VGGFace model, particularly when used in conjunction with different classifiers; for example, SVM produces very high accuracy rates.

In order to make Human-Computer Interaction (HCI) more natural, as proposed by Biswas et al. [19], robots need to be able to comprehend their surroundings, with an emphasis on this facet of human behavior in particular. This chapter presents a model for Facial Expression Recognition using a CNN. The seven main human emotions—happy, sad, angry, neutral, surprised, disgusted, and afraid—were used to train the CNN model. Assigning each picture to one of seven distinct facial expression groups is the goal of this chapter. The FER2013 dataset, made available by Kaggle, was used to train, test, and verify this model. It operates in series, with the final perceptron layer adjusting the weights and exponent values with each iteration. In addition, a new method for removing backgrounds was used so that we wouldn't have to deal with any of the many issues that may arise from the camera's placement.

A new method for extracting and detecting human facial features was developed by Ponnurathinam et al. [20]. It involves the use of Stacked Auto Encoder (SAE), Artificial Feeding Bird (AFB), and Region Based Fully Convolutional Network (RFCN). The rescaling approach is used to normalize the dataset initially. Afterwards, the optimization approach employed to extract face features, and the R-FCN algorithm is employed for detection and classification. Both testing and training are conducted using the WIDER Face dataset. In comparison to the state-of-the-art algorithms, the suggested SAE-AFB-RFCN framework achieves better performance in experiments, as shown by the F1-score, recall, precision, and accuracy metrics.

III. PROPOSED MODEL

Heatmap of Input Dataset

In contrast to previous research, the suggested model's heatmap is immediately constructed using the anticipated landmarks using a formula. This allows for a substantial reduction in both the parameters and the computing complexity. The equation is also expressed as:

$$H(x,y) = \frac{1}{\sqrt{1+min_{(x_i^{'},y_i^{'})\in S_1}\left\|(x,y)-(x_i^{'},y_i^{'})\right\|}} \tag{1}$$

where $H(x,y)$ is the intensity of point (x,y). $(x_i^{'},y_i^{'})$denotes the location of S1's i-th landmark. By setting H(x,y) to 0.5 if the value is less than 0.5, we may prevent the CNN from ignoring features that are far from face landmarks. Then, as seen below, the heatmap is combined with the characteristics by element-wise multiplication.:

$$F_O = F_1 \otimes H \tag{2}$$

The output features are denoted by F_O, whereas the heatmap and features learnt by the backbone network are represented by H and FI, respectively. The output features incorporate both the geometry information of face landmarks and the appearance information of the input photos by merging the heatmap with the intermediate features. In addition, the heatmap may be used as a guide for the proposed model to better face tracking by reducing background interference.

Face Tracking using Deep Learning-Based Object Detection Method

The output features are denoted by F_O, whereas the heatmap and features learned by the backbone network are represented by H and FI, respectively. The output features incorporate both the geometry information and the appearance data of the input photos by merging the heatmap with the intermediate features.The paper presents the Swin transformer and describes the planned SCG-TransNet in detail in this part. Afterwards, SCG-TransNet's two crucial modules—SCD and GLTB with SPPS—incorporate face landmarks. In addition, the heatmap can be used as a guide for the proposed model to better face tracking by reducing background interference.

Overview

Our SCG-TransNet adheres to the paradigm and is a combination of Deeplabv3+ and Swin transformer. In the first stage of the encoder, we use the Swin transformer as the backbone network for feature extraction. In the last stage of the encoder, we add the SCD module. To improve features and address the issue of important pixel information loss due to direct high-multiple upsampling, the decoder uses FPN to fuse features of different resolutions generated by Stages 2 and Stages 3. Then, after SCD, the feature map is stacked on the channel, and the continuity of pixel information is effectively enhanced [21]. Further, to improve feature extraction, a module (NAM) attention mechanism is incorporated prior to SCD and the concatenation of shallow and deep features. This mechanism redistributes the weights of maps. Lastly, prior to each visual upsampling, a GLTB module is included.

Swin Transformer Based Encoder and Decoder

Primary components of the encoder include the SCD and the Swin transformer backbone network. To obtain hierarchical feature maps, one uses Swin Transformer, and to get multiscale contextual information, one uses SCD. Swin transformer blocks, FPN, and GLTB make up the bulk of the decoder. Fusing feature maps of varying depths is done using the FPN.

When working with feature maps, the GLTB is useful for capturing both the global and local semantic information. It is possible to describe this procedure as

$$e_i = Encoder_{swim-Trans}(Image) \tag{3}$$

$$d_i = Decoder_{swim-Trans}(e_i) \tag{4}$$

Backbone networks for Swin transformers revolve on the Swin transformer block. Conventional VIT on the arena has quadratic computational complexity. Liu et al. developed the Swin transformer to lessen the computational burden. The transformer's multi-head self-attention (MSA) module is swapped out with a shift-window-based one between each set of self-attention layers. An improved method for context information is to progressively combine the window-based multi-head self-attention (W-MSA) block with a shifted window-based multi-head self-attention (SW-MSA) block.

The transformer block is connected in series with a W-MSA and a SW-MSA module under this shifted window partitioning arrangement. A W-MSA block is the first Swin transformer block. A residual link is established to acquire x Б^l from the input feature x^(l-1) after it goes through the LayerNorm and W-MSA layers. Subsequently, x Бl. is obtained by re-establishing a residual link after passing through the LayerNorm and multi-layer perceptron (MLP) layers. In comparison to the W-MSA layer, the SW-MSA block's window size offset is half that, and otherwise, the two structures are structurally identical. It is possible to describe this procedure as

$$\hat{x}^l = W_{MSA}\big(LN(x^{l-1})\big) + x^{l-1} \tag{5}$$

$$x^l = MLP\big(LN(\hat{x}^l)\big) + x^l \tag{6}$$

$$\hat{x}^{l+1} = SW_{MSA}\big(LN(\hat{x}^l)\big) + x^l \tag{7}$$

$$x^{l+1} = MLP\big(LN(\hat{x}^{l+1})\big) + x^{l+1} \tag{8}$$

An advantage of the Swin transformer over a CNN-based backbone network is its sequence-to-sequence paradigm, which facilitates the integration of multimodal input. It overcomes the shortcomings of conventional CNN-based models with its attention-based long-range modeling capacity. Since the Swin transformer is free of inductive biases, it accurately captures spatial dependencies in images that span large distances. Second, the Swin transformer has a reduced computing complexity and faster recognition and reasoning speeds as compared to other backbone networks that use transformers.

Swin-Conv-Dspp

It is not possible to capture scales using atrous convolution since it readily results in the loss of continuous space-time information. As a solution, ASPP in Deeplabv3+ employs a series of parallel atrous convolutional layers that use varying sampling rates to gather data from objects of varying sizes. In addition, expanding the receptive field during feature extraction network building is an effective strategy for minimizing data loss. Light intensity and incidence angle are two examples of the many sources of noise that can be found in RS photographs. The challenge of semantic segmentation of RS city sceneries now lies in how to mitigate these noises to an acceptable level. Feature point extraction is a sparse sampling technique that uses atrous convolution to extract information across pixels. This is not helpful for noise suppression since it will cause pixel information loss, which in turn causes long-distance convolution results to be uncorrelated. As a result, holes will emerge, and it will become more difficult to distinguish objects with excessively high levels of interclass similarity or intraclass variances caused by changes in light incoming angle and intensity.

Since atrous convolution information loses important details, the study combined CNN and Swin transformer characteristics to create a dual- layer. This layer uses Swin transformer's strong global context information extraction aptitude to compensate, and it strengthens the ability to extract global context feature information to help ASPP capture the long-range dependence of semantic information, which is difficult for ASPP to do on its own. In general, atrous convolution excels at extracting features' high-frequency information because it is basically a superposition of several high-pass filters, which continuously improves high-frequency information. The transformer is frequently more effective in extracting low-frequency information of features because, like a lowpass filter [22], it continuously enhances the underlying semantic information of the picture. We can successfully suppress the negative impacts of various sounds in RS pictures by combining the two benefits, which are to lower the disparities within classes and enhance the frequency of info across classes. The hole phenomenon is reduced, and the difficulty in distinguishing between classes as a result of high levels of interclass similarity or severe intraclass differences is improved.

In particular, SCD possesses a convolution branch and a Swin transformer branch. To better extract semantic information from patches with varied distances and capture multiscale information, shiftable windows of different widths are employed. Generally speaking, bigger windows try to collect global contextual information, whilst smaller ones try to

catch local information. By increasing the convolution's receptive field, the convolution branch is able to extensively extract objects of varying sizes using rates. In order to retrieve feature information from a variety of scales, it is recommended to increase the receptive field while simultaneously decreasing the loss of information. By integrating convolution's powerful local feature extraction capabilities with the transformer's exceptional dependency capture abilities, SCD demonstrates remarkable anti-noise performance. This effectively resolves the issue of holes caused by classes being too similar.

Global-Local Transformer Block

There are primarily two parts to the planned GLTB: First, the branch dealing with the worldwide context; second, the branch dealing with the local context.

First, we employ typical 1 × 1 difficulty to enlarge the channel input 2-D feature in the global branch, which is chiefly caught by window based multihead autonomous attention. $map \in R^{\mathrm{B\times C\times H\times W}}$ by a factor of 3. Next, the1-D arrangement$\in R(3 \times B \times H/W \times W/W \times h/h \times (w \times w) \times \frac{C}{h}$ isrehabilitated into Q, K, and V vectors using the window divisionoperation. The channel dimension is 64, the window size is 8, and the attention head is 8. The amount of processing is significantly enhanced when using shiftable window-based self-attention, yet it may capture feature information across windows. For this reason, we provide the cross-shaped window's module to merge the feature maps produced by the horizontal and vertical average pooling layers, allowing for more efficient capture of the global context.

Information of neighboringgrouping features derived from the same point in the original feature graph using convolution kernels of varying sizes and expansion rates is referred to as a cell in the feature graph outputted by SPPS. At the same spot in the initial feature map, each cell stores data collected using a kernel size that varies. It is possible to collect multireceptive field map point and use this data to offer multilevel information acquired from the same place. Using SPPS, we can get better at generating unique features and extracting important characteristics from the few viewable pixels of obscured objects. When the feature map is restored by upsampling, it further improves the capacity to extract local info and refines both the feature info. Additionally, the module is easily transferable to different models and may be used plug-and-play.

In addition, the report includes Algorithm 1 that thoroughly explains our suggested SCD-TransNet.

Algorithm 1: Training Process of SCG-TransNet

$Input$: $Vaihingen\ or\ Potsdam\ dataset\ D$;
1: $for\ epoch\ <\ epochs\ do$
2: $Extract\ features\ by\ (1)\ with\ SCD\ module$;
6: $end\ for$
$Output$: $Trained\ SCG-TransNet$;

Fine-tuning using Crocodile Search Algorithm (CSA)

No amount of iteration will get the fine-tuned correct function value closer to the constraint; in fact, it will be unable to do so. The Hunt Attack phase suggests a mutation strategy based on explosions to deal with this. The distance between the value of the need-mutation-fine-tuning goal function and the collection of extreme values in the shock cusp mutation bifurcation point distribution is taken into account by this technique. This method may greatly enhance the precision of the precise function value used for fine-tuning.

Shock Factor

It is challenging to correctly detect shock and mutation fine-tuning data in the search stage of Reptile Search Algorithm (RSA) due to the linear contraction of the evolution factor. In response to this, RSA implemented the shock factor ϕ. To improve RSA's capacity to detect tuning data, the algorithm's iteration speed, or the number of iterations, is adjusted. As an additional metric, the weight is utilized to ascertain the search velocity. The search speed may be determined by taking into account the shock factor current value of the fine-tuning objective function when there is a rapid change trend in the fine-tuning data, which indicates the potential prey.

$$\varphi = (\varphi_{max} - \varphi_{min})^{1-e^{1-t/T}} \times t(iter), else\ = (\varphi_{max} - \varphi_{min})^{1-e^{1-\delta/T}} \times t(iter), \widehat{F_x} > F_x \tag{9}$$

where φ_{max} and φ_{min} are the shockfactor, individually, δ is factor, $\widehat{F_x}$ isthe optimal value of the current fine-tuningobjective function, $F_x$is the greatest possible value for fine-tuning in the prior generation, and t(iter) is the distribution of degrees of freedom for T with respect to the sum of iterations of RSA. To improve the analysis of the shock data, the shock factor is gradually decreased during the initial stage of iteration as the number of iterations increases. The shock data is swiftly identified in the subsequent iteration stage by rapidly reducing the shock factor. It is possible to get alternative decline effects to match mutation data by evaluating the abrupt change trend of fine-tuning parameters and using those values to determine the shock factor's weight.

$$x_{i,j}(t+1) = Best_j(t) - \eta_{i,j}(t)\varphi \times \beta - R_{i,j}(t) \times r, t \leq \frac{T}{4} \quad (10)$$

$$x_{i,j}(t+1) = Best_j(t) \times x_{(r1,j)} \times ES(t) \times \varphi, \frac{T}{4} < t < \frac{T}{2} \quad (11)$$

where $x_{i,j}(t+1)$ is the function value of fine-tuning in the next iteration, $Best_j(t)$function of fine-tuning, $\eta_{i,j}(t)$determines the precise value of the related fine-tuning function, H controls the length of the search step for the objective function value, Ri,j(t) narrows the range of the precise value of the fine-tuning function, and ES(t) is objective function charge of the parameters used for fine-tuning.

Explosion Mutation

We suggest an explosive mutation technique to avoid RSA's local optimum and get a more precise objective function value of fine-tuning. The potential location of the $target$ $(prey)$, or the collection of target values associated with the development of fine-tuning—which might be one, continuous, or several times—is described as the cusp point. To determine the explosion mutation range when the current fine-tuning function value needs to be changed, we center the current function charge and use the distance between it and the global optimal fine-tuning objective function value as the radius. To create holes, which represent potential mutation positions in the objective function values of the fine-tuning parameters, we employ the extreme value probability delivery of the collection. An opening is described as a

$$SF_r = x_v + \{\sum_{k=0}^{\infty} P_k[G(t)]^k\} \times \sqrt{Best(t) + x_v^2} \quad (12)$$

where $r = 1, 2, \cdots, L, L$ is the sum of generated holes, SFr, xv is the impartial function value of fine-tuning that needs to achieve mutation,$\sum_{k=0}^{\infty} P_k[G(t)]^k$where x(r1,j) is the value of the function for random fine-tuning, Best(t) is the value of the global optimal accuracy function, and is the extreme cusp catastrophic bifurcation point set of the tuning parameters. The mutation impact is weak and the objective function value is inaccurate because to an inadequate number of CSA potholes. It is set at 50 because there are too many potholes to cause the calculation to increase. What follows is an enhanced crocodile algorithm hunting stage:

$$x_{i,j}(t+1) = \left[Best_j(t) \times P_{i,j}(t) \times \varphi\right] + \left[Best_j(t) - x_{i,j}(t)\right] \times SF, \frac{T}{2} < t \leq \frac{3T}{4} \quad (13)$$

$$x_{i,j}(t+1) = \left[Best_j(t) \times \eta_{i,j}(t) \times \epsilon \times \varphi - R_{i,j}(t) \times r\right] + \left[Best_j(t) - x_{i,j}(t)\right] \times SF, \frac{3T}{4} < t \leq T \quad (14)$$

$$\eta_{i,j}(t) = Best_j(t) \times P_{i,j} \quad (15)$$

$$R_{i,j}(t) = \frac{Best_j(t) - x_{(r_2,j)}}{Best_j(t) + \epsilon} \quad (16)$$

$$ES(t) = 2 \times r_3 \times \left(1 - \frac{1}{T}\right) \quad (17)$$

$$P_{i,j} = a + \frac{x_{i,j} - M(x_i)}{Best_j(t) \times \left(UB_{(j)} - LB_{(j)}\right) + \epsilon} \quad (18)$$

$$M(x_i) = \frac{1}{n}\sum_{j=1}^{n} x_{i,j} \quad (19)$$

In this context, Pi,j represents the distance between the optimal and objective values of the fine-tuning function, M(xi) is the average of the accurate values of the parameters controlling the fine-tuning function, α controls values of the parameters, t is the current iteration sum, T is the extreme iteration sum, r2 is a random value between -1 and N, and r3 is a value between -1 and 1.

CSA Optimization Steps

In order to solve the worth of the warning purpose under numerous constraints, the CSA method constantly adjusts the independent variables using population information, hence approximating the warning function. As the suggested model is fine-tuned, the CSA maximizes the goal function.

- Enter the parameters α and β, the beginning population, the extremesum of iterations, and the parameters for fine-tuning.
- A fine-tuning goal function and constraint conditions are created. The CSA optimization algorithm's limit approximation is used to solve the fine-tuning objective function.

Formula (20) is utilized to set the starting value of the goal function in the suggested model.

$$.x_{i,j} = r \times (UB - LB) + LB, j = 1,2, \ldots n \quad (20)$$

in which xi,j represents the potential objective function value in line i's column j, n stands for the objective function value's dimension, r is a random integer between 0 besides 1, and LB and UB denote the limits of the objective function value—respectively.

The data driven by mutation trends is used to regulate the search speed by adjusting the shock factor's weight. The explosion mutation is guided by a set of potential target values generated by bifurcation point set.
Find out if the value of the goal function satisfies the termination requirement. At the end of all the iterations, the optimal search outcome is the suggested model's accurate function value.

IV. EXPERIMENTS

Datasets

The WFLW [23] model has 98 landmarks and 10,000 faces, split evenly between training and testing. In addition, features such as big stance, expression, lighting, makeup, occlusion, and blur are annotated for each face.

Implementation Detail

Samples annotated with face landmarks are used during the initial step of training in the study. Setting the learning rate to 0.001 and reducing it by a factor of 0.03 per 4 epochs with a batch size of 128 is implemented. $Intel(R)\ Core\ (TM)\ i7-4510U\ CPU$ @ 2.00 GHz 2.60 GHz, 8.00 $GB\ RAM$, $Windows$ 10 OS, $Google\ Colab\ Setting-Python$ 3, Google Compute Engine backend, 1.13 GB RAM, and 26.26 GB disc space make up the primary machine.

Validation Analysis of Proposed Model

In assess the efficacy of our suggested perfect, the research employed performance measures with F1-score, recall, accuracy, besides precision. The accuracy with which the model classifies data points is the key performance indicator for these measures.

Table 1. Validation Analysis of Proposed Model on Different Ratios

Train/Test Split	Precision	F1-Score	Accuracy	Recall
80/20	98.92	96.52	99.98	94.24
70/30	98.10	97.20	99.97	95.20
60/40	98	97	99.96	95.10

In **Table 1** characterise that the Validation Analysis of Proposed model on different ratios. In the analysis ratio of 80/20 data split-up ratio, the accuracy as 99.98 and precision rate as 98.92 and then recall range of 94.24 and F1-score as 96.52 correspondingly. Then the 70/30 data split-up ratio, the accuracy as 99.97 and precision rate as 98.10 and then recall range of 95.20 and F1-score as 97.20 correspondingly. Then the 60/40 data split-up ratio, the accuracy as 99.96 and precision rate as 98 and then recall range of 95.10 and F1-score as 97 correspondingly. **Fig 1** shows the visual picture of projected model and **Fig 2** shows graphical description of various data splits on proposed model.

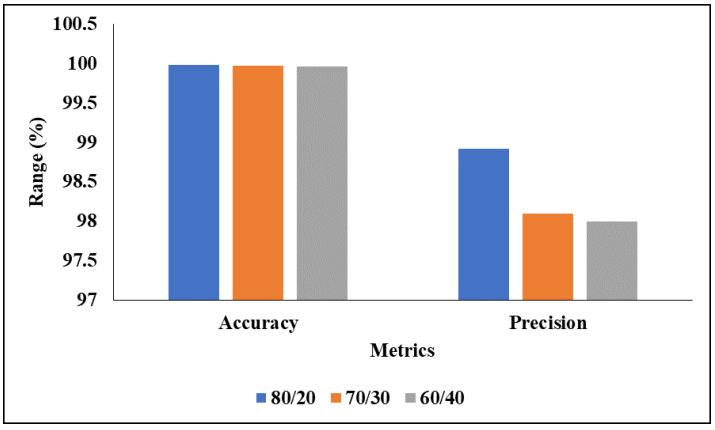

Fig 1. Visual Picture of Projected Model

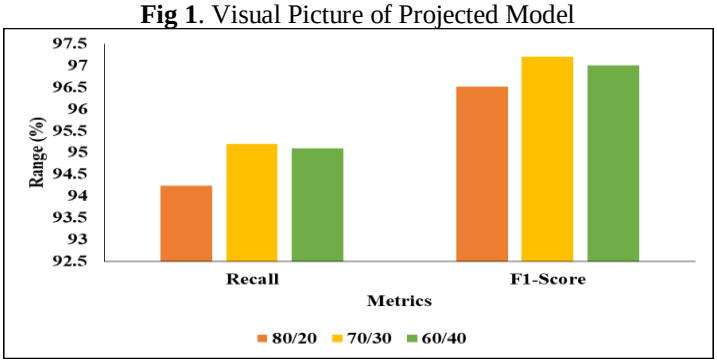

Fig 2. Graphical Description of various data splits on Proposed Model

Table 2. Performance of Proposed model on Timing Analysis

Train/Test Split	Prediction Time (s)	Training Time (s)
80/20	0.74	4.96
70/30	1.32	4.8
60/40	1.44	3.71

In **Table 2** signifies that the Presentation of Projected model on timing analysis. In the analysis of 80/20 data split-up ratio of Training Time as 4.96 and Prediction Time as 0.74 correspondingly. Then the 70/30 data split-up ratio of Training Time as 4.8 and Prediction Time as 1.32correspondingly. Then the 60/40 data split-up ratio of Training Time as 3.71 and Prediction Time as 1.44correspondingly.

V. CONCLUSION

The research suggests a swin transformer network that is better by integrating Deeplabv3+ with Swin transformer. The Swin transformer is able to better depict long-range dependencies than models based on CNN backbone networks since it does not have inductive bias. Swin transformer produces hierarchical feature maps with less computing cost and fewer parameters than competing transformers. Combining the powerful global context information capture capabilities of the Swin transformer with the extraction ability of convolution, the proposed model captures multiscale feature information. This allows for the acquisition of successfully inhibits noise caused by light-induced shadow occlusion. The SPPS can also make the most of the occluded object's incomplete pixels to produce distinguishable representation information, which helps with false or missed detection due to the target's occlusion and significantly enhances the model's edge localization ability. A CSA model, which enhances classification accuracy, carries out the fine-tuning procedure. Experiment findings show that heatmap geometric information can assist networks in remaining resilient under harsh environments. Another way that heatmaps might help with face tracking accuracy is by providing attention cues. The attention and feature sharing mechanisms taught by semi-supervised learning will be the subject of future research.

Data Availability

No data was used to support this study.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding

No funding agency is associated with this research.

Competing Interests

There are no competing interests.

References

[1]. X. Liu., “Collaborative Edge Computing With FPGA-Based CNN Accelerators for Energy-Efficient and Time-Aware Face Tracking System,” IEEE Transactions on Computational Social Systems, vol. 9, no. 1, pp. 252–266, Feb. 2022, doi: 10.1109/tcss.2021.3059318.

[2]. M. Kumar, K. S. Raju, D. Kumar, N. Goyal, S. Verma, and A. Singh, “An efficient framework using visual recognition for IoT based smart city surveillance,” Multimedia Tools and Applications, vol. 80, no. 20, pp. 31277–31295, Jan. 2021, doi: 10.1007/s11042-020-10471-x.

[3]. S. Jha, C. Seo, E. Yang, and G. P. Joshi, “Real time object detection and tracking system for video surveillance system,” Multimedia Tools and Applications, vol. 80, no. 3, pp. 3981–3996, Sep. 2020, doi: 10.1007/s11042-020-09749-x.

[4]. A. K. Biswal, D. Singh, B. K. Pattanayak, D. Samanta, and M.-H. Yang, “IoT-Based Smart Alert System for Drowsy Driver Detection,” Wireless Communications and Mobile Computing, vol. 2021, pp. 1–13, Mar. 2021, doi: 10.1155/2021/6627217.

[5]. S. Meivel et al., “Mask Detection and Social Distance Identification Using Internet of Things and Faster R-CNN Algorithm,” Computational Intelligence and Neuroscience, vol. 2022, pp. 1–13, Feb. 2022, doi: 10.1155/2022/2103975.

[6]. M. F. Alotaibi, M. Omri, S. Abdel-Khalek, E. Khalil, and R. F. Mansour, “Computational Intelligence-Based Harmony Search Algorithm for Real-Time Object Detection and Tracking in Video Surveillance Systems,” Mathematics, vol. 10, no. 5, p. 733, Feb. 2022, doi: 10.3390/math10050733.

[7]. T. A. Kumar, R. Rajmohan, M. Pavithra, S. A. Ajagbe, R. Hodhod, and T. Gaber, “Automatic Face Mask Detection System in Public Transportation in Smart Cities Using IoT and Deep Learning,” Electronics, vol. 11, no. 6, p. 904, Mar. 2022, doi: 10.3390/electronics11060904.

[8]. S. Liu, X. Liu, S. Wang, and K. Muhammad, “Fuzzy-aided solution for out-of-view challenge in visual tracking under IoT-assisted complex environment,” Neural Computing and Applications, vol. 33, no. 4, pp. 1055–1065, May 2020, doi: 10.1007/s00521-020-05021-3.

[9]. B. Varshini, H. Yogesh, S. D. Pasha, M. Suhail, V. Madhumitha, and A. Sasi, “IoT-Enabled smart doors for monitoring body temperature and face mask detection,” Global Transitions Proceedings, vol. 2, no. 2, pp. 246–254, Nov. 2021, doi: 10.1016/j.gltp.2021.08.071.

[10]. M. Geetha, R. S. Latha, S. K. Nivetha, S. Hariprasath, S. Gowtham, and C. S. Deepak, “Design of face detection and recognition system to monitor students during online examinations using Machine Learning algorithms,” 2021 International Conference on Computer Communication and Informatics (ICCCI), Jan. 2021, doi: 10.1109/iccci50826.2021.9402553.

[11]. X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, "Deep-Learning-Enhanced Multitarget Detection for End–Edge–Cloud Surveillance in Smart IoT," IEEE Internet of Things Journal, vol. 8, no. 16, pp. 12588–12596, Aug. 2021, doi: 10.1109/jiot.2021.3077449.

[12]. A. F. Klaib, N. O. Alsrehin, W. Y. Melhem, H. O. Bashtawi, and A. A. Magableh, "Eye tracking algorithms, techniques, tools, and applications with an emphasis on machine learning and Internet of Things technologies," Expert Systems with Applications, vol. 166, p. 114037, Mar. 2021, doi: 10.1016/j.eswa.2020.114037.

[13]. R. Ullah et al., "A Real-Time Framework for Human Face Detection and Recognition in CCTV Images," Mathematical Problems in Engineering, vol. 2022, pp. 1–12, Mar. 2022, doi: 10.1155/2022/3276704.

[14]. M. K. Hasan, Md. S. Ahsan, Abdullah-Al-Mamun, S. H. S. Newaz, and G. M. Lee, "Human Face Detection Techniques: A Comprehensive Review and Future Research Directions," Electronics, vol. 10, no. 19, p. 2354, Sep. 2021, doi: 10.3390/electronics10192354.

[15]. M. B. Satrio, A. G. Putrada, and M. Abdurohman, "Evaluation of Face Detection and Recognition Methods in Smart Mirror Implementation," Lecture Notes in Networks and Systems, pp. 449–457, Sep. 2021, doi: 10.1007/978-981-16-2380-6_39.

[16]. B. B. . Reddy, "Classification Approach for Face Spoof Detection in Artificial Neural Network Based on IoT Concepts", Int J Intell Syst Appl Eng, vol. 12, no. 13s, pp. 79–91, Jan. 2024.

[17]. A. Medjdoubi, M. Meddeber, and K. Yahyaoui, "Smart City Surveillance: Edge Technology Face Recognition Robot Deep Learning Based," International Journal of Engineering, vol. 37, no. 1, pp. 25–36, 2024, doi: 10.5829/ije.2024.37.01a.03.

[18]. M. Ali, A. Diwan, and D. Kumar, "Attendance System Optimization through Deep Learning Face Recognition," International Journal of Computing and Digital Systems, vol. 15, no. 1, pp. 1527–1540, Apr. 2024, doi: 10.12785/ijcds/1501108.

[19]. S. Biswas, T. Saha, P. Banerjee, and S. Datta, "A Novel Facial Emotion Recognition Technique using Convolution Neural Network," Heterogenous Computational Intelligence in Internet of Things, pp. 175–195, Sep. 2023, doi: 10.1201/9781003363606-12.

[20]. Jayabharathi Ponnurathinam and Sripriya Pradabadattan, "A Novel Approach for Human Face Extraction and Detection using SAE-AFB-RFCN Framework," Journal of Advanced Research in Applied Sciences and Engineering Technology, vol. 34, no. 1, pp. 51–62, Nov. 2023, doi: 10.37934/araset.34.1.5162.

[21]. M. D. R, A. Thirumalraj, and R. T, "An Improved ARO Model for Task Offloading in Vehicular Cloud Computing in VANET," Aug. 2023, doi: 10.21203/rs.3.rs-3291507/v1.

[22]. A. Thirumalraj, A. K, R. V, and P. K. Balasubramanian, "Designing a Modified Grey Wolf Optimizer Based Cyclegan Model for Eeg Mi Classification in Bci," 2023, doi: 10.2139/ssrn.4642989.

[23]. W. Wu, C. Qian, S. Yang, Q. Wang, Y. Cai, and Q. Zhou, "Look at Boundary: A Boundary-Aware Face Alignment Algorithm," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Jun. 2018, doi: 10.1109/cvpr.2018.00227.