Deep Learning Based Facial Recognition with Quadcopter

Subhashree panda Sonalika Sahoo Barenya Nayak

Debajyoty Banik

School of Computer Engineering, Kalinga Institute of Industrial Technology, Bhubaneswar, India

{21052539, 21052534, 21052577, debajyoty.banikfcs}@kiit.ac.in

**I. Abstract**

This work depicts the design and improvement of a drone which is made to connect to a remote computer and through this setup we have tried to achieve face recognition of known faces to serve society up to some extent by fighting against security threats. This is achieved by capturing video and photos from the drone and passing it through the written code which results in activating the webcam of the laptop for recognition of the face. We have achieved this by using concepts of deep learning and with the help of Opencv,dlib,cMake, and Numpy packages. The model that supports our code is the CNN model. Many approaches were made earlier also to make this similar kind of project but we have tried to reach a different domain in our method the drone can be used as a decoy in revealing an unknown person's identity as it is highly mobile and can be controlled by skilled hands.

**II. Basic Concept/ Technology Used**

The software uses deep learning algorithms to compare a live photo or video to the stored faceprint to verify an individual's identity.

In our paper, we have made improvements to add to this excellent software. Our proposed work consists of a drone fully equipped camera and wifi. The camera is well-defined to record people’s faces with ease and great precision. The wifi used connects the drone to our remote computer. The data received from the drone is passed on simultaneously to a well-defined face recognition module which is developed using python. In computer vision,one essential problem we are trying to figure out is to automatically detect objects in an image without human intervention. Face detection can be thought of as a problem where we detect human faces in an image. There may be slight differences in the faces of humans but overall,it is safe to say that there are certain features that are associated with all  human faces.

Face detection is usually the first step towards many face-related technologies,such as face recognition or verification. However, face detection can have very useful applications. The most successful application of face detection would probably be photo taking. When we take a photo of our friends,the face detection algorithm built into our digital camera detects where the faces are and adjusts the focus accordingly.

Face recognition is a method of identifying or verifying the identity of an individual using their face. Various algorithms can do face recognition but their accuracy might vary. Here we are going to describe how we do face recognition using deep learning.

So now let us understand how we recognise faces. We make use of face embedding in which each face is converted into a vector and this technique is called deep metric learning. Let me further divide this process into three simple steps for easy understanding:

The very first task we perform is detecting faces in the image or video stream. Now that we know the exact location/coordinates of the face, we extract this face for further processing ahead.

Now that we have cropped the face out of the image, we extract features from it. Here we are going to use face embeddings to extract the features of the face. A neural network takes an image of the person’s face as input and outputs a vector which represents the most important features of a face.

Our project is beneficial for the society in many ways such as for security purposes and can further be modified to deal with some real-life problems.

**III. Literature review**

In this paper,We have used a wifi camera drone to capture faces so that they can be passed through our code to obtain the desired output whether it is a known person or not. We have also made a folder which contains the images of people which we want our drone to capture and recognize.

Many have tried and implemented face recognition projects but our approach differs from others. In our method,the drone can be used as a decoy in revealing an unknown person's identity as it is highly mobile and can be controlled by skilled hands. According to the authors [6],our method can find applications in areas where there is a need for high security like banks. military in any patrolling activities and also in detecting a criminal from a large mass of crowded people since the criminal’s identity has been taken into consideration. This makes our project unique in terms of great mobility,precision and applications. Our face recognition python code connected to a wifi camera drone and when it was operated in the windows and captured the images when the drone was flying in the air nearly about forty meters in height,and then we passed the video in the code to get the desired output.

After passing the captured video in the code,when we ran the code then the output window was activated and it found a match so what we did was,we tried to portray it with a rectangle or a square or that green color line that we see and then it wrote the name,and this happened dynamically and it ran throughout the video and every character that was added went through the same cycle and this kept repeating until there were no frames left and if it didn’t found a match it showed ‘unknown’ at the output window.

We plan to apply more modifications to our project by which normal people can be benefitted like drone delivery and home security etc. We

can also fit our drone with instruments to neutralize enemies. Thus this project proves to be a great help in every field.

**IV. Input and output**

In this work,we propose a drone to fight against the security threats. Our drone is capable of collecting the video and passing the information to a remote computer through a wifi-capable device. Our remote device will identify and recognise the person accordingly.

**V. Proposed model**

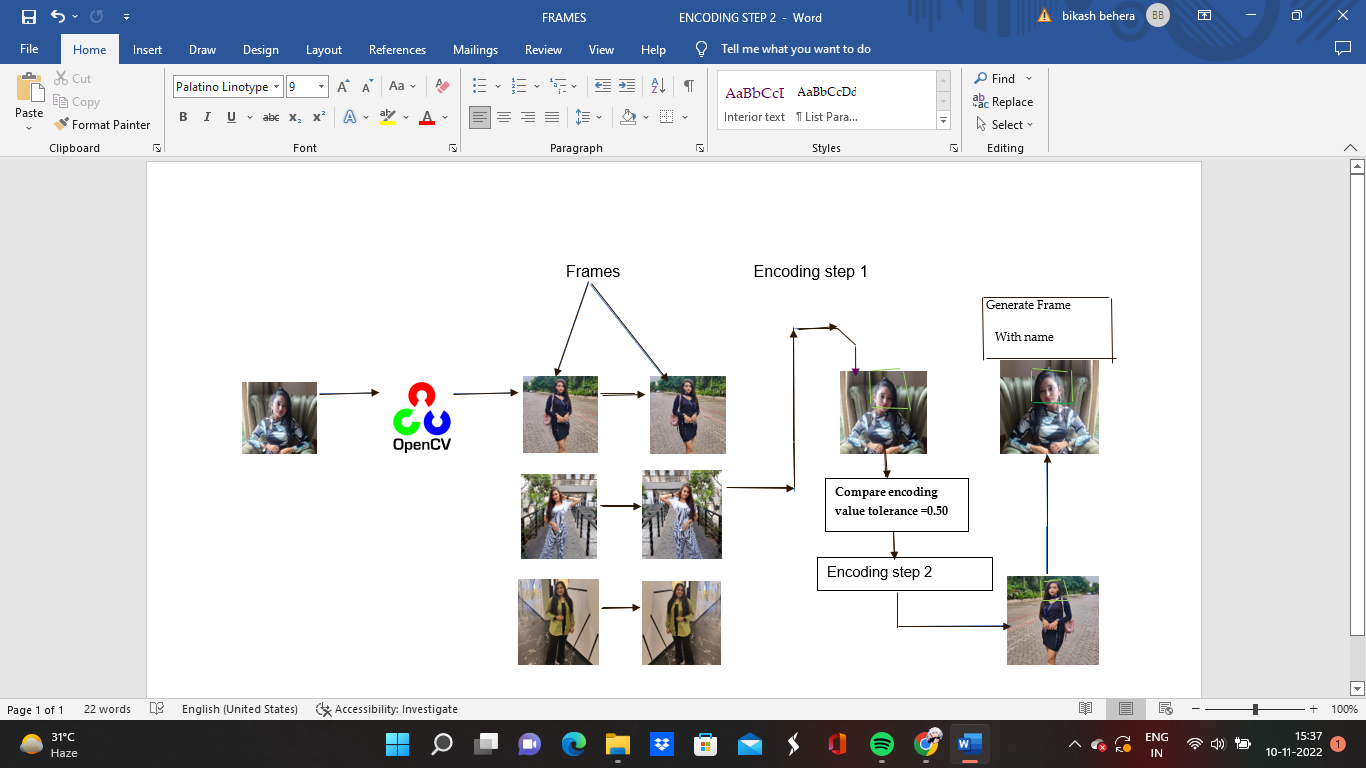


Fig.1: Overview of the Working Model

While streaming a live cam we may know the character in the frames but our system or algorithm which works on deep learning and neural network,although it is very smart it does not know the person in the frames. So we take the images of the persons we want our code to be familiar with. In the third step,we feed these images to the facial recognition algorithm that uses the d-lib to encode these images. And using the d-lib facial recognition network we generate the output feature as a term i.e.,a set of 128 real-valued numbers that will be used to quantify the face.

The default value of tolerance level is 0.6 but if we go low the stricter the recognition gets and the process becomes a bit slow. Based on the posture,the face angle,and the expression of the face the encoding algorithm will generate facial points and compare them with the reference encoding values so that if any one of them matches with these encodings it will tell the faces are similar.

So here using open-cv and python we render the input video file with video capture and then we read and generate every frame until there are no frames left.OpenCV-Python makes use of Numpy,which is a highly optimized library for numerical operations which is the fundamental package for scientific computing in Python. It also uses CMake which is an open-source software tool for managing the build process of software. Dlib is one of the most powerful and easy-to-go *open-source library* consisting of a machine learning libraries and various tools for creating software. For each frame,we repeat the encoding process using face recognition and generate the facial points and then we take the encoded frame and compare it with the encoded frame that we have generated based on the desired tolerance value and if there is a match we create a frame and name below it. This happens dynamically and it runs throughout the video. Every character we add will go through the same cycle and this will keep on repeating until there are no frames left.

.

**VI. Implementation And Proposed methodology**

The quadcopter maintains stability,direction,and motion using four rotors;two rotate clockwise and the other two counter-clockwise. A mounted camera is set at the center to take photos and record videos.

The main principle behind this is Newton's Third Law of motion,which states that for every action there's an equal and opposite reaction. A drone’s propellers push air downwards. This causes an opposite reaction called thrust that pushes it upwards against gravity. When the propellers rotate (for example clockwise),the quadcopter will tend to rotate anti-clockwise. Rotational force is called torque. A drone solves this by driving two diagonal propellers clockwise and the other two anti-clockwise.

In our work according to paper no. 7. We have used a CNN model i.e. (a convolutional neural network). This is precisely what the hidden layers in a CNN do – find features in the image. The convolutional neural network can be broken down into two parts:

When we start streaming our cam the convolution layer works and Extracts features from the connected convolutional layers

Images are fed into the input layer in the form of numbers. These numerical values denote the intensity of pixels in the image. The neurons in the hidden layers apply a few mathematical operations to these values. To perform these mathematical operations,certain parameter values are randomly initialized. Post these mathematical operations at the hidden layer, the result is sent to the output layer which generates the final prediction.

By Loading the input in a variable (say X), We randomly initialize a filter matrix. f is filtered constantly. Images are convolved with the filter as mentioned in Equation 1.

Z1 = X \* f (1)

 By applying the Sigmoid activation function to the result We randomly initialize the weight and bias matrix as mentioned in Equation 2.

A = sigmoid(Z1) (2)

 Apply linear transformation on the values to obtain Z2 as in Equation 3.

Z2 = WT.A + b (3)

By applying the Sigmoid function on the data. This will be the final output as in Equation 4.

O = sigmoid(Z2) (4)

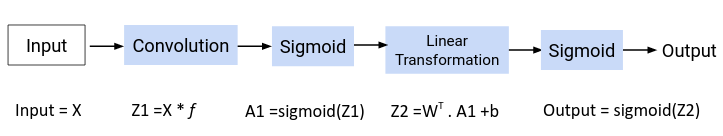


Fig 2: Full convolution process.

The CNN model treats these values as parameters, which are randomly initialized and learned during the training process. For the backward propagation convolution layer,we had the filter matrix as our parameter. During the forward propagation process,we randomly initialized the filter matrix. We will now update these values using the following equation:

new\_parameter = old\_parameter - (learning\_rate \* gradient\_of\_parameter)

∂E/∂f = ∂E/∂O.∂O/∂Z2.∂Z2/∂A1.∂A1/∂Z1.∂Z1/∂f (5)

We have already determined the values for ∂E/∂O and ∂O/∂Z2. Let us find the values for the remaining derivatives.

For finding the value for ∂Z2/∂A1, we need to have the equation for Z2 in terms of A1 as obtained in Equation 6.

Z2 = WT.A1 + b (6)

On differentiating the above Equation concerning A1, we get WT as the result as in Equation 7.

∂Z2/∂A1 = WT (7)

The next value that we need to determine is ∂A1/∂Z1. Have a look at the equation of A1 as obtained in Equation 8.

A1 = sigmoid(Z1) (8)

This is simply the Sigmoid function. The derivative of Sigmoid would be as mentioned in Equation 9.:

∂A1/∂Z1 = (A1)(1-A1) (9)

Finally, we need the value for ∂Z1/∂*f.* Here’s the equation for Z1 obtained in Equation 10.

Z1 = X \* f (10)

Differentiating Z concerning X will simply give us X as in Equation 11:

∂Z1/∂f = X (11)

Now that we have all the required values, let’s find the overall change in error concerning the filter as in Equation 12 :

∂E/∂f = ∂E/∂O.∂O/∂Z2.∂Z2/∂A1.∂A1/∂Z1 \* ∂Z1/∂f (12)

Notice that in the equation above,the value (∂E/∂O.∂O/∂Z2.∂Z2/∂A1.∂A1/∂Z) is convolved with ∂Z1/∂*f* instead of using a simple dot product. The main reason is that during forward propagation,we perform a convolution operation for the images and filters.

This is repeated in the backward propagation process. Once we have the value for ∂E/∂f, we will use this value to update the original filter value as mentioned in Equation 13:

f = f - LR\*(∂E/∂f) (13)

This completes the backpropagation section for convolutional neural networks.

We have implemented our project using python opencv and some other packages that are listed below:  
OpenCV-Python is a library of Python bindings designed to solve technical problems.

The complete setup is done by first connecting the different parts of the drone. It was then connected to windows through wifi and when the drone was switched on and made to fly in the air simultaneously videos and photos were captured and passed through the code for the recognition of known faces.

By using the facial recognition network we generate the output feature which is the list of 128 real-valued numbers. These points will collectively be used to recognize and differentiate one face from the other one. Also,these 128 points will help to determine the face in the different scenes based on tolerance value. If we are getting multiple matches for the same person it might be that people in your photo look very similar and a low tolerance value is needed to make face comparisons more effective or more strict. As we already know that the default tolerance value is 0.6 and the lower you go the stricter it gets but it will also affect the performance and it will be slower as well. So here we can see two images of a known person based on posture. the face angle,the expression of the face,the encoding algorithm will generate facial points and compare them with the reference encoding values. So let's suppose this is the reference encoding value that we have so when we generate facial recognition encoded value for this type of face it will compare both of them and based on the tolerance value that we have as an input in the encoding step 1 it will compare with the reference encoding value which was provided as the input. So as we said that we are very fortunate we are living in an age where the games we have right now with us are so close to real life now we have some idea of how these things work and export these things together so let’s talk about the implementation now.

Here comes the interesting part here. What happens is using OpenCV and python we render the input video file with video capture and then we read and generat every frame until there are no frames left. So this video file that you see here will consist of several frames. so when OpenCV reads this file it converts it into multiple frames and will keep on reading until there are no frames left with OpenCV video capture we generate one frame per loop so the cycle that you will see is a presentation of a single loop just for example we have shown a bunch of frames but actually in real time what happens is it generates one frame at a time. And for each frame that we generate using OpenCV we read the encoding process using face recognition like we generated the facial points. and then we take this encoded frame and compare it with the encoded frame created in step 1 based on the desired tolerance value. So the tolerance point is like 0.50 and if there is a match then we create a frame and name text so that we can show that as a part of the frame itself. so as we can see it found a match so what we did was we tried to portray it with a rectangle or a square or that green colour line that you see and then it wrote the name. And this happens dynamically and it runs throughout the video every character that you add will go through the same cycle and this will keep repeating until there are no frames left. So what we do is we take the video and convert that into frames using OpenCV and then we fill each frame to the face recognition algorithm. Then it converts into 128 real-valued numbers which is the encoding step that we have and then with the reference encoding step that we have already created before compares both the values based on the tolerance value then if there is a match it generates the frame. so that’s why you see the green boxes here. Once it generates it tells us yes we have found a match.

**VII. Dataset and results**

We have proposed this methodology to recognize facial characteristics on a dataset of 13233 images and videos as cited in Table 1. On the dataset this network compares to other states of the methods reaching 99.3% accuracy. Labeled faces in the wild (http://vis-www.cs.umass.edu/lfw/)

Table 1: Dataset statistics

|  |  |  |
| --- | --- | --- |
| Total | Images | Videos |
| No. of Images | 13233 | 15680 |
| No. of Persons | 5749 | 5876 |
| No. of Persons with two or more snaps. | 1680 | 1997 |

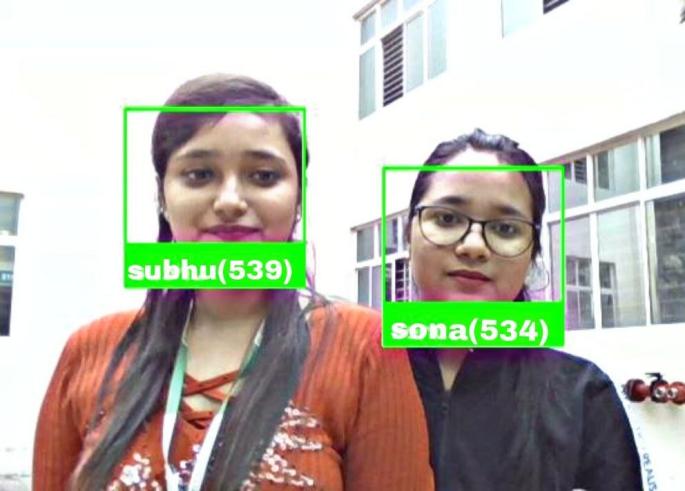
 

Fig 3: Proposed model case study captured by the shown quadcopter drone

As a case study,We drove the drone over the crowd. So,several people were captured by the drone and it identifies the faces very nicely as per expectation which can solve our problem expectedly which is shown in Fig. 3.

**VIII. Application for social impact**

This project paper proposes an idea to deal with some of the rea- life problems to become an asset for society and to make people’s life a bit easier.

If we place the drone at the main gate of a college and track the students incoming,the drone would recognize it as a known or unknown student which would be very helpful for security guards to decide whether the student belongs to the college or not. This can easily happen with a CCTV camera but the extra plus point with the drone is that the unknown person can escape the CCTV easily but the drone can be controlled with the remote and made to follow the person to capture him.

**IX. Conclusion**

From the above synopsis and after doing a lot of research we hereby conclude that we succeeded in making and developing a wifi camera drone to fight against the security threats of society by collecting video and photos by capturing them and passing them through the code to get the desired output.

Also, the drone is very helpful to mankind in many ways it can detect if a student belongs to our college or not by recognizing it as known or unknown apart from wearing identity cards. Also by making further improvements in our project we can make a move in sociological advancement by engaging in delivery services which will help in making delivery by knowing a person’s identity also there is 1,2 limitation of our drone according to paper no 8 like it can fly up to forty meters in height but that is not a major issue in our project. So,we consider this project as successful and hope to modify it in an advanced way in future.

**X. Future scope**

Our face recognition work is linked through wifi camera drone can be very beneficial to tackle some real-life problems if a robber breaks into a house and then tries to run and goes into a crowd full of people then we can easily capture and detect the unknown robber’s face by controlling the drone and through the captured path passed through the code instead of running behind the robber.

Secondly,we can detect if there’s any accident on a busy road and capture it so that it would help people to know so they can take an alternative path that would not lead to heavy traffic jams and save time.

Another fact is to use drones for security. With the appropriate license, operators can use unmanned aerial vehicles to provide security and surveillance to private companies, sporting events, public gatherings, and other venues. Drones can also gather valuable data during and after natural disasters to aid in security and recovery efforts.

**XI. References**

[1] Zhao, W., Chellappa, R., Phillips, P.J. and Rosenfeld, A., 2003.

Face recognition: A literature survey. *ACM computing surveys (CSUR)*, *35*(4), pp.399-458.

[2] Kamsani, M.N.L. and Mohd, M.N., 2021. Implementation of deep learning and motion control using drones. *Evolution in Electrical and Electronic Engineering*, *2*(1), pp.157-168.

[3] Suwarno, S. and Kevin, K., 2020. Analysis of face recognition algorithm: Dlib and OpenCV. *Journal of Informatics and Telecommunication Engineering*, *4*(1), pp.173-184.

[4] Valente, J. and Cardenas, A.A., 2017, November. Understanding security threats in consumer drones through the lens of the discovery quadcopter family. In *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy* (pp. 31-36).

[5] Hassaballah, M. and Aly, S., 2015. Face recognition: challenges, achievements and future directions. *IET Computer Vision*, *9*(4), pp.614-626.

[6] Merkert, R. and Bushell, J., 2020. Managing the drone revolution: A systematic literature review into the current use of airborne drones and future strategic directions for their effective control. *Journal of air transport management*, *89*, p.101929.

[7] Sharma, S., Shanmugasundaram, K. and Ramasamy, S.K., 2016, May. FAREC—CNN-based efficient face recognition technique using Dlib. In *2016 international conference on advanced communication control and computing technologies (ICACCCT)* (pp. 192-195). IEEE.

[8] Fantino, M., de Paula Albuquerque, O., De Albuquerque, A.P., Kelner, J. and Yankson, B., 2020, January. A Literature Survey on Smart Toy-related Children's Privacy Risks. In *HICSS* (pp. 1-10)