International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320-088X

IMPACT FACTOR: 7.056

IJCSMC, Vol. 11, Issue. 1, January 2022, pg.55 – 65

FACE MASK DETECTION USING PYTHON AND DEEP LEARNING

J. Tanvi; K. Sreeja; K. Likhitha

B.E. Scholars, Dept. of Electronics and Communications Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, Telangana

Dr. Vikram Palodiya

(Mentor), Associate Professor, Dept. of Electronics and Communications Engineering, Sreenidhi Institute of Science and Technology, Hyderabad, Telangana

DOI: 10.47760/ijcsmc.2022.v11i01.008

ABSTRACT

A software programme that can detect whether or not a person is wearing a mask and can be used with a CCTV camera. It's a computer vision technology that's commonly utilised in public places including theatres, malls, parking lots, and schools. Face has been used as a public and personal health indicator during the COVID-19 epidemic. To avoid the cause of covid-19, it is required to use a face mask. The camera can recognise whether a person is wearing a face mask or not using python and deep learning, and if he isn't, he won't be able to use public areas. As a result, the software will recognise whether or not the user is wearing a face mask, preventing the spread of covid-19.

Keywords: face mask detection, computer vision technology, python and deep learning.

INTRODUCTION

It uses technologies such as python and deep learning with some modules and libraries to recognise whether a person is wearing a face mask or not. In actuality, the problem is reverse engineering of face detection, in which the face is detected for security, authentication, and surveillance purposes utilising various deep learning algorithms. Face detection is a critical component in the field of computer vision technology.

Effective methods to contain the COVID-19 pandemic demand close attention to lessen adversely harmed communal health and the global economy, with the horizon still brim-full. In the absence of efficient antivirals and insufficient medical resources, WHO recommends a variety of measures to manage infection rates and prevent draining scarce medical resources.

One of the non-pharmaceutical strategies that can be used to minimise the main source of SARS-CoV2 droplets ejected by an infected individual is to wear a mask. Regardless of arguments over medical resources and mask types, all governments demand the usage of masks that cover the nose and mouth by the general public. In order to contribute to communal health, this research aims to build a highly accurate and real-time system for recognising non-mask faces in public spaces. As a result, it is necessary to wear a mask.

In this system, by using Python programming language and deep learning, the project has been implemented by using OpenCV which is an open-source library for working with image processing and performing computer vision tasks like face-detection, keras which is also an open-source software library which provides python interface for artificial neural networks and it also acts as an interface for the TensorFlow library and other python modules such as NumPy, matplotlib and SciPy.

It works magnificently in real time environment and doesn't need human interaction to find whether a person is wearing mask or not.

RELATED WORK

Pattern learning and object recognition are the inherent tasks that a computer vision (CV) technique must deal with. Object recognition encompasses both image classification and object detection. The task of recognizing the mask over the face in the public area can be achieved by deploying an efficient object recognition algorithm through surveillance devices. The object recognition pipeline consists of generating the region proposals followed by classification of each proposal into related class.

A face is detected from an image using numerous attributes in the face detection method. Facial detection research necessitates expression recognition, face tracking, and position estimation, according to. The objective is to recognise the face in a single photograph. Face detection is a tough task since faces fluctuate in size, shape, colour, and other characteristics throughout time.

It becomes a difficult task when an opaque image is obstructed by something not facing the camera, and so on. The authors believe that occlusive face detection has two key challenges: the first is the lack of large datasets containing both masked and unmasked faces, and the second is the removal of facial expression in the covered area.

Because of the distinctive shape of human faces, existing criteria for recognising the face with masks on it do not perform well. Face recognition is one of the most recent biometric technologies being researched, and it has a wide range of applications. Face identification, on the other hand, is one of the most difficult tasks in image processing.

Face detection's main goal is to determine if there is a face in an image and then to pinpoint the position of that face in the image. Face identification is clearly the first step toward developing an automated system that may include additional face processing. With a training set of faces and non-faces, the neural network is formed and trained. Python software is used to implement all of the results.

1. Muhammad et al presented MRGAN, an interactive technique. The method relies on the user providing the microphone area, which is then rebuilt using the Generative Adversarial Network.
2. Deep learning real-time face emotion classification and recognition was employed by Shaik et al. They classified seven face expressions using VGG-16. The suggested model was trained using the KDEF dataset and had an accuracy of 88 percent.
3. This is proposed system dataset which has been used was created by pyImageSearch reader PrajnaBhandary

PROBLEM DESCRIPTION

Due to the fact that most of the literature on covid-19 focuses on face creation and identification, wearing a face mask will not limit covid-19 trade, but it will help to reduce covid-19 transmission.

It is primarily concerned with detecting whether or not someone is wearing a mask on their face. If the person is not wearing the mask, it emits a beam warning sound, which allows us to detect mask identification and prevent the person from entering public areas.

OBJECTIVE

The main objective of the system is to detect whether the person is wearing the face mask or not, this is to prevent the spread of covid-19. The goal of this study is to identify those who are not wearing face masks in order to assist reduce COVID-19 transmission and spread.

Deep learning is a significant advancement in the science of artificial intelligence. It has recently demonstrated significant potential in image analysis for retrieving small characteristics. Due to the COVID-19 outbreak, some deep learning algorithms for detecting patients infected with the coronavirus have been proposed.

By using the python, deep learning and other modules and libraries it can implement face mask detection.

DATASET COLLECTION

The photographs have been used to train our deep learning architecture. CNN plays a significant role in the learning technique's architecture. The model is trained and tested using data from source. Only photos of faces are included in the dataset. It contains approximately 1,315 photos, including 658 images of people wearing face masks and 657 images of people without face masks. Eighty percent of each class's photographs are used for instruction, while the remaining images are used for testing.

ALGORITHM USED FOR FACE MASK DETECTION

For face mask detection the modules and libraries used are-

1. TensorFlow
2. Keras
3. Imutils
4. NumPy
5. OpenCV-python
6. Matplotlib
7. SciPy

OPENCV

OpenCV is an excellent tool for image processing and computer vision. It's an open-source library for tasks including face detection, objection tracking, landmark detection, and more. Python, Java, and C++ are among the languages supported. In this section, we'll utilise OpenCV to do real-time face detection from a live webcam broadcast. Videos, as you may know, are made up of frames, which are still images. Face detection is done for each frame in a movie. So, there isn't much of a difference between detecting a face in a still image and detecting a face in a real-time video stream.

To detect faces, we'll use the Haar Cascade method, also known as the Viola-Jones algorithm. It is a machine learning object detection algorithm that is used to recognise items in images and videos. We have numerous trained Haar Cascade models saved as XML files in OpenCV.

We use this file instead of starting from scratch and training the model. Now will use the "haarcascade_frontalface alt2.xml" file.

- Face Detection
- Geometric Transformations
- Image Thresholding
- Smoothing Images
- Canny Edge Detection
- Background Removals
- Image Segmentation

IMAGE PROCESSING

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it.

STEPS TO PERFORM IMAGE PROCESSING:

- Load images using Python.
- Convert images into array.
- And finally apply some algorithm on that array.

Fig1. Flowchart of face mask detection

TENSORFLOW

Google designed and distributed TensorFlow, a Python library for fast numerical computing. It's a foundation library that can be used to develop Deep Learning models directly or via wrapper libraries built on top of TensorFlow to make the process easier.

TensorFlow installation comes with number of deep learning models that can be used and can be implemented directly. TensorFlow is an end-to-end opensource platform for machine learning.

It has a large, flexible ecosystem of tools, libraries, and community resources that allow academics to advance the state-of-the-art in machine learning and developers to quickly construct and deploy ML applications.

KERAS

Keras is a Google-developed high-level deep learning API for implementing neural networks and it is built in Python and is used to make neural network implementation simple. It also allows for the computation of numerous neural networks in the backend.

Keras is simple to understand and use since it gives a high-level python frontend with the possibility of different backends for computation. Keras is slower than other deep learning frameworks because of this, yet it is incredibly user-friendly. Keras allows you to swap back and forth between various back ends.

Keras supports the following frameworks:

- TensorFlow
- Theano
- PlaidML
- MX Net
- CNTK (Microsoft Cognitive Toolkit)

SCIKIT-LEARN

Scikit-learn is Python's most usable and robust machine learning library. It supports Python numerical and scientific libraries like NumPy and SciPy, as well as techniques like support vector machine, random forests, and neighbours.

PROCEDURE

Now, by using the two-phase COVID-19 face mask detector in this system, as well as how our computer vision/deep learning pipeline will be built. The dataset we'll use to train our unique face mask detector will next be reviewed. Then, using Keras and TensorFlow, I'll teach you how to use a Python script to train a face mask detector on our dataset. We'll train a face mask detector with this Python script and then evaluate the results. With the trained COVID-19 face mask detector in place, we'll proceed to construct two more Python scripts to:

COVID-19 face masks can be detected in photographs.

Face masks can be detected in real-time video broadcasts.

Finally, it has a look at the outcomes of using our face mask detector.

BLOCK DIAGRAM

Fig.2. Phases and individual steps for building a COVID-19 face mask detector with computer vision and deep learning using Python, OpenCV, and TensorFlow/Keras.

Fig.3. Block Diagram of the proposed model

Now, just divide the system into two parts, each with its own set of sub-steps, in order to train a bespoke face mask detector:

Training: Now will focus on loading our face mask detection dataset from disc, training a model on it (using Keras/TensorFlow), and serialising the face mask detector to disc in this section.

Deployment: Then load the mask detector, run face detection, and categorise each face as with mask or without mask once the face mask detector has been trained.

In the rest of this tutorial, we'll go over each of these phases and their associated subsets in depth, but first, let's look at the dataset that has been using to train our COVID-19 face mask detector.

Fig.4. facemask detection dataset

To make the dataset, follow these steps:

Taking ordinary photographs of people's faces

After that, I created a custom computer vision Python script to add face masks to them, resulting in an artificial (but still useful) dataset.

Once you apply face landmarks to the problem, this strategy is actually a lot easier than it sounds.

The facial landmarks which is automatically deduce the position of face structures such as:

1. Eyes
2. Eyebrows
3. Nose

4. Mouth
5. Jawline

To generate a dataset of faces wearing face masks using facial landmarks, we must first start with an image of a person who is not wearing a face mask:

Fig.5. To build a COVID-19/Coronavirus pandemic face mask dataset, we'll first start with a photograph of someone not wearing a face.

The bounding box location of the face in the image is then computed using face detection:

Fig.6. The next step is to apply face detection. Here it is used as deep learning method to perform face detection with OpenCV.

Now it can extract the face Region of Interest (ROI) after it is known where the face is in the image:

Fig.7. The next step is to extract the face ROI with OpenCV and NumPy slicing.

Then use face landmarks to locate the eyes, nose, mouth, and other features:

Fig.8. Then, it detects facial landmarks using dlib so that it knows where to place a mask on the face.

Now the person's face is detected with facial landmarks i.e., eyes, nose, mouth and other features. After detecting those it is displayed that the person is not wearing the mask.

Fig.9. No mask

Fig.10. with mask

Let's look at how to use Keras and TensorFlow to train a classifier to automatically recognise whether someone is wearing a mask or not now that it's gone over our face mask dataset.

RESULT AND INFERENCES

As a result, it is a critical mechanism that is utilised all over the world to limit the spread of covid-19 in public spaces. It uses programming language Python and some modules and libraries such as Keras and OpenCV to create a deep learning architecture model for face mask identification in this proposed system.

The main aim of this project is to detect whether the person is wearing face mask or not by using the face mask detector model. By using Keras with network architecture is trained to the model. The first half of this project is to train the model, and the second part is to test it using a webcam and OpenCV. This is a great project for novices to apply what they've learned and acquire experience with.

Fig.11. Training and Loss Accuracy

BENEFITS

Officers find it difficult to check whether persons are wearing masks when doing manual monitoring. As a result, with our method, we use a webcam to detect people's faces and prevent virus transmission.

- It is quick and accurate.
- This technology can be used in ATMs, banks, and other financial institutions.
- We are able to keep individuals safe from our method.
- It emits a buzzer sound when the mask is worn.

CONCLUSION AND FUTURE SCOPE

This paper presents a deep learning-based solution for identifying masks over faces in public locations in order to reduce Coronavirus community spread. At the pre-processing level, the suggested technique efficiently tackles occlusions in dense settings by employing an ensemble of single and two-stage detectors. The ensemble approach not only aids in reaching high accuracy, but it also significantly improves detection speed.

Furthermore, the use of transfer learning on pre-trained models combined with considerable experimentation on an unbiased dataset resulted in a system that was both extremely reliable and low-cost. Face recognition, which violates mask conventions much more, boosts the system's utility for public benefits. Finally, the study points scholars in exciting new paths. To begin with, the suggested technique is not limited to mask detection and may be included into any high-resolution video surveillance device. Second, the model can be used with a facemask to detect facial landmarks for biometric reasons.

REFERENCES

- [1]. https://en.m.wikipedia.org/wiki/Face_masks_during_the_COVID-19_pandemic
- [2]. <https://data-flair.training/blogs/wp-content/uploads/sites/2/2020/07/face-mask-detector-project.gif>
- [3]. <https://www.ijert.org/real-time-face-mask-detection-and-recognition-using-python>
- [4]. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8223067/>
- [5]. <https://en.wikipedia.org/wiki/Keras>
- [6]. <https://www.ijert.org/real-time-face-mask-detection-and-recognition-using-python>
- [7]. <https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/>
- [8]. <https://www.analyticsvidhya.com/blog/2021/09/some-amazing-applications-of-opencv-library/>
- [9]. <https://machinelearningmastery.com/introduction-python-deep-learning-library-tensorflow/#:~:text=TensorFlow%20is%20a%20Python%20library,built%20on%20top%20of%20TensorFlow.>
- [10]. <https://www.tensorflow.org/>
- [11]. https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.researchgate.net%2Ffigure%2FFace-detection-and-recognition-flow-chart_fig6_338431245&psig=AOvVaw2UllJpW_8OaT8s6yboUGvI&ust=1642074672408000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCKDS6JOTrPUCFQAAAAAdAAAAABAD
- [12]. https://www.google.com/url?sa=i&url=https%3A%2F%2Flink.springer.com%2Fchapter%2F10.1007%2F978-3-030-73882-2_9&psig=AOvVaw2UllJpW_8OaT8s6yboUGvI&ust=1642074672408000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCKDS6JOTrPUCFQAAAAAdAAAAABAn
- [13]. https://www.google.com/url?sa=i&url=https%3A%2F%2Flink.springer.com%2Fchapter%2F10.1007%2F978-981-33-4236-1_16&psig=AOvVaw0S0CPzXJn7uJZjRclJfeFz&ust=1642074860163000&source=images&cd=vfe&ved=0CAsQjRxqFwoTCODp0vKTrPUCFQAAAAAdAAAAABAs
- [14]. https://www.researchgate.net/publication/355827770_A_Deep_Learning_Model_for_Face_Mask_Detection
- [15]. <https://www.dpublication.com/wp-content/uploads/2021/03/28-418.pdf>
- [16]. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8223067>