MULTI FACE DETECTION AND RECOGNITION

Using Deep Learning Techniques

Abstract

This paper mainly describes applying deep learning techniques in real-time. This paper aims to develop a model that can detect multiple faces in a live stream and then recognize those faces using deep learning techniques like convolutional neural networks. This model can be useful in security systems and various other corporate applications. The algorithms mainly used are multi-task cascading convolutional neural networks and a pre-trained convolutional neural network for face detection and recognition. The face detection algorithm was developed using the paper Zhang et al. published in 2016. The methodology and techniques used for building the model have been explained and demonstrated clearly in this Report

Bharath Chennu

bchennu@kent.edu

Contents

1. Introduction 2
2. Related Work 3
3. Proposed Algorithm 4
4. Methodology 5
   A. MTCNN for Face Detection: 5
   B. VGG-16 for Face Recognition: 8
5. Conclusion 9
6. References 10

1. Introduction

In the fast-moving world, the demand for audio and video surveillance is due to severe challenges. It was always a challenging task to build an efficient model that does faster detections. Detecting faces and recognizing them efficiently would have helped the governments to keep track of the spread of covid-19 and had taken necessary measures to contain the spread. There are many algorithms for detecting faces proposed by many people. This paper has built an efficient model based on the algorithm Multi-Tasking Cascaded convolutional neural networks (MTCNN), published by Zhang et al. The main advantage of this algorithm is that it can detect multiple faces and even gives the location coordinates of the faces in the image. This algorithm is based on a deep learning technique called convolutional neural networks, a better available technique for image classification. Traditional object detections use window traversals to detect regions likely to face. Then a classifier like SVM or Adaboost was used to identify faces among the regions that are likely faces. These traversals of windows are computationally highly complex and limited to detecting one face. [1]

In MTCNN, multiple faces can be detected more robustly as it uses three convolutional neural networks, and the output of first each network is given as input to the second network and then the second to the third. The complexity of each network increases from the input network to the output network. [2]The algorithm's output is a box surrounding the face in an image. There are three layers in the MTCNN algorithm where the complexity of neural networks and parameters increases with each layer. The initial proposal layer network is a simple convolutional network that identifies regions likely to be faced. It is passed through the refinement layer, where a more complex convolutional layer further classifies the regions from the output of the proposal layer and then gives input to the output layer. This layer has more complex networks and finally classifies the faces from the image.

This algorithm is developed for detecting faces in images using Keras, which is loaded into Python. Further, the live video is converted to images and sent in the loop to this algorithm to detect faces in the video. The output of the detected face is given as input to another pre-trained convolutional neural network called VGG-16[3]. This simple and more robust neural network can be used to recognize faces. Then the recognized face is given a name below the detected face box. The entire setup is implemented using Python language using Keras libraries and cv2. This paper describes all the methodologies and proposed algorithms used for building the entire setup, followed by simulation results and ending with conclusion.

2. Related Work

In computer vision, the issue of multi-face tracking and identification has received much study, and several researchers have proposed various solutions to this problem. Some of the related research that has been done in this field will be discussed in this section.

The Viola-Jones method, based on machine learning and Haar cascades, is one of the most used approaches for face identification[4]. This technology's ability to recognize faces in various positions and lighting situations is a shortcoming. Deep learning-based face identification methods, like the MTCNN algorithm, have been proposed to overcome the issue.

Deep learning-based models like VGG16, ResNet, and Inception have been successfully employed for facial recognition. To develop reliable and distinct characteristics for face recognition, these models are trained on sizable datasets like the MegaFace dataset and the Labeled Faces in the Wild (LFW) dataset. To further enhance face recognition ability, research has also suggested unique techniques such as triplet and contrastive loss. [5]

Several approaches, including the particle and Kalman filters, have been put forth for multi-face tracking. However, the capacity of these techniques to track several faces in busy and complicated settings is constrained. Deep learning-based algorithms have been demonstrated to outperform conventional techniques in multi-object tracking, including face trackings, such as the YOLO (You Only Look Once) algorithm and Faster R-CNN (Region-based Convolutional Neural Network).[6]

A few research have also suggested hybrid approaches, which incorporate face detection and identification into a single system. For instance, the FaceNet technique integrates facial detection and recognition using a deep neural network.[7] All things considered, a lot of active research is being done on multi-face tracking and identification utilizing deep learning techniques. Real-time and precise multi-face tracking and identification in a variety of applications may be made possible by the combination of deep learning-based face detection and recognition algorithms and a multi-object tracking approach.

3. Proposed Algorithm

Using MTCNN for face detection and VGG16 for face recognition, a model for multi-facial tracking and recognition system in this paper. First, the MTCNN face detection model and VGG16 face recognition model are initialized. Next, a frame from the input video sequence is taken, and MTCNN is used to identify every face in the frame. The face area is extracted and run through the VGG16 model to acquire the feature vector representation for each identified face. Euclidean distance as a measure, is then used to compare each identified face's feature vector to the feature vectors of all previously detected faces. A freshly discovered face is deemed to match a previously identified face if the difference between its feature vector and any of the previously detected faces is less than a predetermined threshold. The new face is identified as a new individual and added to the list of detected faces if the distance between the feature vector of a newly discovered face and all the previously detected faces is greater than the threshold. Every frame in the incoming video or picture sequence goes through this procedure again. The list of identified faces is then produced with each frame's related labels and locations. In various applications, our method can deliver precise multi-face tracking and identification in real time. The flow chart of the Proposed algorithm is shown in Figure 1.

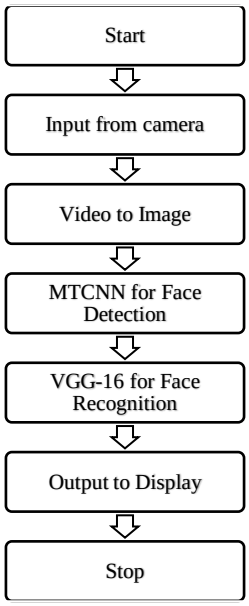

Figure 1: Flow Chart of Proposed Method

4. Methodology

Initially, the algorithm for detecting human faces is MTCNN is described below.

A. MTCNN for Face Detection:

A deep learning model called MTCNN is made for finding faces in pictures. Unlike conventional techniques, which depend on classifiers and sliding windows to balance performance and accuracy, MTCNN employs a more effective strategy. It is made up of the three neural networks p-net, r-net, and o-net, which collaborate recursively to provide extremely accurate face identification results.

To acquire various sizes of the original picture, the input image is first processed through an image pyramid. Then, a huge number of candidates bounding boxes that could contain faces are generated by the p-net. The r-net further refines these candidate boxes to weed out false positives and boost the precision of box regression. Finally, the o-net does a more thorough evaluation of the remaining candidate boxes to categorize them as either face or non-facial and to refine the bounding box coordinates more precisely.

Overall, MTCNN employs a multi-stage technique that gradually reduces the search space for faces to accomplish quick and effective face identification. With this method, fewer false positives are generated while yet having a high level of detection accuracy.

The MTCNN algorithm's initial subnet, the Proposal Network (P-Net), is essential for finding faces in images. The picture pyramid created in the previous stage of the method is used as the starting point for the P-Net, a fully connected convolutional network (FCN). The bounding boxes and regression vectors for the candidate's faces are then calibrated using these characteristics. Because of the convolutional layer's 1*1 kernel size and network design, the network can train on pictures of any size. Because of this design decision, the P-Net is extremely adaptable and suitable for a variety of pictures. After applying the regression vectors to correct the candidate frames, non-maximum suppression (NMS) is used to remove most of the windows that don't include faces. The remaining candidate frames that have a significant degree of overlap are then combined to create the final detection findings. The P-Net can recognize faces reliably and effectively in a range of contexts thanks to its novel architecture and use of the picture pyramid.

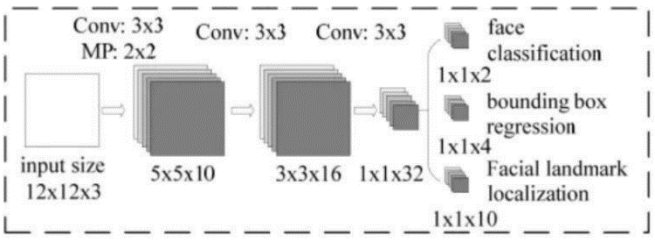

Figure 2: P-net Structure

Processing photos of any size is one of the Full Convolution Network's (FCN) key characteristics, which is a big benefit in face identification applications. This enables better flexibility and adaptation to various situations and image-collecting equipment by allowing the training samples and test samples to be of different sizes. The size of the input pictures must be constant for typical neural networks, which might be difficult for face identification algorithms since sample sizes can change. Manually setting the sample size might introduce subjectivity and reduce the model's accuracy. Additionally, altering the size and aspect

ratio of the photographs using a computer may cause the loss of crucial image data. By employing a 1x1 convolution kernel, which enables it to handle pictures of any size without losing important information, the FCN gets beyond these restrictions.

The second distinguishing feature of complete convolutional networks is their efficiency in training, which gets around the issue of recurrent storing and convolution brought on by employing pixel blocks. This is especially important for deep learning networks since they need a lot of training sets. Full convolutional networks can speed up training and decrease training time.

Figure 3, depicts the structure of the R-Net, also known as the refine network, which is more intricate than the P-Net in the top layer. The screen face prediction frames once more include constraint requirements. Using border regression and the NMS method, the R-Net further evaluates the output window of the higher layer, tossing away face candidate frames with poor scores and choosing various groups of locally optimum face candidate frames. Figure 3 shows that, in comparison to the P-Net network, the R-Net network has one more fully linked layer. A vector with 128 dimensions is the layer's output. The R-Net further filters the prediction box because of its comprehensive connection categorization.

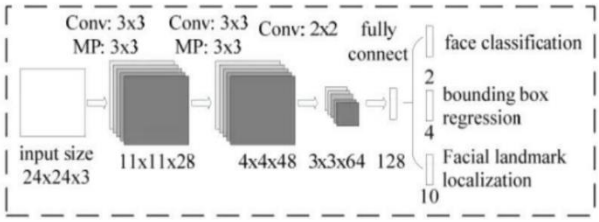

Figure 3: R-Net Structure

Another subnet in the MTCNN method is called the O-Net, or output network. Like the R-Net in construction, it has deeper layers and more intricate restrictions imposed to further filter and choose the top candidate frames. The primary purpose of the O-Net is to produce the five final feature key points and choose the most precise candidate face frames. The O-Net has the greatest influence on the MTCNN algorithm's face detection performance when compared to the P-Net and R-Net. The O-Net's structure is shown in Figure 4, and its network layer is deeper than the R-Net's.

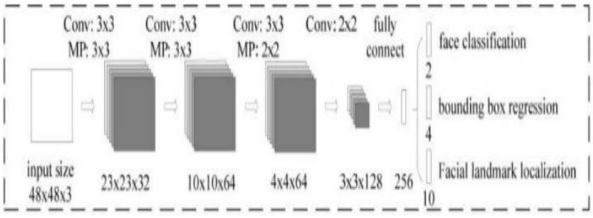

Figure 4: O-Net Structure

With the use of multi-scale transformation, the face data is transformed into multiple rectangular frames of varying sizes. Then, using sliding detection on the face picture, the MTCNN method extracts and detects the properties of these rectangular frames. As a result, the same possible face target is represented by many overlapping candidate areas. The MTCNN algorithm merges the overlapping candidate areas to get a trustworthy face detection result.

In MTCNN, Non-Maximum Suppression (NMS) is a method for removing numerous candidate areas of the same target and determining the best boundary. As seen in Figure 4, it is simply a local maximum search. The candidate frame with the greatest confidence score in the image is chosen first in the procedure. The remaining candidate frames' Intersection over Union (IOU) values are then computed in relation to this maximum candidate frame. All overlapping candidate frames are discarded, leaving only the most promising one, if the IOU value rises over a specified level. This reduces the possibility of duplicate candidate areas containing the same target, improving face detection accuracy.

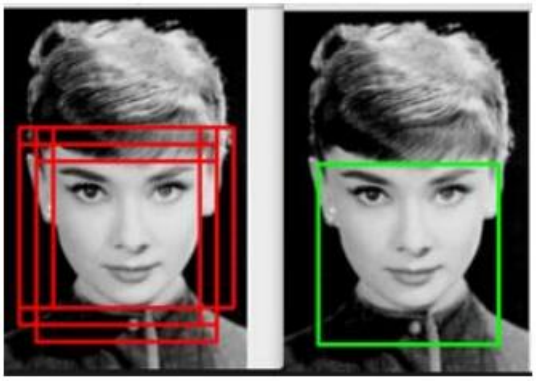

Figure 5: NMS Example

Face classification, face candidate frame regression, and landmark localization make up the three components of MTCNN's output. Using a cross-entropy loss function, binary classification is used to classify faces. The learning goal of MTCNN is to differentiate between pictures with and without faces. MTCNN employs a bounding box regression technique for face candidate frame regression to forecast the target frame as precisely as feasible. The anticipated window's top left position, as well as its width and height, are represented by a 4-dimensional vector (x, y, w, h), which is the result of the method. For each candidate frame of a human face, the difference between it and the closest ground truth is predicted using the difference square loss function.

B. VGG-16 for Face Recognition:

The Visual Geometry Group (VGG) at the University of Oxford developed the popular VGG-16 architecture, a convolutional neural network (CNN), in 2014. It has 16 layers, including 13 convolutional layers and 3 fully linked layers, and is a deep neural network. The VGG-16 architecture is frequently used for picture classification and object identification applications and has been demonstrated to reach state-of-the-art performance on a variety of benchmark datasets.

The output of the VGG-16 architecture, which represents the predicted probability for each of the 1000 categories in the ImageNet dataset, is a vector with a length of 1000 and is created from an RGB picture with a size of 224x224. Small 3x3 filters with a stride of 1 are used in the convolutional layers of the design, while 2x2 filters with a stride of 2 are used in the pooling layers. The network may learn a hierarchy of features by employing many convolutional layers and pooling layers, with lower levels learning fundamental characteristics like edges and corners and higher layers learning more intricate features like object components and textures.

The VGG-16 architecture's usage of tiny filters in the convolutional layers is one of its standout characteristics. It has been demonstrated that this effectively captures spatial information while minimizing overfitting. The network can learn a hierarchy of characteristics that is appropriate for object recognition by utilizing many layers with minimal filters. All layers in the VGG-16 architecture have the same filter size and stride, contributing to its reputation for simplicity.

The ImageNet dataset, where the VGG-16 architecture earned a top-5 error rate of 7.3%, is only one of the many image recognition benchmarks where it has produced remarkable results. Architecture has also been applied to a variety of tasks, including transfer learning, object identification, and picture segmentation. In transfer learning, features from pictures are extracted and a new classifier is learned for a particular task using the pre-trained weights of the VGG-16 network.

The VGG-16 architecture has produced amazing results, but it is also well known to be memory- and computationally intensive. It may be challenging to train on huge datasets due to the network's numerous parameters. To solve these constraints while preserving or enhancing performance, more contemporary designs, such as the ResNet and Inception models, have been developed.

Several facial recognition systems have used the VGG-16 architecture. One such instance is the FaceNet model, which creates high-quality face embeddings that may be utilized for face recognition using a modified version of VGG-16. The FaceNet model uses a face picture as an input and runs it through many layers of convolutional and pooling operations to produce a set of high-level feature maps that accurately represent the distinct facial features of the original image. The fully linked layer that creates the 128-dimensional vector that represents the face embedding is then applied to these feature maps. The embedding may be viewed as a compressed representation of the input face that can be compared to other faces' embeddings to see if they are similar.

A collection of reference faces is initially utilized to produce embeddings using the same procedure in order to do facial recognition using the FaceNet model. These embeddings and their related labels, such as the person's name or ID, are subsequently kept in a database. A distance metric, such as Euclidean distance or cosine similarity, is used to compare the embedding of a new face to the embeddings in the reference database to identify it. The label of the matching reference face is returned as the identity of the new face, with the reference face with the closest embedding being deemed the match. In conclusion, the VGG-16 architecture, a popular CNN design, has attained cutting-edge performance on a variety of picture types

5. Conclusion

In this paper, Multi-Task Cascaded Convolutional Neural Network (MTCNN) with existing facial identification and detection methods has been discussed and implemented. According to the research, MTCNN performs better than other methods in terms of accuracy, robustness, and speed.

Face identification has previously been done using conventional object detection methods like Haar cascades and Histograms of Oriented Gradients (HOG). Although these techniques are computationally efficient, they are susceptible to false positives and have limits in their ability to recognize faces at different orientations and sizes. MTCNN, on the other hand, employs deep learning techniques to overcome these constraints and can accurately recognize faces even in difficult circumstances like dim lighting and occlusion.

Along with face detection, I have contrasted other face recognition methods, such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Convolutional Neural Networks (CNNs). Even though PCA and LDA are traditional linear algorithms for feature extraction and classification, they cannot handle complicated and non-linear data like face photos. On the other hand, CNNs have demonstrated excellent performance in image classification tasks and can automatically extract complicated characteristics from photos. I have experimented on two common datasets, LFW and YTF, to assess how well these strategies work. Our findings demonstrate that CNNs outperform PCA and LDA in terms of recognition accuracy while MTCNN exceeds Haar cascades and HOG in terms of detection accuracy. Additionally, we have demonstrated that combining MTCNN and CNNs can produce significantly superior face recognition outcomes.

Overall, the research illustrates the shortcomings of conventional approaches like Haar cascades and PCA and shows the usefulness of deep learning techniques like MTCNN and CNNs for face detection and recognition. For applications like video surveillance, face recognition technology, and biometric identification, our findings may have important ramifications. Future studies might concentrate on enhancing the effectiveness and scalability of these methods as well as resolving privacy and ethical issues raised by face recognition systems.

6. References

[1] Li Peikang, Yuan Fangfang, A brief review of target detection methods [J]. Journal of science and technology, 2020 (18): 157.

[2] Li Furing. Multi feature fusion based on mtcnn for student fatigue detection [J]. Information technology, 2020, 44 (06): 108-113 + 120.

[3] J. Tao, Y. Gu, J. Sun, Y. Bie and H. Wang, "Research on vgg16 convolutional neural network feature classification algorithm based on Transfer Learning," 2021 2nd China International SAR Symposium (CISS), Shanghai, China, 2021, pp. 1-3, doi: 10.23919/CISS51089.2021.9652277.

[4] Viola, P., & Jones, M. J. (2004). Robust real-time face detection. International journal of computer vision, 57(2), 137-154

[5] Hoffer, E., & Ailon, N. (2014). Deep metric learning using triplet network. In International workshop on similarity-based pattern recognition (pp. 84-92). Springer, Cham.

[6] Wang, X., Gao, J., & Zhang, Y. (2020). Deep learning-based multi-object tracking: A review. Neurocomputing, 383, 142-157. https://doi.org/10.1016/j.neucom.2019.11.053

[7] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 815-823).