AI to Design a Mask Insensible to the Distance from Camera to the Sense Objects

Raghavendra B [1], V Siva Poojitha [2] ,Zeba Rehaman [3], Mala M V [4]

[1] Assistant professor, Department of Computer Science and Engineering, Nagarjuna College of Engineering and Technology, Bangalore, India

[2,3,4] B.E. Students, Department of Computer Science and Engineering, Nagarjuna College of Engineering and Technology, Bangalore, India

ABSTRACT

Face recognition has grown in popularity in image processing, and as a result, has made significant progress. A significant amount of new algorithm design is centred on the usage of convolutional structures to improve the algorithm's precision. For the first time, these convolutional structures enabled the extraction of even the most minor pixel information. Our objective is to construct a binary face classifier capable of identifying each face in the picture, independent of its alignment. We show how to efficiently create accurate face segmentation masks from any random input picture using the following method. Before performing the approach, the picture must be an RGB picture of any size, as previously stated. The following preconfigured VGG-16 training weights are used: https://deeplearning.net/software.html">https://deeplearning.net/software Convolutional Neural Networks, which utilise both supervised and unsupervised learning methodologies, are used to semantically separate the faces in the picture. Binomial cross-entropy is the loss function, while gradient descent is the training technique. Unwanted noise is removed from the image, and incorrect predictions are avoided, allowing the image to be classed in such a way that a bounding box around the faces is created. Furthermore, the proposed model has been found to be quite good at recognising non-facial faces. It's also capable of identifying many face masks inside a single picture. The Multi Parsing Human Dataset was used in the experiments, and the average pixel-level accuracy was 93.884 percent. Experiments were also carried out on segmented face masks, with a mean pixel-level accuracy of 93.884 percent.

Face Recognition, Image Processing, Convolutional Neural Networks, Semantic-Segmentation, Face-Segmentation, and Face Detection are some of the indexing terms.

INTRODUCTION

Face recognition has gained in relevance as a study subject in image processing and computer vision, making it an interesting topic. It offers a wide variety of applications, from facial motion capture to face recognition, despite the fact that it is still in its early phases. To begin, accuracy is essential. Because of its use in image-based applications such as real-time surveillance and video-based applications such as real-time face recognition in video surveillance, face recognition is more important than ever. We can now categorise images with great accuracy using convolutional networks. After face identification, pixel-level information is often required, which most face detection algorithms cannot provide. Semantic Segmentation has hit a snag due to the time and effort necessary to gather data down to the pixel level. Semantic segmentation is the process of assigning a name to each pixel in a picture. In our instance, the labels are either face-to-face or non-face-to-face. As a consequence, semantic Segmentation is used to distinguish between the face and the background by classifying each pixel in the image as a face or a backdrop. Furthermore, frontal face identification is prioritised in practically all commonly used face recognition systems.

This research develops a face recognition model by sorting each pixel in an image into two categories: those that belong to faces and those that don't. Members of the face class are regarded members of the class, but non-face pixels are not. The model works better with photographs that have frontal faces, but it may also be utilised with photos that don't have frontal faces. It also comes with the extra difficulty of coping with the inevitable blunders that will arise. A Convolutional Neural Network is used to finish the semantic segmentation of the human face. The next section looks at face detection research and how it pertains to this project. In Section III, you'll learn how to use semantic segmentation to do face segmentation and detection on any RGB image. Finally, laboratory tests are conducted to put the face masks that have been made to the test.

SYSTEM ANALYSIS

To create objectives, purposes, and systems, one must first understand the technique or business, and then learn how to create processes and procedures that allow those objectives to be met effectively.

EXISTING SYSTEM

At first, researchers concentrate on the edge of the grey value of the facial picture. It was based on a paradigm of pattern recognition.

1. The “Viola-Jones Detector's face detection technology”, which significantly enhanced “real-time face detection”.
2. The “Viola-Jones detector” was unable to address real-world issues and was impacted by various parameters such as “facial brightness and direction”.
3. “Viola-Jones” detect well-lit frontal faces and did not perform well in low-light conditions or with non-frontal photos.
4. These difficulties motivated separate researchers to build new face identification algorithms based on deep learning to provide more accurate findings for various facial circumstances.

PROPOSED SYSTEM

We recommend pursuing two objectives: developing a binary face classifier and keeping people safe.

1. To locate faces in any orientation, independent of their location or training, a network capable of learning and accurately recognizing them must be used.
2. To run the model, the user must first provide the model with an RGB image of any size (feature extraction and class prediction).
3. We are focusing on creating realistic face masks for human items by using RGB channel images that include items in separate places.
4. Additionally, erroneous forecasts have been addressed, with a box created around the segmented area to improve its accuracy.
5. To create our face recognition model, we used convolutional neural networks and the Multi Human Parsing Dataset to create a network that enables the system to recognize the face in any circumstance, regardless of whether the face is frontal or not.
6. Historically, CNNs have been primarily employed for image classification applications.

SYSTEM REQUIREMENTS AND SPECIFICATIONS

SOFTWARE REQUIREMENTS

- Platform
- Coding Language
- Client
- Operating system

HARDWARE REQUIREMENTS

- Ram 4GB
- System
- Hard Disk

SYSTEM DESIGN

The purpose of this project is to design a binary face classifier that can recognise faces in any orientation, independent of alignment, and to use a neural network to develop the classifier so that the findings are reliable. To utilise the model, you must first provide an RGB image of any size. The basic goal of the model is to extract characteristics and predict classes. The feature vector is optimised using gradient descent and gradient descent, using Binomial Cross-Entropy as the loss function.

PROPOSED WORKFLOW

We show how to generate segmentation masks straight from photographs with faces organised in various orientations and no evident noise. Each image of any size is downsized to 224 224 3 before being sent to the CNN network for feature extraction and prediction. After it has been transmitted over the network, the output is subjected to post-processing. When applying global thresholding on face and background pixel values, the initial operation is to threshold the face and background pixel values. After that, a median filter is applied to the original signal to remove high-frequency noise. Fill in the gaps in the segmented region using a closure operation, resulting in a bounding box around the segmented area as seen below.

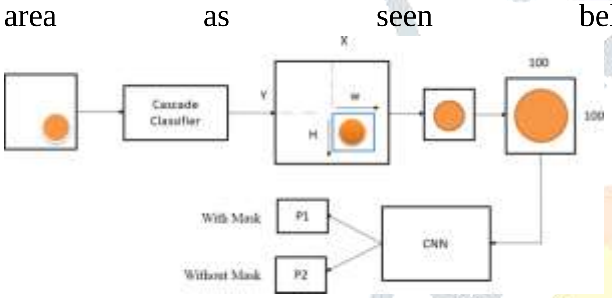

Fig 1. Workflow Overview

ARCHITECTURE

The specified training weights of the VGG-16 architecture are used to conduct feature extraction and prediction. The basic VGG-16 architecture is shown in Figure 5.1. There are 17 convolutional layers and 5 max-pooling layers in our model. When the model is given a picture with a starting size of 224x224x3, it begins to generate predicted pictures. As it passes through the various feature extraction layers, the picture is convolved and then processed via a series of convolutional and max-pooling layers. The picture being processed is used by the convolutional layer to convolute (also known as tiling) the picture. At the same time, the max-pooling process halves the size of the feature vector in each layer, making model parameter management easier. Reduce the amount of parameters in feature extraction because if the parameters are not reduced, it becomes much more difficult to forecast the classes of each pixel in a convolutional neural network. The complexity of segmentation required pixel-by-pixel storing of spatial information, which we accomplished by adding convolutional layers to the VGG layers. After the final max-pooling layer, expansion begins. This picture is sampled to the typical 224 224 2 sizes and the objects in the picture are classified as binary, resulting in the picture having two channels for both the face and background classes.

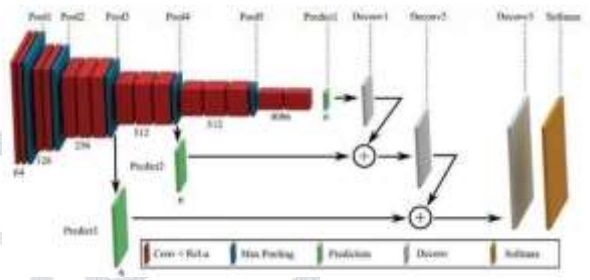

Fig 2: CNN Architecture

FACE DETECTION AND ERRONEOUS DETECTION

On fill in any anomalies in the area and erase mistakes made during image processing, post-processing is done to the predicted mask (which may have crept during the processing). Finally, we apply the Median filter on the mask before applying the Closing Operation to the final image.

This, together with the procedure before it, ensures that all holes in the segmented area are filled and the majority of inaccurate predictions are deleted. While certain critical issues may endure, their existence must be taken into account. When exhibiting the final recognised faces, we constructed the model such that all wrong predictions are eliminated. For each region, we calculate the following statistics: Center of Mass, Minor Axis Length, and Major Axis Length. This data is kept in a table (websites utilise segmented photos instead of simple image tiles) (including false predictions). The undesirable erroneous predictions persist even after the median filter and dilation have been employed to eliminate unnecessary post-processing. Inadvertently, this leads to erroneous facial detection.

TRAIN MASK DETECTOR

Teach the model to identify the mask by taking every picture and creating an image array, and then concatenating all of those arrays together.

Preprocessing processes are done to all the raw input photos to turn them into clean copies, which can then be supplied to a neural network machine learning model.

FACE MASK DETECTION IN WEBCAM STREAM

This article discusses how the flow of information works to determine whether or not the subject on the camera is wearing a face mask.

The technique is divided into two stages.

1. Attempting to recognize the faces captured on the camera.
2. Classify the faces according to the kind of face mask they wear.

IDENTIFY THE PERSON APPEARING ON THE WEBCAM

To identify the faces, we utilized a pre-trained model made accessible through the Open CV platform. To develop an accurate model, the dataset was first created using photographs accessible online. OpenCV provides two models for facial Detection:

1. A 16-bit floating-point implementation of C make.

2. TensorFlow was used to create an 8-bit quantized version. This face mask detector incorporates the Cmake model. The phrase "deep learning" refers to a large class of machine learning methods that use deep structural data and training samples. We chose to find out how to do this independently since this increased our prospects for an efficient answer. Collecting data to tackle the range of face masks used by workers entails determining which kind of face masks are used. The face mask detection model comprises two components: a face recognition model that uses camera feeds to identify existing faces and a mask identification model that runs those faces through facial feature detectors.

DETECT MASK VIDEO

Leveraging a model in a camera for face detection is feasible by identifying faces with the face detector model and identifying faces with or without masks using the mask detector model. Additionally, we employ OpenCV for camera operations.

OUTPUT

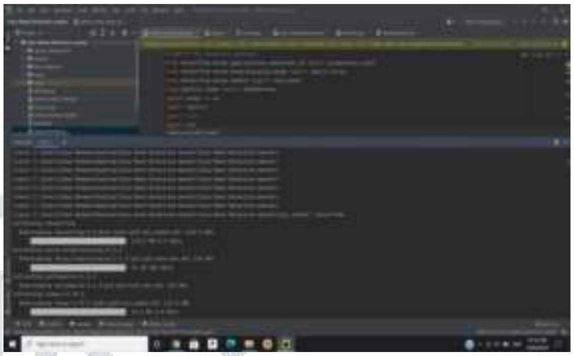

Fig 3:- Install Tensorflow

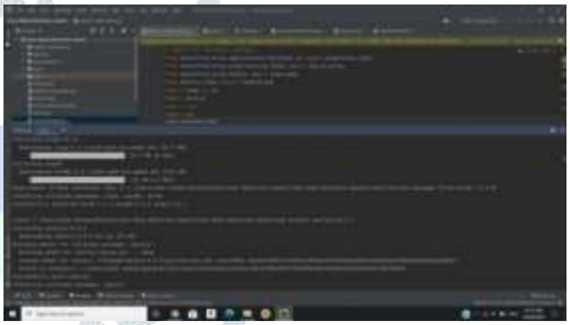

Fig 4:- Install imutils

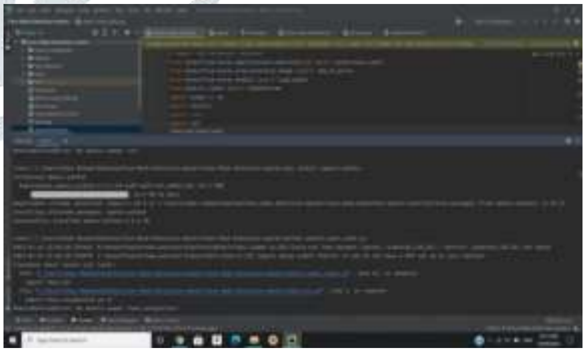

Fig 5:- Install open CV

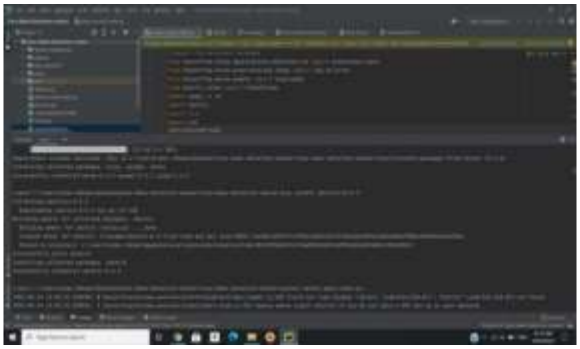

Fig 6:- Cam initialization

REFERENCES

[1] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, 1097-1105.

[2] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

[3] Iliadis, M., Spinoulas, L., & Katsaggelos, A. K. (2018). Deep fully-connected networks for video compressive sensing. *Digital Signal Processing*, *72*, 9-18.

[4] Li, J., Zhao, J., Wei, Y., Lang, C., Li, Y., Sim, T., ... & Feng, J. (2017). Multiple-human parsing in the wild. *arXiv preprint arXiv:1705.07206*.

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[6] Li, K., Ding, G., & Wang, H. (2018, December). L-FCN: A lightweight fully convolutional network for biomedical semantic segmentation. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* (pp. 2363-2367). IEEE.

[7] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

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

[9] Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001* (Vol. 1, pp. I-I). IEEE.

[10] Kim, T. H., Park, D. C., Woo, D. M., Jeong, T., & Min, S. Y. (2011, October). Multi-class classifier-based adaboost algorithm. In *International Conference on Intelligent Science and Intelligent Data Engineering* (pp. 122-127). Springer, Berlin, Heidelberg.

[11] Ojala, T., Pietikainen, M., & Maenpaa, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, *24*(7), 971-987.

[12] Fu, X., & Qu, H. (2018, December). Research on semantic segmentation of high-resolution remote sensing image based on full convolutional neural network. In *2018 12th International Symposium on Antennas, Propagation and EM Theory (ISAPE)* (pp. 1-4). IEEE.