Automated Face Recognition in the Wild

02/05/2024

Mogli Rohit Kumar - IMT2021503
Josh Jack Thomson Immanuvel - IMT2021515

Introduction

The paper highlights the challenges of masked face recognition (MFR) due to the limited exposure of facial regions caused by mask-occlusion. To address this issue, the authors proposed a dual-branch bidirectional attention module (BAM) that consists of a spatial attention block (SAB) and a channel attention block (CAB) in each branch. The SAB performs bidirectional interactions between the feature maps and their augmented versions to highlight informative spatial locations for feature learning. The learned bidirectional spatial attention maps are then passed through the CAB to assign high weights to informative feature channels, generating an attention-aware feature representation for MFR. The authors claim that their proposed BAM is superior to various state-of-the-art methods in recognizing mask-occluded face images under complex facial variations.

Problem Motivation

The paper is motivated by the challenge of Masked Face Recognition (MFR) which has emerged as a critical problem in facial biometrics due to the widespread use of face masks during the COVID-19 pandemic. The limited exposure of facial regions caused by mask occlusion significantly degrades the performance of existing deep learning-based face recognition methods. To address this issue, the paper proposes a novel Bidirectional Attention Module (BAM) that effectively leverages both spatial and channel attention mechanisms. The key idea is to enable the network to focus on the informative non-occluded facial regions and suppress irrelevant features, enhancing the discriminative power of the learned face representations for robust masked face recognition under complex variations.

Literature Survey

To extend our understanding, we also read the following survey papers on Face Mask Recognition: -

1. "Convolutional Block Attention Module"
2. “Dual Attention Network for scene segmentation”
3. “ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks”

For masking the images of our dataset we have used: -
Convert face dataset to masked dataset

For Plotting the maps of visualization of the attention maps we have used GradCam:-
Visualization of the attention maps

Dataset Analysis

The analysis of the datasets used for evaluating the proposed Bidirectional Attention Model for Masked Face Recognition is as follows.

Training Datasets

1. The training data consists of the CASIA-WebFace-Mixed dataset containing 0.5 million unmasked and 0.3 million masked face images.
2. The face images were aligned and resized to 112 x 112 pixels using MTCNN before applying a face masking tool to generate the synthetic-masked training samples.

Testing Datasets

1. LFW(Labelled Faces in the Wild): This dataset evaluates masked-to-unmasked face matching.
2. CFP-FP(Celebrities in Frontal-Profile): This dataset is also used for masked-to-unmasked face-matching evaluation.
3. CPLFW(Cross-Pose LFW): This dataset tests the model's performance under large pose variations, with masked-to-unmasked face matching.
4. CALFW(Cross-Age LFW): This dataset evaluates masked-to-masked face matching under complex facial variations.

The evaluation metrics used in the paper are :

1. 1:1 face verification accuracy using 10-fold cross-validation.
2. ROC(Receiver Operating Characteristic) curves.
3. AUC(Area under the curve) values.

The results show that the proposed BAM method outperforms various state-of-the-art face recognition baselines and attention modules on most of the testing datasets, demonstrating its effectiveness in recognizing mask-occluded face images under complex facial variations.

Literature Review

Spatial Attention Module:-

• We need to create a mechanism that helps the computer to focus on important facial features when trying to recognize them.

• We start with a grid-like representation of the face, which we call a "feature map". This map has different layers, each representing different aspects of the face, like shape, color, etc.

• Let 'X' be the feature map. SAB rotates X by 90 degrees along the height dimension. It reduces the number of channels bypassing the feature maps through convolution layers

• Using the learned dependencies, the SAB generates attention maps that highlight the most relevant spatial locations within the feature maps. These attention maps guide the network to focus more on informative regions while processing the data.

Channel Attention Module:-

• The CAB takes the feature maps obtained from the previous stages of the network as input. These feature maps contain information about different aspects or features of the input data.

• Initially, the CAB reduces the dimensionality of the feature maps to simplify processing. This reduction helps in focusing on the most relevant information.

• After dimension reduction, the CAB compares different channels within the feature maps to understand their importance relative to each other. It identifies which channels carry more useful information for the task at hand.

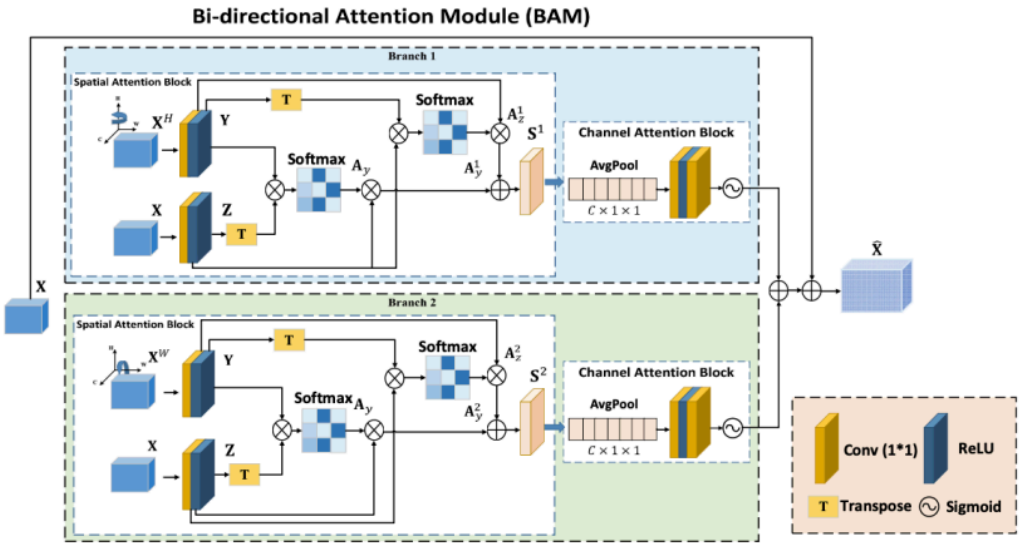
• Using the comparison results, the CAB generates attention weights for each channel. These weights indicate the importance of each channel in contributing to the overall representation of the data.

Bidirectional Attention Module:-

• The focus is to make sure that the computer pays attention to all parts of the face equally.

• BAM incorporates spatial and channel attention blocks to highlight informative spatial locations and feature channels.

• It is the combination of two branches of spatial attention module and channel attention module



Experiments conducted

1. CNN
2. ResNet50
3. VGG16

CNN:-

Convolutional Neural Networks (CNNs) are a class of deep neural networks that are particularly effective for tasks involving images, although they can be applied to other types of data with spatial relationships as well. CNNs are designed to automatically and adaptively learn spatial hierarchies of features from input data.

Here's an overview of the typical architecture of a CNN:

1. Input Layer: The input layer receives the raw data, which in the case of image data, could be an image represented as a grid of pixel values. Each pixel may have multiple channels, such as Red, Green, and Blue (RGB) in the case of color images.
2. Convolutional Layers: Convolutional layers are the core building blocks of CNNs. Each convolutional layer applies a set of learnable filters (also called kernels) to the input image. These filters slide or convolve across the input image, performing element-wise multiplication with the input values and then summing the results to produce feature maps. Convolutional layers are responsible for learning various features of the input data, such as edges, textures, and shapes.
3. Activation Function: After each convolutional operation, a nonlinear activation function is typically applied to introduce nonlinearity into the network. The most common activation function used in CNNs is the Rectified Linear Unit (ReLU), which replaces negative values with zero, helping the network to learn complex patterns more effectively.
4. Pooling Layers: Pooling layers are used to downsample the feature maps produced by the convolutional layers, reducing their spatial dimensions. The most commonly used pooling operation is max pooling, which selects the

maximum value from each patch of the feature map. Pooling helps in reducing the computational complexity of the network and making the learned features more invariant to small translations and distortions in the input data.

5. Fully Connected Layers: After several convolutional and pooling layers, the high-level features learned by the network are flattened into a vector and passed to one or more fully connected layers. These layers act as a classifier, mapping the learned features to the output classes or labels. Each neuron in the fully connected layer is connected to every neuron in the previous layer, allowing the network to learn complex relationships between features.
6. Output Layer: The output layer produces the final predictions or classifications based on the input data. The number of neurons in the output layer corresponds to the number of classes in the classification task, and the output is usually passed through a softmax function to produce probability scores for each class.

CNN architectures can vary significantly in terms of the number of layers, the size of filters, the number of filters, and other hyperparameters. Different architectures may be better suited for specific tasks or datasets, and researchers often experiment with different configurations to achieve optimal performance.

ResNet50:-

ResNet-50 is a convolutional neural network (CNN) architecture that has been widely used for various computer vision tasks, particularly in image classification. It was introduced by Kaiming He et al. in their paper "Deep Residual Learning for Image Recognition" in 2015.

Here's an overview of the architecture:

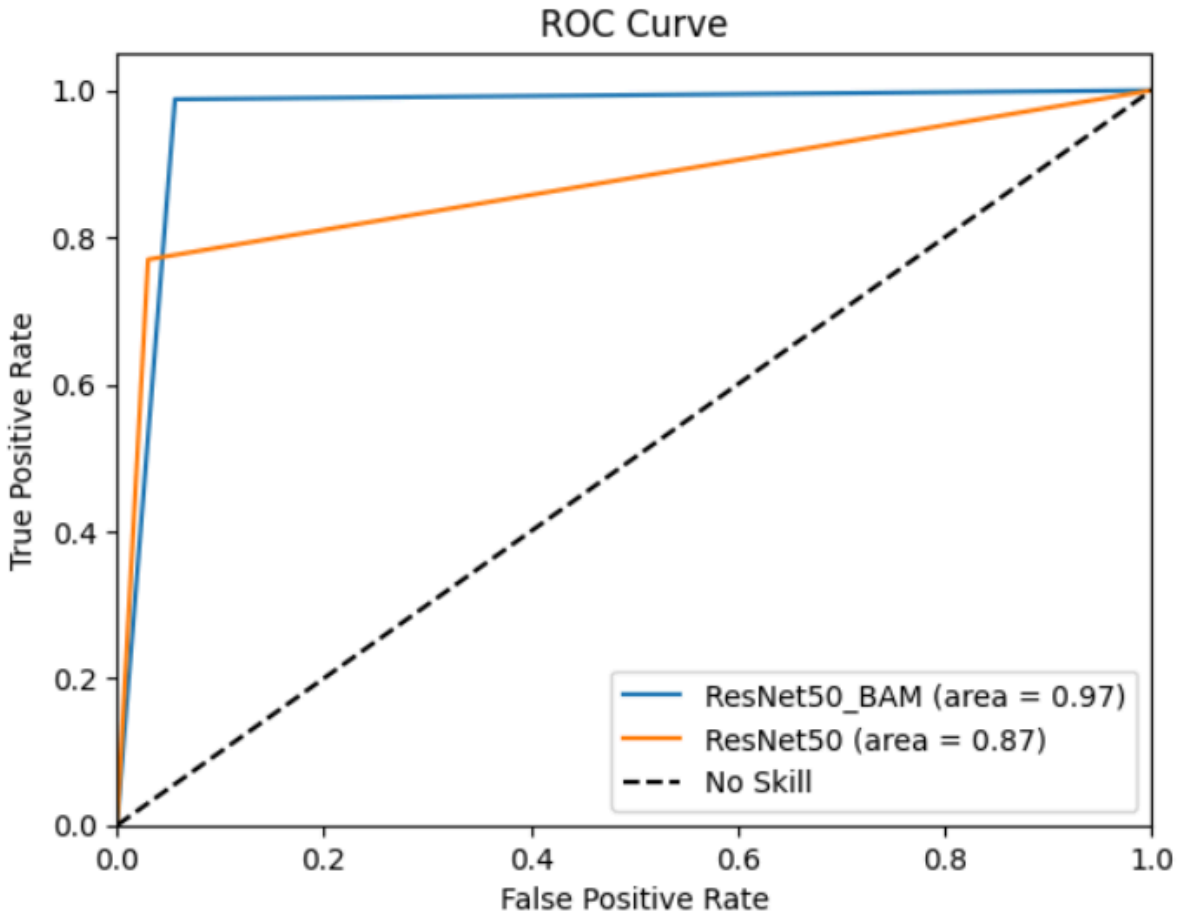
1. Input Layer: The network takes an input image of size 128x128x3 (RGB channels).
2. Convolutional Layers: The initial layers consist of several convolutional layers with small 3x3 filters. These layers perform feature extraction from the input image.
3. Residual Blocks: The core innovation of ResNet is the residual block. Instead of simply stacking layers, ResNet uses residual blocks to learn residual functions with

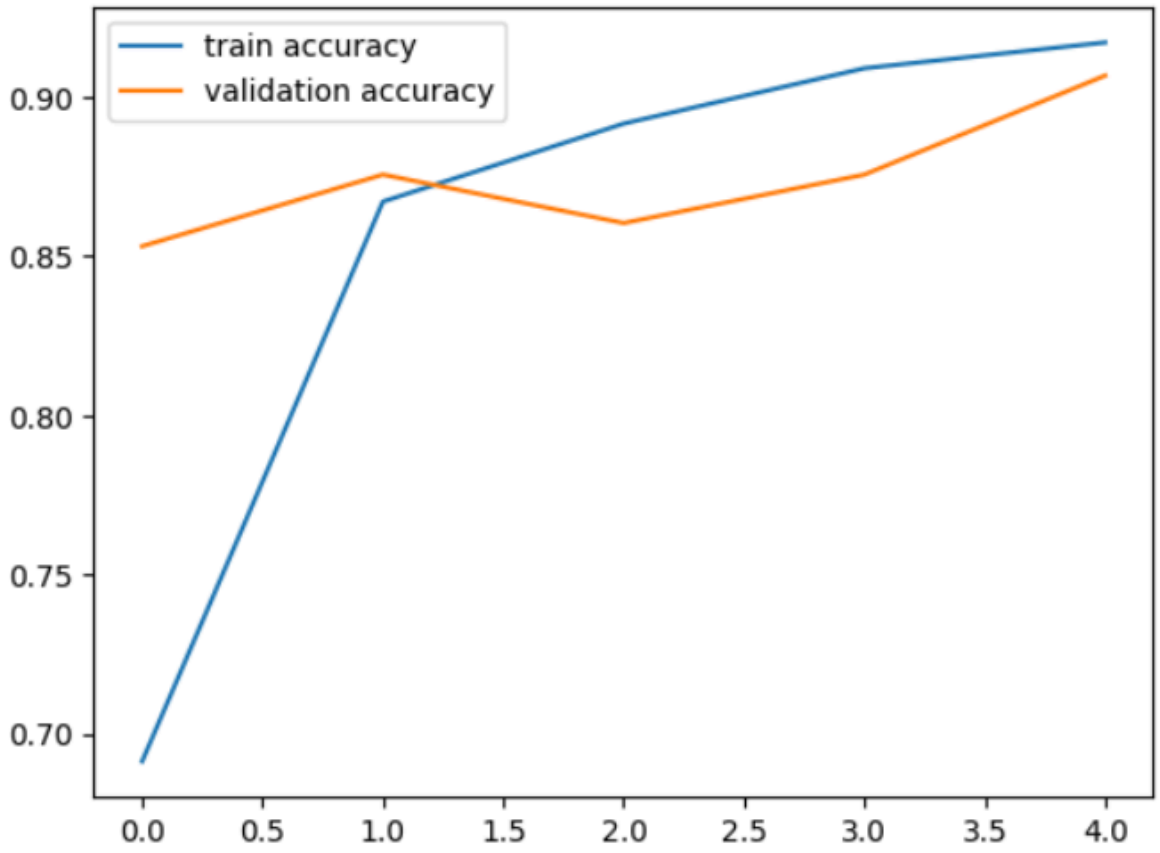
respect to the layer inputs. This is achieved by adding a shortcut or "skip connection" that bypasses one or more layers. The skip connections help alleviate the vanishing gradient problem, making it easier to train very deep networks.ResNet-50 consists of 16 residual blocks grouped into 4 stages. Each stage contains a different number of blocks, with increasing numbers of filters in the convolutional layers.

4. Pooling Layers: After each stage, there is a max-pooling layer that reduces the spatial dimensions of the feature maps, helping to decrease computational complexity and control overfitting.
5. Output Layer: The output layer produces a vector of probabilities indicating the likelihood of the input image belonging to each class in the classification task. In the case of ImageNet classification (for which ResNet-50 was originally designed), there are 1000 output classes.

ResNet-50 is called "50" because it has 50 layers, including convolutional and fully connected layers. However, this count doesn't include the pooling layers and the activation layers.

ROC curve:-


ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
ResNet50_BAM (area = 0.97)
ResNet50 (area = 0.87)
No Skill



VGG16:-

Certainly! VGG16 is a convolutional neural network (CNN) architecture proposed by the Visual Geometry Group (VGG) at the University of Oxford. It was introduced in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" by Karen Simonyan and Andrew Zisserman in 2014.

Here's an overview of the VGG16 architecture:

1. Input Layer: Like many CNN architectures, VGG16 takes an input image of fixed size. In the original implementation, the input size is 224x224x3 (RGB channels).

2. Convolutional Layers: VGG16 consists of 13 convolutional layers stacked one after the other. Each convolutional layer uses a small receptive field of 3x3 pixels, and the stride is fixed to 1 pixel. These layers are followed by rectified linear unit (ReLU) activation functions, which introduce non-linearity into the network.
3. Pooling Layers: After every two or three convolutional layers, max-pooling layers with a 2x2 filter and a stride of 2 pixels are applied. Max-pooling reduces the spatial dimensions of the feature maps, helping to decrease computational complexity and control overfitting.
4. Fully Connected Layers: Following the convolutional and pooling layers, VGG16 has three fully connected layers. These layers serve as a classifier, mapping the high-level features learned by the convolutional layers to the output classes. The last fully connected layer usually has 1000 units, corresponding to the 1000 classes in the ImageNet dataset, which was the original dataset used for training VGG16.
5. Output Layer: The output layer produces a vector of probabilities indicating the likelihood of the input image belonging to each class in the classification task. In the case of ImageNet classification, as mentioned earlier, there are 1000 output classes.

VGG16 is characterized by its simplicity and uniformity in architecture. It has a straightforward design with a focus on stacking multiple layers of small-sized filters and pooling layers, which allows it to learn complex features from images. Despite being deeper than many previous architectures at the time of its introduction, VGG16 is relatively easy to understand and has been widely adopted as a baseline architecture for various computer vision tasks.

Summary Table

Machine Learning Model	Accuracy	BAM
CNN	92.06	With
CNN	91.17	Without
ResNet50	97.02	With
ResNet50	96.67	Without
VGG16	96.64	With
VGG16	95.78	Without

GitHub Link for Colab notebook