Detecting whether the person is wearing masks or not

Manav Kapoor, Neeraj

May 17, 2021

Abstract

The requirements of the final project of the course requires us to build a model capable of classifying a dataset of images with and without masks.The initial model used Principal Component Analysis and fully connected Convolutional neural network. Since the data is in the form of images, the input features for any of these models will be too huge for these models to handle. Hence we used a pretrained model for extracting facial features using. For this we used he concept of HuMoments and Haralick Texture. By using these concepts we were able to get an feature vector as an output which we later used these extracted features to train various Machine Learning Models and classify the images.

I. Introduction

Nowadays, a large amount of data is available everywhere. Therefore, it is very important to analyze this data in order to extract some useful information and to develop an algorithm based on this analysis. This can be achieved through data mining and machine learning. Machine learning is an integral part of artificial intelligence, which is used to design algorithms based on the data trends and historical relationships between data. Now let's talk about the given problem statement. We are given a dataset in which we need to classify whether the person is wearing a mask or not.

We used a fully fledged Convolutional Neural Network and MobileNet version 2 as a base classifier in our second method of analysis for detection of images with masks and without masks

II. Feature-Extraction

First we extracted features for each image from our given dataset. For image extraction we used the concepts of HuMoments, Haralick Texture and Histogram. Let's understand each of them first one by one:

A. Understanding HuMoments

In computer vision and image processing, image moments are often used to characterize the shape of an object in an image. These moments capture basic information such as the area of the object, the centroid (i.e. the center (x, y)-coordinates of the object), the orientation, and other desirable properties. Hu Moments are an image descriptor utilized to characterize the shape of an object in an image. The Hu Moments descriptor returns a real-valued feature vector of 7 values. These 7 values capture and quantify the shape of the object in an image. We can then compare our shape feature vector to other feature vectors to determine how "similar" two shapes are.

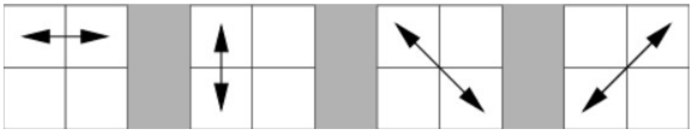


Fig. 1. Four directions of adjacency as defined for calculation of the Haralick texture features. The Haralick statistics are calculated for co-occurrence matrices generated using each of these directions of adjacency.[2].

B. Understanding Haralick Texture

Haralick texture features are used to describe the "texture" of an image. Haralick features are derived from the Gray Level Co-occurrence Matrix (GLCM). This matrix records how many times two gray-level pixels adjacent to each other appear in an image. Then based on this matrix, Haralick proposes 13 values that can be extracted from the GLCM to quantify texture. An additional 14 values can be computed; however, they are not often used due to computational instability

C. Understanding cv2.calcHist

A histogram represents the distribution of pixel intensities (whether color or grayscale) in an image. It can be visualized as a graph (or plot) that gives a high-level intuition of the intensity (pixel value) distribution. We are going to assume a RGB color space in this example, so these pixel values will be in the range of 0 to 255.

When plotting the histogram, the x-axis serves as our "bins." If we construct a histogram with 256 bins, then we are effectively counting the number of times each pixel value occurs.

III. Evaluation Techniques

Central to evaluating the performance of supervised learning algorithms is the notion of training and testing datasets. The training set contains examples of network flows from different classes (network applications) and is used to build the classification model. The testing set represents the unknown network traffic that we wish to classify. The flows in both the training and testing sets are labelled with the appropriate class. As we know the class of each flow within the datasets we are able to evaluate the performance of the classifier by comparing the predicted class against the known class. We were given images of random sizes. We first resized them into a 264x264 array, then performed the above feature extraction technique on each image and concatenated each of the output into a numpy array which resulted in a vector size of 276. Finally we were having 7127 vectors each of size 276 out of which 5000 were labelled as 0(without masks) and 2127 were labelled as 1(with masks). The main motive behind such feature extraction was to detect any sudden histographic change in the face features or color feature on the face which could only be caused due to the presence of mask on the face. Our task is to identify such spike using different models and comparing their accuracies.

To test and evaluate the algorithms we use k-fold cross validation. In this process the data set is divided into k subsets. Each time, one of the k subsets is used as the test set and the other k-1 subsets form the training set. Performance statistics are calculated across all k trials. This provides a good indication of how well the classifier will perform on unseen data. We use k=5 and compute the accuracy of the models Accuracy, i.e. the percentage of correctly classified instances over the total number of instances.

A. Comparison of Accuracies b/w different models

Once we had the visualized dataset we did train-test split of 85-15 We used the training dataset to train 4 models that are - Logistic Regression,Linear Discriminant Analysis and Random Forest Classifier

*1) **Logistic Regression**:* A model of Logistic Regression was trained with default parameters from sklearn. On the validation set, the model gave an accuracy of 0.827999

*2) **Decision Tree Classifier**:* A model of Decision Tree Classifier was trained with default parameters from sklearn. On the validation set, the model gave an accuracy of 0.8755555555555555

*3) **Random Forest Classifier**:* A model of Random Forest Classifier was trained with default parameters from sklearn. On the validation set, the model gave an accuracy of 0.8412

*4) **Gaussian Naive Bayes Classifier**:* A model of Gaussian Naive Bayes Classifier was trained with default parameters from sklearn. On the validation set, the model gave an accuracy of 0.7888

*5) **SVM**:* A model of SVM was trained with two parameters Linear and RBf(Radial Basis Function). On the validation set, the model gave an accuracy of SVM(Linear) 0.8410 and SVM(Radial Basis Function) 0.741

IV. Dimensionality reduction

For improving accuracies we thought to reduce the dimensions which was initially 276 features,using techniques of PCA and LDA:
1. **Principal Component Analysis(PCA)**: Our features sets got reduced to 7 principal components on preserving 90 percent of data variation.

TABLE I
Comparison of model scores on test set after PCA

Logistic Regression	Gaussian Naive Bayes	SVM(RBF)	SVM(Linear)	Random Forest Classifier	Decision Tree Classifier
0.820	0.77	0.84	0.74	0.82	0.80

It is evident that after applying PCA, it does cause any significant increase in accuracies of various classifiers.

2. **Linear Discriminant Analysis**: On applying LDA without any preset parameters we got reduction to only dimension LD.

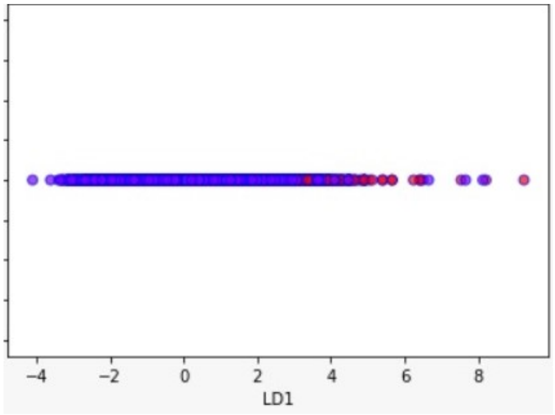


TABLE II
Comparison of model scores on test set after PCA

Logistic Regression	Gaussian Naive Bayes	SVM(RBF)	SVM(Linear)	Random Forest Classifier	Decision Tree Classifier
0.8596	0.9074	0.8956	0.8922	0.8532	0.8599

It is evident that we have received a significant increase in accuracies of classifiers specially in Gaussian and SVM(Radial Basis Function)

V. Using Dense Neural network

As discussed above, it is evident that we need some sort of high level feature extraction mechanism to detect images with and without masks. Also, since these features are usually abstract, we cannot supply/extract these features manually hence we need some sort of Deep Learning method to do this for us. A natural choice would be Deep Neural Networks. Further thought into this problem also leads us to think that CNNs (Convolutional Neural Networks) will be an excellent choice for this task as they are great for extracting features which depend on the data around it, i.e. it has a short range context dependence. They are also one of the most popular Neural Network Architectures that is used for image style data.

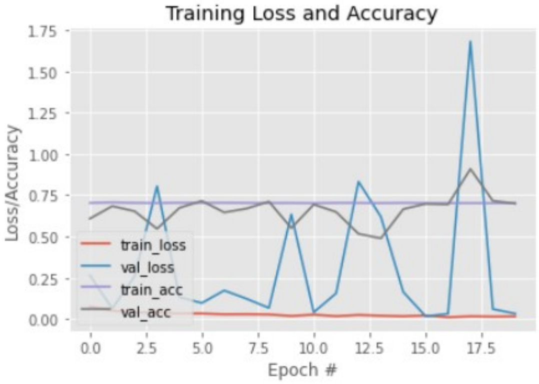
The MobileNetV2 architecture is based on an inverted residual structure where the input and output of the residual block are thin bottleneck layers opposite to traditional residual models which use expanded representations in the input an MobileNetV2 uses lightweight depthwise convolutions to filter features in the intermediate expansion layer. Additionally, we find that it is important to remove non-linearities in the narrow layers in order to maintain representational power. We demonstrate that this improves performance and provide an intuition that led to this design. Finally, our approach allows decoupling of the input/output domains from the expressiveness of the transformation, which provides a convenient framework for further analysis. We measure our performance on Imagenet classification, COCO object detection, VOC image segmentation. We evaluate the trade-offs between accuracy, and number of operations measured by multiply-adds (MAdd), as well as the number of parameters

A. *Results after applying DNN*

TABLE III
CONFUSION MATRIX

Classes	Precision	Recall	F1- Score	Support
0	0.99	0.99	0.99	1000
1	0.98	0.99	0.98	426

We finally ended up with an accuracy of **99 percent** WOW



VI. CONCLUSION AND FUTURE WORK

In this paper we analysed two different techniques for face mask detection. In one we extracted the features based on histographic data of face colors.

We observed that we were getting maximum accuracies when our classifiers were linear(SVM(Linear) and SVM(RBF)). We also saw a significant change in accuracy of Gaussian Naive Bayes classifier when we applied LDA on our dataset.

Ensembled and tree classifiers were not able to classify the data much accurately and also on increasing the dimensionality of data using kernel such as RBF there is a significant drop in accuracy of linear Kernels.

In the second technique we used a pretrained model MobileNetV2 as a base model with three more dense layers with RELU activation and finally compiled our model with Adam optimizer and loss function as CategoricalCrossEntropy. We received an accuracy of 99 percent our this dense network

Future Work

For future work we proposed investigating low accuracies on applying PCA reduction and by investigating linearity of our feature data and it collapsed into a one dimensional LDA vector. We would also like to deploy our Neural Network on live data cams to capture more data and analyse its performance. On deploying our model on to a live cam to fit our model with environmental conditions which could be a downside of low performance of our first analysis.

VII. CONTRIBUTIONS

Both the Authors have equal Contribution and have been sorted alphabetically

VIII. REFERENCES

[1] https://cvexplained.wordpress.com/2020/07/21/10-4-hu-moments/
[2] https://cvexplained.wordpress.com/2020/07/22/10-6-haralick-texture/
[3] https://www.pyimagesearch.com/2021/04/28/opencv-image-histograms-cv2-calchist/
[4] https://www.researchgate.net/publication/341798630$_{C}lassification_{o}f_{t}omato_{l}eaf_{d}iseases_{u}sing_{M}obileNet_{v}2/fulltext/5ed544252$
$of-tomato-leaf-diseases-using-MobileNet-v2.pdf$