Mask detectio

Contents

No table of contents entries found.

No table of figures entries found.

project's domain background

the project is in deep learning field and computer vision domain, and image preprocessing.

Computer vision is one of the most used field these days specially with the improvement in deep learning. I choose this project for two reasons:

- Improve my CNN and image preprocessing and intuition and having more experience in this field.
- For fun, I am enjoying of learning and doing new things.

One of the similar projects in real life is a social distance detection in amazon warehouse, where it is detect if two workers were near to each other.

problem statement

the problem is how to detect the people are not wearing mask and localizing the person in the image by putting a box around his face. The detected faces should be labeled to three categories wear mask, not wear mask, wear mask incorrectly. This is a research paper for face recognition and localizing it should be useful for this project, link HERE.

datasets

the dataset used is a face mask detection data from Kaggle website. The data set is created by Ashish Jangra.

The dataset page here.

Data set info:

Training examples	12k
Number of labels (classes)	2
labels	• With mask • Without mask
bounding boxes	In PASCAL VOC format using mediapipe library

Data set features:

- Generalization
- People in the images has different background
- More than one face in a one image

Why this data set?

It has more than person in one image, different image size, collected from different distribution and labeled correctly.

solution statement

the suggested solution is using deep learning to solve the problem by the following techniques Convolutional neural network and VVG19 CNN architecture. With help of keras and tensor flow packages and OpenCV API this problem can be solved easily and fast because of GPU acceleration.

benchmark model

There is a lot of projects done on the same data set and with great result, for example HERE a project that use VVG19 and reach 98 % accuracy, this project will the reference project of this project.

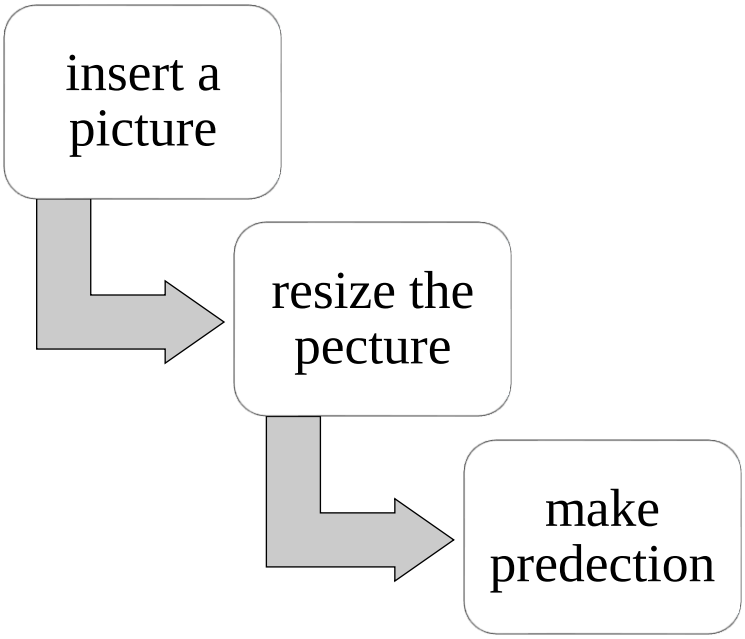
evaluation metrics

the dataset is unbalanced so the preferred evaluation metrics here is F1-score and accuracy.

project design

the project design will be in two phases training and testing. In training phase, the data will be loaded and processed if needed then the train the VVG19 model. In the testing phase the model will be evaluated according to the chosen metric which is F1-score.

The deploy pipeline



Project implementation and deploying

In the project there are two main tasks:

- Load and preprocess data
- Training
- Testing

After the tasks above completed the project will be ready for *deployment*.

Load and preprocess data

All project files are in Faris-ML git hub repository, data set cloned from git hub in three groups (Train , Validation, Test).

	Number of examples
Train	10000
Test	992
Validation	800

Process the data by applying data augmentation methods, methods used:

- Scaling.
- Brightness.
- Zoom.
- Shear.
- Horizontal flip.

Training

The used algorithm for mask detection is CNN (convolutional neural network) and the architecture is VVG19 with adding a few layers, all neural network architecture specification will be written in details in table 2.

Type	Filter size	Filter channel	stride	Zero padding	Output size
Input	-	-	-	-	128 x 128 x 3
Conv 2d	3 x 3	64	1	1 x 1	127 x 127 x 64
Conv 2d	3 x 3	64	1	1 x 1	126 x 126 x 64
Max pooling	2 x 2	-	2	-	63 x 63 x 64
Conv 2d	3 x 3	128	1	1 x 1	62 x 62 x 128
Conv 2d	3 x 3	128	1	1 x 1	61 x 61 x 128
Max pooling	2 x 2	-	2	-	31 x 31 x 128
Conv 2d	3 x 3	256	1	1 x 1	30 x 30 x 256
Conv 2d	3 x 3	256	1	1 x 1	29 x 29 x 256
Conv 2d	3 x 3	256	1	1 x 1	28 x 28 x 256
Conv 2d	3 x 3	256	1	1 x 1	27 x 27 x 256
Max pooling	2 x 2	-	2	-	14 x 14 x 256
Conv 2d	3 x 3	512	1	1 x 1	13 x 13 x 512
Conv 2d	3 x 3	512	1	1 x 1	12 x 12 x 512
Conv 2d	3 x 3	512	1	1 x 1	11 x 11 x 512
Conv 2d	3 x 3	512	1	1 x 1	10 x 10 x 512
Max pooling	2 x 2	-	2	-	5 x 5 x 512
Conv 2d	3 x 3	512	1	1 x 1	4 x 4 x 512
Conv 2d	3 x 3	512	1	1 x 1	3 x 3 x 512
Conv 2d	3 x 3	512	1	1 x 1	2 x 2 x 512
Conv 2d	3 x 3	512	1	1 x 1	1 x 1 x 512
Max pooling	1 x 1	-	1	-	1 x 1 x 512
Flatten	-	-	-	-	512
Dense	4096	-	-	-	-
Dense	4096	-	-	-	-
Dense	1000	-	-	-	-
Dense	128	-	-	-	-
Dense	64	-	-	-	-
Dense	32	-	-	-	-
Dense	16	-	-	-	-
Dense	8	-	-	-	-
Dense	4	-	-	-	-
Dense	2	-	-	-	-

All layers has Relu activation function and the output layer has softmax activation function.

The model trained by transfer learning on image-net weights for 20 epochs.

Testing

The goal is to reach 98 % accuracy and with help with confusion matrix all performance measures can be calculated.

Confusion matrix:

		actual	
		1	0
predicted	1	476	7
	0	4	505

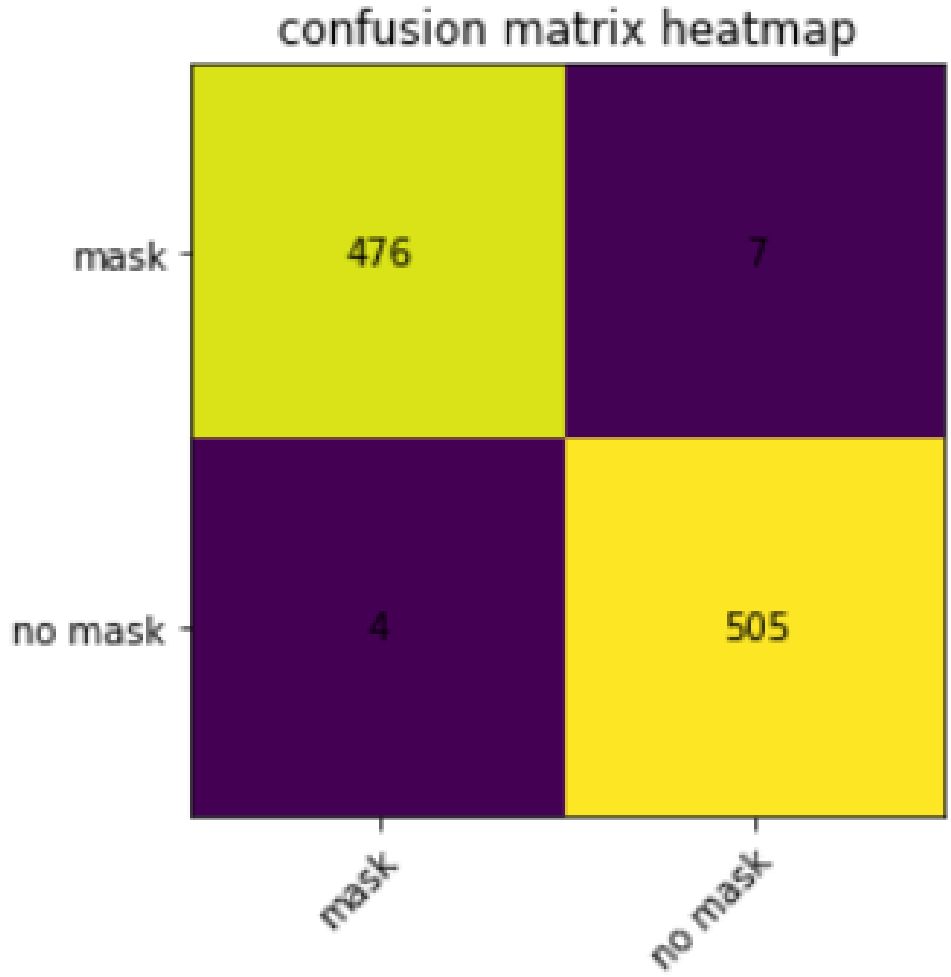
Performance measures:

Performance measure	The score

Accuracy	0.989
Misclassification rate	0.011
True positive rate	0.992
False positive rate	0.014
True negative rate	0.986
precision	0.986
prevalence	0.516
Balanced accuracy	0.989
F1 score	0.989

The model is accurate with 0.989 score.

Heat map:



Used tools

For accomplished this task some needed tools used.

APIs:

- Sklearn.

For performance measures.

- Tensorflow:

For training and building the neural network.

- Matplotlib:

For plotting and visualization.

- Open CV

For camera capture and image reading.

- Keras
- Numpy

Linear algebra operations.

- Mediapipe

For face detection.

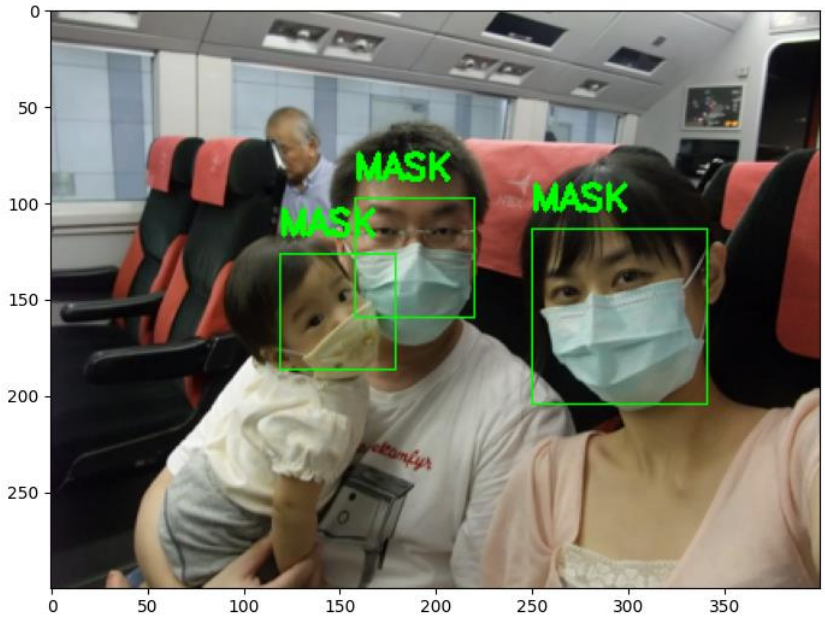
IDEs:

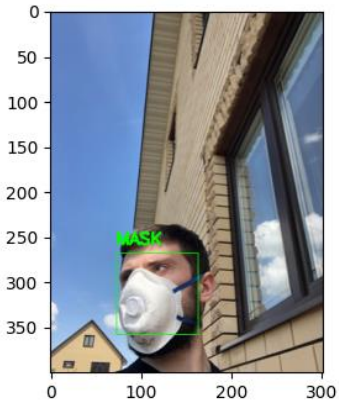
Google colab: because the limitations in computational power.

Pycharm: using the personal device camera for maskdetecting.

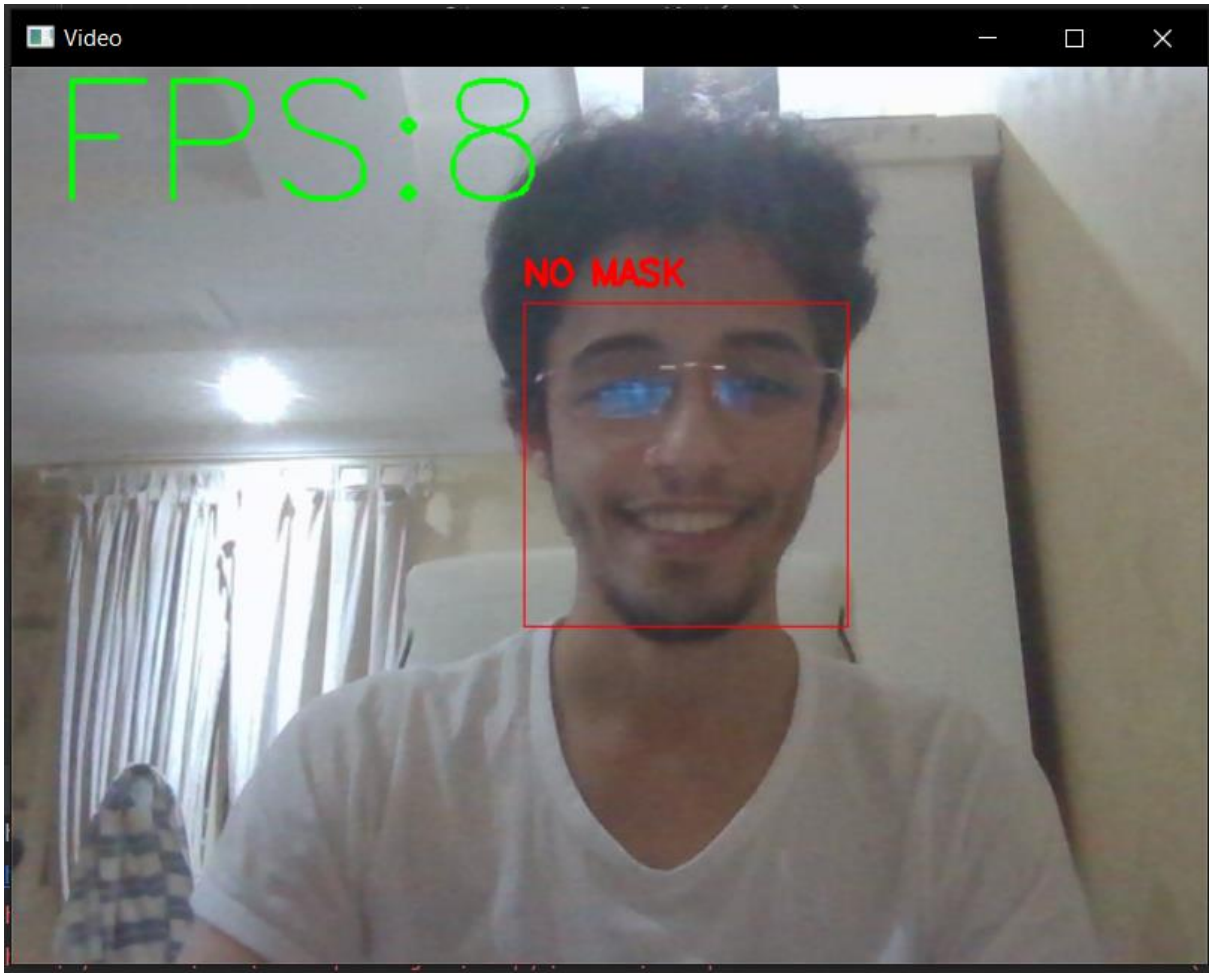
Sample output

Images:





My own video capturing:



Submitted files:

All submitted files with any necessary hints for using the file content.

mrain.py: python file for training task.

mrain.ipynb: notebook file for training task.

model_testing.py: python file for testing task.

model_testing.ipynb: notebook file for testing task.

mask_detection.py: python file for deployment and pipeline inference.

masknet.h5: saved model in h5 format.

The rest files are data set and reports.