

FACE MASK DETECTOR

Oggetto: Face mask detector utilizzando la Visione Artificiale in Python. Progetto per la sicurezza contro il COVID-19.

In questo progetto, costruiremo un modello per rilevare se una persona indossa una mascherina in tempo reale. Lo faremo usando i concetti di visione artificiale, usando la libreria OpenCV e Keras.

Linguaggio di programmazione: Python

Strumenti e librerie: OpenCV, TensorFlow, Numpy, Keras

IDE: Visual Studio Code

Prerequisites: Python, Deep Learning, Machine Learning

PRIMA DI INIZIARE INSTALLARE IL SEGUENTE

- Python 3.9
- TensorFlow
- Numpy
- OpenCV
- imutils

CODICE

- **train.py**

Con questo codice alleniamo la Rete Neurale.

Costruzione della Rete Neurale.

```
model = Sequential([
    Conv2D(100, (3,3), activation='relu', input_shape=(150, 150,
3)),
    MaxPooling2D(2,2),

    Conv2D(100, (3,3), activation='relu'),
    MaxPooling2D(2,2),

    Flatten(),
    Dropout(0.5),
    Dense(50, activation='relu'),
    Dense(2, activation='softmax')
])
model.compile(optimizer='adam', loss='binary_crossentropy',
metrics=['acc'])
```

Questa rete di convoluzione è costituita da due coppie di livelli Conv e MaxPool per estrarre le funzionalità dal set di dati. È seguito da un livello Flatten e Dropout per convertire i dati in 1D e garantire l'overfitting.

Successivamente, due Dense layer per la classificazione.

Generazione/aumento dei dati di immagine.

```
TRAINING_DIR = "Dataset/train"
train_datagen = ImageDataGenerator(rescale = 1.0 / 255,
                                    rotation_range = 40,
                                    width_shift_range = 0.2,
                                    height_shift_range = 0.2,
                                    shear_range = 0.2,
                                    zoom_range = 0.2,
                                    horizontal_flip = True,
                                    fill_mode = 'nearest')

train_generator = train_datagen.flow_from_directory(TRAINING_DIR,
                                                    batch_size = 100,
                                                    target_size = (150, 150))
```

Inizializzazione di un checkpoint di richiamata per continuare a salvare il modello migliore dopo ogni epoch durante l'allenamento.

```
checkpoint = ModelCheckpoint('model2-{epoch:03d}.model', monitor='val_loss', verbose=0, save_best_only=True, mode='auto')
```

Allenamento del modello.

```
history = model.fit_generator(train_generator,
                              epochs = 30,
                              validation_data=validation_generator,
                              callbacks=[checkpoint])
```

- **test.py**

Con questo codice testiamo la Rete Neurale.

Una volta che il programma è eseguito, premi ESC per uscire dal programma.

Caricamento del modello.

```
model = keras.models.load_model("./model-010.h5")
```

Inizializzazione della webcam.

```
webcam = cv2.VideoCapture(0) # Use camera 0
```

Caricamento del file .xml in cui é definita la faccia da un punto di vista frontale.

```
classifier =  
cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
```

Inizio del rilevamento in tempo reale.

```
while True:  
    (rval, im) = webcam.read()  
    # Resize the image to speed up detection  
    mini = cv2.resize(im, (im.shape[1] // size, im.shape[0] //  
size))  
  
    # detect MultiScale / faces  
    faces = classifier.detectMultiScale(mini)
```

Dopo aver rilevato il viso, disegna un rettangolo attorno ad esso e considera solo quel pezzo di immagine per rilevare la mascherina.